ITHEA

International Journal INFORMATION TECHNOLOGIES & KNOWLEDGE

2015 Volume 9 Number 4

International Journal
INFORMATION TECHNOLOGIES & KNOWLEDGE

Volume 9 / 2015, Number 4

International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" (IJ ITK)
is official publisher of the scientific papers of the members of
the ITHEA International Scientific Society

IJ ITK rules for preparing the manuscripts are compulsory.
The **rules for the papers** for IJ ITK are given on www.ithea.org

Responsibility for papers published in IJ ITK belongs to authors.

International Journal "INFORMATION TECHNOLOGIES & KNOWLEDGE" Volume 9, Number 4, 2015
Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
Institute of Mathematics and Informatics, BAS, Bulgaria; V.M.Glushkov Institute of Cybernetics of NAS, Ukraine;
Universidad Politechnika de Madrid, Spain; Hasselt University, Belgium;
St. Petersburg Institute of Informatics, RAS, Russia; Institute for Informatics and Automation Problems, NAS of the Republic of Armenia.
Printed in Bulgaria
Publisher ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Technical editor: Ina Markova

Издател: ИТЕА®, София 1000, ПК 775, България, www.ithea.org, e-mail: info@foibg.com

ISSN 1313-0455 (printed) **ISSN 1313-048X (online)**

MAIN DIFFERENCES BETWEEN MAP/REDUCE AND COLLECT/REPORT PARADIGMS

Krassimira Ivanova

Abstract: *This article presents main differences between Map/Reduce (MRP) and Collect/Report (CRP) paradigms. The most important difference is that in MRP the calculations and data must be all completely independent. In opposite, the CRP assumes that all data are interconnected and may be processed in common, taking in account all interconnections.*

Keywords: *Map/Reduce Paradigm, Collect/Report Paradigm, Big Data, Cloud computing*

ACM Keywords: *E.1 Data Structures; Distributed data structures.*

Introduction

Nowadays, *data-intensive* computing problems are emerging. In contrast to the traditional computing problems, data-intensive problems demonstrate the following features [Lockwood, 2015]:

- **Input data is far beyond gigabyte-scale**: datasets are commonly on the order of tens, hundreds, or thousands of terabytes;
- **They are I/O-bound**: it takes longer for the computer to get data from its permanent location to the CPU than it takes for the CPU to operate on that data.

The Map/Reduce Paradigm (MRP) is a way of solving a certain subset of parallelizable problems that gets around the bottleneck of ingesting input data from disk. Whereas traditional parallelism brings *the data to the compute*, map/reduce does the opposite, it brings *the compute to the data.* Below we will remember the main features of map/reduce paradigm following the work [Lockwood, 2015].

In Map/Reduce, the input data is not stored on a separate, high-capacity storage system. Rather, the data exists in little pieces and is permanently stored on the compute elements. This allows our parallel procedure to follow these steps:

1. We do not have to move any data since it is pre-divided and already exists on nodes capable of acting as computing elements;
2. All of the parallel worker functions are sent to the nodes where their respective pieces of the input data already exist and do their calculations;

3. All of the parallel workers communicate their results with each other move data if necessary, and then continue the next step of the calculation.

Thus, the only time data needs to be moved is when all of the parallel workers are communicating their results with each other in point 3 above. There is no more serial step where data is being loaded from a storage device before being distributed to the computing resources because the data already exists on the computing resources.

Hadoop is an actual implementation of Map/Reduce. Hadoop, perhaps the most widely used Map/Reduce framework, accomplishes this feat using the Hadoop Distributed File System (HDFS). HDFS is fundamental to Hadoop because it provides the data chunking and distribution across compute elements necessary for Map/Reduce applications to be efficient.

The main features of Map/Reduce and Hadoop are:

- Map/Reduce brings *compute to the data* in contrast to traditional parallelism, which brings data to the compute resources;
- Hadoop accomplishes this by storing data in a replicated and distributed fashion on HDFS;
- HDFS stores files in chunks which are physically stored on multiple compute nodes;
- HDFS still presents data to users and applications as single continuous files despite the above fact;
- Map/Reduce is ideal for operating on very large, flat (unstructured) datasets and perform trivially parallel operations on them;
- Hadoop jobs go through a Map stage and a Reduce stage where:
    - The Mapper transforms the raw input data into key-value pairs where multiple values for the same key may occur;
    - The Reducer transforms all of the key-value pairs sharing a common key into a single key with a single value.

Of course, for the compute elements to be able to do their calculations on these chunks of input data, the calculations and data must be all completely independent from the input data on other compute elements. This is the principal constraint in Map/Reduce jobs: *Map/Reduce is ideally suited for trivially parallel calculations on large quantities of data*, but if each worker's calculations depend on data that resides on other nodes, one will begin to encounter rapidly diminishing returns [Lockwood, 2015].

This constraint causes the need of principally other paradigm for storing and processing Big Data. Such paradigm is so called “Collect/Report Paradigm” [Markov & Ivanova, 2015]. The goal of this paper is to outline the main differences between these two paradigms as well as to propose a possible convergent paradigm which may unite the positive features of both paradigms.

Collect/Report Paradigm

The Collect/Report Paradigm is based on the possibility of so called “Natural Language Addressing” (NLA) [Markov et al, 2015].

CRP assumes that incoming information is coded in RDF format. In Collect/Report Paradigm, all nodes have to “listen” in parallel the incoming stream of RDF-data and to “collect” (to store) information only in the layers the nodes have to support. In the same time, nodes have to “listen” incoming stream of requests and only nodes, which have information corresponded to given request has to “report” (to send answer).

Main advantages of Collect/Report Paradigm are:

- *Collecting information is done by all nodes independently in parallel. It is possible one node to send information to another;*
- *Reporting information is provided only by the nodes which really contain information related to the request; the rest nodes do not react, they remain silent;*
- *Input data as well as results are in RDF-triple or RDF-quadruple format.*

CRP is a good foundation for intelligent data processing based on multi-dimensional memory structures [Markov et al, 2013]. In addition, via CRP and natural language addressing, three main problems of storing Big Data may be solved [Markov et al, 2014]:

- *Volume – avoiding additional indexing, duplication of keywords, and corresponded pointers, leads to reducing additional memory needed for accessing information i.e. we may use addressing but not classical search engines;*
- *Velocity – avoiding recompilation of information base permits high speed of storing and immediately readiness of information to be accessed. This is very important possibility for stream data;*
- *Variety – natural language addressing permits creating a special kind of graph information bases which may operate both with structured as well as semi-structured information.*

Main Differences Between Map/Reduce and Collect/Report Paradigms

Map/Reduce Paradigm (MRP) and Collect/Report Paradigm (CRP) are two different approaches for operating on very large, flat (unstructured) datasets (Big Data). The main differences between these two paradigms may be systematizes as follow:

- MRP performs trivially parallel operations and results are couples (keyword, value). The CRP is designed to cover the case when the data are represented in RDF-format and the results are triples (subject, relation, object);
- *In CRP, reporting information is provided only by the nodes which really contain information related to the request; the rest nodes do not react, they remain silent. In MRP all nodes send resulting information assuming that all reported data is needed for end user;*
- *An important advantage of the CRP is reducing the traffic to and from the cloud structures for storing data in RDF-format and readiness to extract information within microseconds after it has been stored;*
- *The most important difference is that in MRP the calculations and data must be all completely independent. In opposite, the CRP assumes that all data are interconnected and may be processed in common, taking in account all interconnections.*

Conclusion

In this article we have outlined the main differences between the Map/Reduce and Collect/Report paradigms. Concluding, we may propose a convergent paradigm “Map/Collect/Report” (MCRP).

MRP expects that the data is pre-divided and already exists on nodes capable of acting as computing elements. After Mapping phase the data is formatted in format of couples (keyword, value).

CRP expects that the incoming data is in RDF format. How the data are prepared in RDF format is not commented. One possible variant is to use mapping, similar to MRP but to generate triples as intermediate result.

This way, in MCRP, we have the sequence:

- Mapping Big Data to RDF-triples or quadruples;
- Collecting RDF information in hyper-graph structures;
- Reporting only requested information.

Bibliography

[Lockwood, 2015] Glenn K. Lockwood, “Conceptual Overview of Map/Reduce and Hadoop”, http://www.glennklockwood.com/di/hadoop-overview.php#compare (accessed on 23.02.2015)

[Markov & Ivanova, 2015] Krassimir Markov, Krassimira Ivanova, “General Structure of Collect/Report Paradigm for Storing and Accessing Big Data”, International Journal “Information Theories and Applications”, Vol. 22, Number 3, 2015, ISSN 1310-0513 (printed), ISSN 1313-0463 (online), pp. 266 - 276.

[Markov et al, 2013] Krassimir Markov, Koen Vanhoof, Iliya Mitov, Benoit Depaire, Krassimira Ivanova, Vitalii Velychko and Victor Gladun, "Intelligent Data Processing Based on Multi-Dimensional Numbered Memory Structures", Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems, IGI Global, 2013, pp. 156-184, doi: 10.4018/978-1-4666-1900-5.ch007, ISBN: 978 1-4666-1900-5, EISBN: 978-1-4666-1901-2

[Markov et al, 2014] Krassimir Markov, Krassimira Ivanova, Koen Vanhoof, Benoit Depaire, Vitalii Velychko, Juan Castellanos, Levon Aslanyan, Stefan Karastanev, „Storing Big Data Using Natural Language Addressing”, In: N. Lyutov (ed.), int. Sc. Conference “Informatics in the Scientific Knowledge”, VFU, Varna, Bulgaria, 2014, ISSN: 1313-4345, pp. 147-164.

[Markov et al, 2015] Krassimir Markov, Krassimira Ivanova, Koen Vanhoof, Vitalii Velychko, Juan Castellanos, „Natural Language Addressing”, ITHEA® Hasselt, Kyiv, Madrid, Sofia, IBS ISC No.: 33, 2015, ISBN: 978-954-16-0070-2 (printed), ISBN: 978-954-16-0071-9 (online), 315 p.

Authors' Information

Krassimira Ivanova *– University of National and World Economy, Sofia, Bulgaria;*

e-mail: krasy78@mail.bg

Major Fields of Scientific Research: Software Engineering, Business Informatics, Data Mining, Multidimensional multi-layer data structures in self-structured systems

ОЦЕНКА РАСПРЕДЕЛЕНИЯ РЕШЕНИЯ НЕЧЁТКИХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Алексей Бычков, Евгений Иванов, Ольга Супрун

Abstract: *В статье разработаны методы нахождения оценки распределения решения для нового класса нечётких дифференциальных уравнений, которые содержат нечёткий процесс в правой части. Приведён пример использования полученных методов.*

Keywords: *нечеткая логика, теория возможностей, нечеткие уравнения, оценка распределения.*

ACM Classification Keywords*: G.1.7 – Ordinary Differential Equations. J.3 Life and Medical Sciences.*

Введение

Теоретико-вероятностные методы широко и успешно используются в научных исследованиях для моделирования в терминах случайности разных аспектов неопределенности, которая отображает неполноту знаний и их недостоверность.

Вместе с тем вероятностные методы оказались неэффективными при моделировании широкого класса процессов и явлений (социальных систем, субъективных суждений и так далее [Cobb,1981]). Неопределенность (нечеткость) в этих явлениях неадекватно моделируется вероятностными методами, поскольку не наблюдается многоразового повторения событий в одинаковых условиях, в то время как вероятностные модели предназначены для описания событий, которые повторяются многократно. В связи с этим, начиная с 1960-х годов были разработаны разные не вероятностные модели неопределенности и соответствующие теории: субъективная вероятность Северджа, теория возможности Заде и др. В 1974 году М. Сугено ввел понятие возможности события (альтернатива понятию вероятности в невероятностных моделях). После этого разными авторами были созданы варианты теории возможностей [Song, 2000], [Zadeh, 1978], [Бычков, 2007], [Пытьев, 1990]. Следует выделить теорию возможностей, построенную в [Пытьев, 1990], которая в отличие от других, строится по схеме, аналогичной теории вероятности.

В теории возможностей для события вместо вероятности предоставляется относительная оценка возможности ее появления, в шкале возможностей $[0,1]$. На основе числового значения возможности происходит сравнение событий (более возможное, менее возможное, равновозможные). Определенное свойство события считается теоретико-возможностным в случае его инвариантности относительно произвольного преобразования возможностной шкалы, которая хранит порядок на ее элементах. Лишь таким свойствам предоставляется содержательное толкование.

Для практического приложения теории возможностей к моделированию реальных процессов используют нечеткие дифференциальные уравнения. Известные подходы к нечеткому моделированию рассматривают такие уравнения, как дифференциальные уравнения с нечеткими параметрами [Song, 2000], [Zadeh, 1978], [Пытьев, 1990]. Главным недостатком этих подходов является то, что нечеткость моделирует лишь погрешности в вычислении параметров, в то время как для приложений важным является моделирование неопределенности в возмущении правой части уравнения. Из-за этого часто используют стохастические дифференциальные уравнения даже в случаях их неадекватности изучаемому процессу.

В данной работе разработаны конструктивные методы нахождения распределения решения для нового класса нечетких дифференциальных уравнений, лишенных отмеченных выше недостатков, а также приведен пример применения полученных результатов.

Основной результат

Для практического применения важными являются дифференциальные уравнения вида $y' = f(t,y) + g(t,y)\upsilon(t)$, где $\upsilon(t)$ – процесс в котором сосредоточена определенная неопределенность (“шум”). Для случая вероятностной природы неопределенности такие уравнения формализируются как стохастические.

Выполним формализацию и нахождение методов решения такого вида уравнений для случая, когда неопределенность в $\upsilon(t)$ имеет теоретико-возможностную природу, т.е. является нечеткостью.

Приведем определение основных понятий теории возможностей из [Бычков, 2007]. Пусть X – непустое множество (пространство элементарных событий), $\mathbf{A}$ – класс подмножеств X, который содержит $\varnothing, X$. Множества класса $\mathbf{A}$ будут интерпретироваться как (нечеткие) события.

Обозначим $L=[0,1]$ – шкалу возможностей. Ее элементы характеризуют степень возможности события. Для предоставления значения возможности события используется мера возможности.

Определение 1. Мерой возможности на $\mathbf{A}$ называется функция $P:\mathbf{A}\to L$, которая удовлетворяет условию: если $\{A_t \mid t\in T\}$ – семейство множеств из $\mathbf{A}$ таких, что $\bigcup\limits_{t\in T} A_t \in \mathbf{A}$, то $P(\bigcup\limits_{t\in T} A_t)=\sup\limits_{t\in T} P(A_t)$.

Мера возможности P называется нормируемой, если $P(X)=1,\ P(\varnothing)=0$.

В дальнейшем все меры возможности будут считаться нормируемыми.

Определение 2. P-моделью теории возможностей называется тройка $(X,\mathbf{A},P)$, где P является мерой возможности на $\mathbf{A}$.

В работе [Бычков, 2007] доказано, что мера возможности может быть продолжена с алгебры множеств на булеан пространства элементарных событий, потому без ограничения общности будем рассматривать P-модель теории возможностей вида $(X,2^X,P)$. В этом случае [Бычков, 2007], мера возможности P может быть представлена в виде $P(A)=\sup\limits_{x\in A} f(x)$, для некоторой функции $f:X\to L$.

Введем обозначение $X_\varepsilon=\{x\in X \mid P\{x\}>\varepsilon\}$ – множество элементарных событий, возможность которых превышает заданный уровень ε.

События возможности нуль будем считать невозможным. Все они являются подмножествами дополнения множества X_0.

Запись $P\{E(x)\}$, где E – определенный предикат, будем использовать в качестве сокращения для записи $P\{x\in X \mid E(x)\}$.

Определение 3. Нечеткой величиной (скалярной или векторной) называется функция вида $\xi: X\to \mathbf{R}^n$, для которой $\mathrm{Dom}\,\xi \supseteq X_0$.

Определение 4. Нечеткие величины ξ_i, $i=1,\dots,n$, называются независимыми в совокупности, если

$$\forall y_1,..y_n : P\{\xi_i=y_i, i=1..n\}=\min_{i=1..n} P\{\xi_i=y_i\}.$$

Определение 5. Нечетким процессом (с непрерывным временем) называется функция вида $p(t,x):\mathbf{T}\times X\to\mathbf{R}^n$, для которой $\mathrm{Dom}\,p\supseteq \mathbf{T}\times X_0$, где $\mathbf{T}=[0,+\infty)$.

Иногда второй аргумент в записи нечеткого процесса опускается.

Нечеткие процессы являются основными для моделирования реальных процессов с точки зрения теории возможностей. Базовым примером нечеткого процесса, который будет использован в данной работе, есть аналог винеровского процесса: процесс нечеткого блуждания [Бычков, 2005]:

Определение 6. Нечеткий процесс (скалярный или векторный) $\mathbf{w}(t,x)$ называется процессом нечеткого блуждания (ПНБ), если:

1. для любых моментов времени $0 \le t_1 < t_2 .. < t_n < t_{n+1}$ нечеткие величины

$\mathbf{w}(t_{i+1}) - \mathbf{w}(t_i)$,

$i = 1,\dots,n$ независимые в совокупности (независимость приращений);

2. переходная возможность процесса имеет вид:

$$\forall t > t_0 > 0 \, \forall y \in \mathbf{R}^n : P\{\mathbf{w}(t) - \mathbf{w}(t_0) = y\} = \varphi\left(\frac{\left\|\Xi^{-1/2} y\right\|^2}{(t - t_0)^2}\right),$$

где (Ξ – положительно определенная матрица, $\varphi : [0,+\infty) \to L$ – убывающая непрерывная функция, такая, что $\lim\limits_{x \to +\infty} \varphi(x) = 0$ и $\varphi(0) = 1$;

3. $\mathbf{w}(0,x) = \mathbf{0}, \forall x \in X_0$.

Функция $\varphi(x)$ из этого определения называется функцией распределения ПНБ.

Скалярный ПНБ в тексте будет обозначаться как $w(t,x)$.

Приведем некоторые свойства ПНБ, которые будут использованы далее.

Теорема 1. Существует P-модель $(X^1, 2^{X^1}, P^1)$ и нечеткий процесс $p(t,x)$ на ней, которые удовлетворяют условиям:

1) события $p(t_i,x), i = 1,..,n$ независимы при $t_i \ne t_j (i \ne j)$ и $P^1\{p(t_i) = y_i\} = \varphi(y_i^2), \quad i = 1,\dots,n$;

2) $p(t,x)$ – ограниченная измеримая функция для каждого фиксированного $x \in Y_0$ и произвольная ограниченная измеримая функция на $[0,T]$ является траекторией p.

3) $w^1(t,x) = \int_0^t p(\tau,x) d\tau$ является процессом нечеткого блуждания

Доказательство. Пусть X^1 - множество всех ограниченных измеримых функций вида $[0,T]\mapsto \mathbf{R}$ и $P^1(A)=\sup\limits_{x(\cdot)\in A}\varphi(\sup\limits_{t\in[0,T]}\left|x(t)\right|^2), A\subseteq X^1$ - мера возможности на этом множестве.

Докажем, что процесс, определенный как $p(t,x)=x(t)$ является искомым, то есть удовлетворяет свойствам 1-3.

1) $P^1\{p(t_0,x)=y\}=P^1\{x(t_0)=y\}=\varphi(y^2)$.

Покажем независимость. Имеем

$$P^1\{p(t_i,x)=a_i, i=1,\dots,n\}=P^1\{x(t_i)=a_i, i=1,\dots,n\}=$$

$$=\varphi(\max_{i=1..n}(a_i^2))=\min_{i=1..n}\varphi(a_i^2)=\min_{i=1..n}P^1\{x(t_i)=a_i\}=\min_{i=1..n}P^1\{p(t_i,x)=a_i\}.$$

2) Это свойство очевидно.

3) Покажем независимость приращений процесса $w^1(t,x)$:

$$P^1\{w^1(t_{i+1},x)-w^1(t_i,x)=a_i, i=1,\dots,\mathrm{n}\}=P^1\{\int_{t_i}^{t_{i+1}}x(t)dt=a_i\}=\varphi(k^2),$$

где k является максимальным значением целевого функционала для оптимизационной задачи $\max\left|x(t)\right|\to\min$, при условиях $\int_{t_i}^{t_{i+1}}x(t)dt=a_i, i=1,\dots,n$. Поскольку промежутки (t_i,t_{i+1}) непересекающиеся, то решение этой задачи получается склеиванием решений задач $\max\left|x_i(t)\right|\to\min$ при условиях $\int_{t_i}^{t_{i+1}}x_i(t)dt=a_i$ и таким образом:

$$x^{opt}(t)=\begin{cases}x_i^{opt}(t), t\in[t_i,t_{i+1}],\\ 0, \quad t\notin[t_i,t_{i+1}],\end{cases}$$

$$\varphi(k^2)=\min_{i=1,\dots,n}P^1\{\int_{t_i}^{t_{i+1}}x(t)dt=a_i\}=\min_{i=1,\dots,n}P^1\{w^1(t_{i+1},x)-w^1(t_i,x)=a_i\}.$$

Решением i-ой задачи является функция почти везде константа $a_i/\Delta t_i$, что следует из неравенства

$$\left|a_i\right|=\left|\int_{t_i}^{t_{i+1}}x_i(t)dt\right|\le\int_{t_i}^{t_{i+1}}\left|x_i(t)\right|dt\le\max\left|x_i(t)\right|\Delta t_i.$$

Таким образом, выполняется второе условие из определения ПНБ:

$$P^1\{w^1(t,x)-w^1(t_0,x)=a\}=\varphi(a^2/(t-t_0)^2).$$

Наконец третье условие из определения ПНБ $w^1(0,x)=0$ тоже, очевидно, выполняется. Теорема доказана.

Лемма 1. Если $\mathbf{w}(t,x)$ - ПНБ, то $\forall x\in X_0$ траектория $\mathbf{w}(t,x)$ есть почти везде дифференцируемой.

Доказательство. Рассмотрим цепочку неравенств

$$P\{\|\mathbf{w}(t_1,x)-\mathbf{w}(t_2,x)\|\ge C|t_1-t_2|\}=\sup\left\{P\{\mathbf{w}(t_1,x)-\mathbf{w}(t_2,x)=\mathbf{a}\}:\|\mathbf{a}\|\ge C|t_1-t_2|\right\}=$$

$$=\sup\left\{\varphi\left(\frac{\|\Xi^{-1/2}a\|^2}{(t_1-t_2)^2}\right):\|a\|\ge C|t_1-t_2|\right\}\le\varphi\left(\frac{\lambda_m^2C^2(t_1-t_2)^2}{(t_1-t_2)^2}\right)=\varphi(\lambda_m^2C^2),$$

где λ_m – наименьшее собственное число матрицы $\Xi^{-1/2}$. То есть выполняется

$$P\{\exists t_1\ne t_2:\|\mathbf{w}(t_1,x)-\mathbf{w}(t_2,x)\|\ge C|t_1-t_2|\}\le\varphi(\lambda_m^2C^2).$$

Получили, что имеет место следующее условие Липшица

$$\forall\varepsilon>0\exists C_\varepsilon>0:\forall x\in X_\varepsilon,t_1,t_2:\|\mathbf{w}(t_1,x)-\mathbf{w}(t_2,x)\|<C_\varepsilon|t_1-t_2|.$$

Тогда по теореме Радемахера, $\forall x\in X_0$ функция $\mathbf{w}(t,x)$ является почти везде дифференцируемой. Лемма доказана.

Следствие 1. Функцию $\mathbf{w}'(t,x)$ можно продолжить до нечеткого процесса, равномерно ограниченного на каждом из множеств $R\times X_\varepsilon,\varepsilon>0$, поскольку $\forall x\in X_\varepsilon:\|\mathbf{w}'(t,x)\|\le\sqrt{\varphi^{-1}(\varepsilon)}/\lambda_m$.

Лемма 1 придает содержательный смысл интегралу $\int_0^T f(\tau)w'(\tau,x)d\tau$ от ограниченной на $[0,T]$ измеримой функции f . Если его понимать как интеграл Лебега при каждом $x\in X_0$, то функция $\xi(x)=\int_0^T f(\tau)dw(\tau,x)$ является нечеткой величиной. Аналогично можно придать смысл интегралу по векторному ПНБ:

$$\xi(x)=\int_0^T\mathbf{F}(\tau,x)d\mathbf{w}(\tau,x),\mathbf{F}(t)\in\mathbf{R}^{n\times m},\mathbf{w}(\tau,x)\in\mathbf{R}^m.$$

Теперь есть возможность строго сформулировать определение уравнения, зависящего от процесса нечеткого блуждания.

Рассмотрим следующее интегральное уравнение относительно функции $y(t,x)$ вида $[0,T]\times X\rightarrow\mathbf{R}$:

$$y(t,x)=y_0+\int_0^t f\left(\tau,y\left(\tau,x\right)\right)d\tau+\int_0^t g\left(\tau,y\left(\tau,x\right)\right)dw\left(\tau,x\right), \tag{1}$$

где $t\in[0,T]$, $w(\tau,x)$ – ПНБ, известные функции $f(t,y),g(t,y)$ имеют вид $[0,T]\times\mathbf{R}\rightarrow\mathbf{R}$, y_0 – известное значение.

Допустимо, что f и g является ограниченными, измеримыми при каждом значении y и выполняются условия Липшица:

$$\begin{gathered}\exists L>0\ \forall t\in[0,T]\ \left|f(t,y_1)-f(t,y_2)\right|\le L\left|y_1-y_2\right|,\\ \left|g(t,y_1)-g(t,y_2)\right|\le L\left|y_1-y_2\right|.\end{gathered} \tag{2}$$

Тогда $\forall x\in X_\varepsilon$ почти везде на $\left[0,T\right]$ выполняется неравенство:

$$\left|f(t,y_1)-f(t,y_2)+(g(t,y_2)-g(t,y_2))w'(t,x)\right|\le L\left(1+\sqrt{\varphi^{-1}(\varepsilon)}\right)\left|y_1-y_2\right|.$$

По теореме Каратеодори о существовании решения для дифференциальных уравнений, для каждого $x\in X_0$ существует единственная абсолютно непрерывная траектория $y(t,x)$, $t\in[0,T]$, которая удовлетворяет уравнению (1).

Таким образом, при указанных условиях, уравнение (1) определяет единственным образом нечеткий процесс $y(t,x)$, который будем называть решением уравнения. Дифференциальная форма записи уравнения (1) имеет вид:

$$dy(t,x)=f(t,y(t,x))+g(t,y(t,x))dw(t,x).$$

Аналогично дается смысл уравнению с векторным ПНБ

$$\mathbf{y}(t,x) = \mathbf{y}_0 + \int_0^t \mathbf{f}(\tau, \mathbf{y}(\tau,x))d\tau + \int_0^t \mathbf{G}(\tau, \mathbf{y}(\tau,x))d\mathbf{w}(\tau,x),\ t \in [0,T]. \quad (3)$$

Важной для приложений характеристикой процесса-решения сформулированного уравнения (1) или (3) есть множество $Y(t,\varepsilon)$ значений решения $y(t)$ в момент t, возможность которых больше заданного порога $\varepsilon > 0$, то есть $Y(t,\varepsilon) = \{y : P\{y(t,x) = y\} > \varepsilon\}$.

Заданное параметрическое семейство множеств дает нечеткий аналог распределению решения стохастических дифференциальных уравнений.

Будем искать конструктивный метод оценивания множеств $Y(t,\varepsilon)$ для этого уравнения.

Пусть $w(t,x)$ – скалярный ПНБ. Введем обозначение

$$p(t,x) = \begin{cases} w'(t,x), \text{ если определено и } x \in X_0, \\ 0, \text{ иначе.} \end{cases}$$

$\varphi(x)$ – функция распределения ПНБ $w(t,x)$.

Обозначим через E следующий шар ограниченных измеримых функций вида:

$$[0,T] \to \mathbf{R} : E(r) = \{u \in E : \|u\|_\infty < r\}.$$

Определим на E функционал:

$$P[u] = P\{x : p(\cdot,x) \equiv u(\cdot)\}.$$

Лемма 2. Имеют место следующие свойства:

1) $\forall u \in E : P[u] \le \varphi(\|u\|_\infty^2)$;

2) если $0 \le t_1 < t_2 < .. < t_{n+1} \le T$, то

$$\sup\left\{ P[u] | u \in E : \int_{t_i}^{t_{i+1}} u(\tau)d\tau = a_i \Delta t_i, i = 1,\ldots,n \right\} = \varphi\left(\max_{i=1,\ldots,n} a_i^2\right).$$

Доказательство. 1) Из леммы 1 следует, что $P\{x\} > \varepsilon \Rightarrow \|p(t,x)\|_\infty \le \sqrt{\varphi^{-1}(\varepsilon)}$, переходя к inf в неравенстве и учитывая непрерывность функции φ^{-1}, получаем $\|p(t,x)\|_\infty \le \sqrt{\varphi^{-1}(P\{x\})}$, откуда $P\{x\} \le \varphi(\|p(t,x)\|_\infty^2)$.

Тогда выполняется неравенство $P[u] = \sup\limits_{x:p(t,x)\equiv u(t)} P\{x\} \le \varphi(\|u\|_\infty^2)$.

$$2)\ P\{w(t_{i+1}) - w(t_i) = a_i\} = \min_{i=1,\dots,n} P\{w(t_{i+1}) - w(t_i) = a_i\} = \min \varphi(a_i^2) = \varphi(\max a_i^2).$$

$$\varphi(\max a_i^2) = P\{\int_{t_i}^{t_{i+1}} p(\tau,x)\,d\tau = a_i\Delta t_i, i = 1,\dots,n\} =$$

$$= \sup\{P\{p(\cdot,x) \equiv u\} \mid u \in E, \int_{t_i}^{t_{i+1}} u(\tau)\,d\tau = a_i\Delta t_i, i = 1,\dots,n\} =$$

$$= \sup\{P[u] \mid u \in E : \int_{t_i}^{t_{i+1}} u(\tau)\,d\tau = a_i\Delta t_i, \forall i\}.$$

Лемма доказана.

Лемма 3. Если $\varepsilon \in (0,1)$, то для каждой $u^* \in E(\sqrt{\varphi^{-1}(\varepsilon)})$ существует последовательность $u_n \in E$ такая, что $P[u_n] > \varepsilon$ и $\forall f \in R[0,T]$ $\int_0^t f(\tau)u_n(\tau)\,d\tau \to \int_0^t f(\tau)u^*(\tau)\,d\tau$ равномерно по t на отрезке $[0,T]$.

Доказательство. Выберем произвольную функцию $u^* \in E : \varphi(\|u^*\|_\infty^2) > \varepsilon$.

Положим $\{0 = t_1^n < t_2^n < ..t_{n+1}^n = T\}_{n\ge 1}$ – последовательность разбиений отрезка $[0,T]$ такая, что $\max \Delta t_i^n \to 0, n \to \infty$.

Положим в пункте 2 леммы 2 $t_i = t_i^n$, $a_i = \frac{1}{\Delta t_i^n}\int_{t_i^n}^{t_{i+1}^n} u^*(\tau)\,d\tau$ и получим, что

$$\sup\{P[u] \mid \int_{t_i^n}^{t_{i+1}^n} u(\tau)\,d\tau = \int_{t_i^n}^{t_{i+1}^n} u^*(\tau)\,d\tau, \forall i\} = \varphi(\max_{i=1..n} \frac{1}{\Delta t_i^n}\left|\int_{t_i^n}^{t_{i+1}^n} u^*(\tau)\,d\tau\right|)^2 \geq \varphi(\left\|u^*\right\|_\infty^2) > \varepsilon\,.$$

Поэтому существует последовательность $u_n \in E$, такая, что $P[u_n] > \varepsilon$ и

$$\int_{t_i^n}^{t_{i+1}^n} u_n(\tau)\,d\tau = \int_{t_i^n}^{t_{i+1}^n} u^*(\tau)\,d\tau, i = 1,\ldots,n\,.$$

Выберем функцию, интегрируемую по Риману на $[0,T]$ и докажем сходимость, которая следует из условия леммы. Обозначим через $\chi_t(\tau)$ – индикатор $[0,t]$.

$$\left|\int_{t_i^n}^{t_{i+1}^n} f(\tau)\big(u_n(\tau) - u^*(\tau)\big)\,d\tau\right| \leq$$

$$\leq \left|\int_{t_i^n}^{t_{i+1}^n} f(t_i^n)\big(u_n(\tau) - u^*(\tau)\big)\,d\tau\right| + \int_{t_i^n}^{t_{i+1}^n} \omega\big(f,\big[t_i^n, t_{i+1}^n\big]\big)\big|u_n(\tau) - u^*(\tau)\big|\,d\tau \leq$$

$$\leq 2\sqrt{\varphi^{-1}(\varepsilon)}\,\omega(f,[t_i^n, t_{i+1}^n])\Delta t_i^n\,,$$

где ω – колебание функции на промежутке.

Тогда

$$\left|\int_0^t f(\tau)\big(u_n(\tau) - u^*(\tau)\big)\,d\tau\right| \leq \sum_{i=1}^n \left|\int_{t_i^n}^{t_{i+1}^n} \chi_t(\tau) f(\tau)\big(u_n(\tau) - u^*(\tau)\big)\,d\tau\right| \leq$$

$$\leq \sum_{i=1}^n \left|\int_{t_i^n}^{t_{i+1}^n} f(\tau)\big(u_n(\tau) - u^*(\tau)\big)\,d\tau\right| + \left|\int_{t_{k(n,i)}^n}^{t_{k(n,i)+1}^n} \chi_t(\tau) f(\tau)\big(u_n(\tau) - u^*(\tau)\big)\,d\tau\right| \leq$$

$$\le 2\sqrt{\varphi^{-1}(\varepsilon)}\sum_{i=1}^{n}\omega\left(f,\left[t_i^n,t_{i+1}^n\right]\right)\Delta t_i^n+\|f\|_\infty \max_{i=1,\dots,n}\left|\Delta t_i^n\right|\to 0\,,$$

где $k(n,t)$ – номер отрезка, на котором $\chi_t(\tau)$ изменяет значение с 1 на 0. Первое слагаемое стремится к нулю как разница сумм Дарбу для функции f. Сходимость не зависит от t, а потому равномерная. Лемма доказана.

Непосредственно из определения $Y(T,\varepsilon)$ и мер возможности следует равенство $Y(T,\varepsilon)=\{y(T,x)\,|\,x\in X_\varepsilon\}$ – множество достижимых состояний за время T траекториями процесса - решениями уравнения (1), возможность которых больше ε.

Свяжем с уравнением (1) или (3) уравнение с ограниченным управлением $u(t)$:

$$z(t)=y_0+\int_0^t f(\tau,z(\tau))d\tau+\int_0^t g(\tau,z(\tau))u(\tau)d\tau\,. \tag{4}$$

Пусть функции удовлетворяют условия (2) и кроме того, для каждого фиксированного y, функция $g(t,y)$ интегруема по Риману на промежутке $[0,T]$.

Обозначим $U(T,\varepsilon)=\{z(T,u)\,|\,u\in E(\sqrt{\varphi^{-1}(\varepsilon)})\}$ – множество достижимости для (4) с начального состояния y_0 с помощью управлений $u(t)$, которые удовлетворяют условию $u\in E(\sqrt{\varphi^{-1}(\varepsilon)})$.

Заметим, что из леммы 1 следует включение $Y(t,\varepsilon)\subseteq U(t,\varepsilon),\varepsilon\in(0,1)$.

Теорема 2. Для $\varepsilon\in(0,1)$ множество $Y(T,\varepsilon)$ плотно в множестве $U(T,\varepsilon)$.

Доказательство. Пусть точка z^+ достигается во время T с помощью управления $u^*(t)$, причем $\|u^*\|_\infty<\sqrt{\varphi^{-1}(\varepsilon)}$. Соответствующую траекторию обозначим $z^*(t)$.

Пусть $t_s\in\mathbf{R}^+\setminus\bigcup_{q\in Q}D_q$, D_q – множество точек разрыва $g(\cdot,q)$, которое имеет меру нуль в силу предположения об интегрируемости по Риману. Тогда

$$\left|g(t,z^*(t))-g(t_s,z(t_s))\right|\le$$

$$\le\left|g(t,z^*(t))-g(t,q)+g(t,q)-g(t_s,q)+g(t_s,q)-g(t_s,z(t_s))\right|\le$$

$$\le L\left|z^*(t)-q\right|+\left|g(t,q)-g(t_s,q)\right|+L\left|q-z^*(t_s)\right|.$$

Поскольку функция z^* непрерывна, то $\overline{\lim_{t\to t_s}}\left|g(t,z^*(t))-g(t_s,z(t_s))\right|\le 2L\left|q-z^*(t_s)\right|$. В силу произвольности $q\in\mathbf{Q}$, t_s является точкой непрерывности функции $g(t,z^*(t))$. Отсюда следует, что множество точек разрыва $g(t,z^*(t))$ имеет меру нуль и эта (ограниченная) функция является интегруемой по Риману.

Выберем $\delta>0$. По лемме 3, существует $x\in X_\varepsilon$, такое что для $u_0(t)\equiv p(t,x)$ выполняется

$$\left|\int_0^t g\left(\tau,z^*(\tau)\right)\left(u^*(\tau)-u_0(\tau)\right)d\tau\right|<\delta, t\in[0,T].$$

Пусть $z_0(t)$ – решение интегрального уравнения

$$z_0(t)=y_0+\int_0^t f\left(\tau,z_0(\tau)\right)d\tau+\int_0^t g\left(\tau,z_0(\tau)\right)u_0(\tau)d\tau.$$

Обозначим $h(t)=z^*(t)-z_0(t)$. Тогда из условий (2) следует неравенство

$$\left|h(t)\right|\le\int_0^t\left|f\left(\tau,z_0(\tau)+h(\tau)\right)-f\left(\tau,z_0(\tau)\right)\right|d\tau+\delta\le\int_0^t Lh(\tau)d\tau+\delta,$$

Откуда $\left|z_0(T)-z^+\right|=\left|h(T)\right|\le\delta\exp(LT)\to 0,\delta\to 0$. Следовательно, множество $Y(T,\varepsilon)$ является плотным в $U(T,\varepsilon)$. Теорема доказана.

Заметим, что для ПНБ $w^1(t,x)$ из теоремы 1 выполняется равенство $Y(T,\varepsilon)=U(T,\varepsilon),\varepsilon\in(0,1)$.

Теоремы 1 и 2 легко обобщаются на случай векторных ПНБ.

Таким образом, для получения практически значимой информации о процессе-решении уравнения, целесообразным является нахождение множества $U(t,\varepsilon)$ вместо $Y(t,\varepsilon)$, что является задачей теории управления.

Множества $U(t,\varepsilon)$ будем называть оценками *ε-среза* решения уравнения.

Поиск $U(t,\varepsilon)$ в одномерном случае может быть осуществлен на основе следующей простой леммы [Cobb,1981]:

Лемма 4. Пусть функция u удовлетворяет условию $\forall t\in[0,T]:|u(t)|<C$, функции $y_1, y_2, y\in C^1[0,T]$ и удовлетворяют таким условиям:

$$y_1' = f(t,y_1) - C|g(t,y_1)|,\ y_1(0)=y_0;$$

$$y_2' = f(t,y_2) + C|g(t,y_2)|,\ y_2(0)=y_0;$$

$$y' = f(t,y) + g(t,y)u(t),\ y(0)=y_0.$$

Тогда $\forall t\in[0,T]: y_1(t)\le y(t)\le y_2(t)$.

Следствие 1. Множество $U(T,\varepsilon)$ для уравнения (1) может быть представлено в виде $\{y\mid y_1(t)<y<y_2(t), t\in[0,T]\}$, где y_1, y_2 – функции из леммы 4, полученные при значении $C=\sqrt{\varphi^{-1}(\varepsilon)}$.

В многомерном случае для поиска множеств достижимости можно использовать, например, метод динамического программирования Беллмана.

Применение полученных результатов

Эпидемическая модель Росса строится исходя из нижеприведенных предположений [Medlock, 2008]. Популяция состоит из группы риска $S(t)$ и инфицированных индивидов $I(t)$, причем

- размер популяции N большой и постоянный;
- не учитываются выздоровление, отставание; перемешивание равномерное;
- скорость заболевания пропорциональна количеству инфицированных.

Данная модель описывается уравнениями:

$$S(t)=N-I(t),\ I'(t)=aI(t)(N-I(t)). \tag{5}$$

Обозначим через $y(t)=I(t)/N$ – часть инфицированных.

Модель (5) может быть уточнена путем учета выздоровления ($b > 0$) и передачи заболевания из постороннего источника ($c > 0$)[Cobb,1981]:

$$y'(t) = ay(t)(1 - y(t)) - by(t) + c(1 - y(t)). \tag{6}$$

Здесь $a > 0$ – скорость передачи заболевания между индивидами.

Внесем в уравнение (6) нечеткую поправку, призванную компенсировать возможную неточность модели (здесь $w(t)$ – ПНБ):

$$y'(t) = ay(1 - y) - by + c(1 - y) + \sigma(y)w'(t). \tag{7}$$

Сделаем предположение ([Cobb,1981]) о том, что функция $\sigma(y)$ принимает максимальное значение при $y = 1/2$ и минимальное значение при $y \in \{0,1\}$. В качестве $\sigma(y)$ можно взять $\delta y(1 - y)$, $\delta > 0$.

Согласно следствия из леммы 4, для уравнения (7) оценка ε -среза решения является областью расширенной фазовой плоскости

$$\{(t,y) : t > 0, y_1(t) < y < y_2(t)\},$$

где

$$y'_{12}(t) = y_{12}(1 - y_{12})(\mathrm{a} \pm \delta\sqrt{\varphi^{-1}(\varepsilon)}) - by_{12} + c(1 - y_{12}), \tag{8}$$

Причем начальные условия имеют вид $y_1(0) = y_2(0) = y_0$.

Неотрицательные стационарные решения уравнений (8) при $C_\varepsilon = \sqrt{\varphi^{-1}(\varepsilon)}$ имеют вид:

$$z_1 = \frac{a - b - c + \delta C_\varepsilon + \sqrt{\left(a - b + c + \delta C_\varepsilon\right)^2 + 4bc}}{2\left(a + \delta C_\varepsilon\right)},$$

$$z_2 = \frac{a - b - c - \delta C_\varepsilon + \sqrt{\left(a - b + c - \delta C_\varepsilon\right)^2 + 4bc}}{2\left(a - \delta C_\varepsilon\right)}.$$

Таким образом, можно принять, что часть больных в популяции при больших t, с возможностью большей ε, лежит в промежутке $[z_1(\varepsilon), z_2(\varepsilon)]$ и с меньшей, чем ε, лежит вне этого промежутка.

Пусть $\varphi(x) = \exp(-x)$ $a = 1$, $b = 0.4$, $c = 0.01$, $\delta = 0.1$.

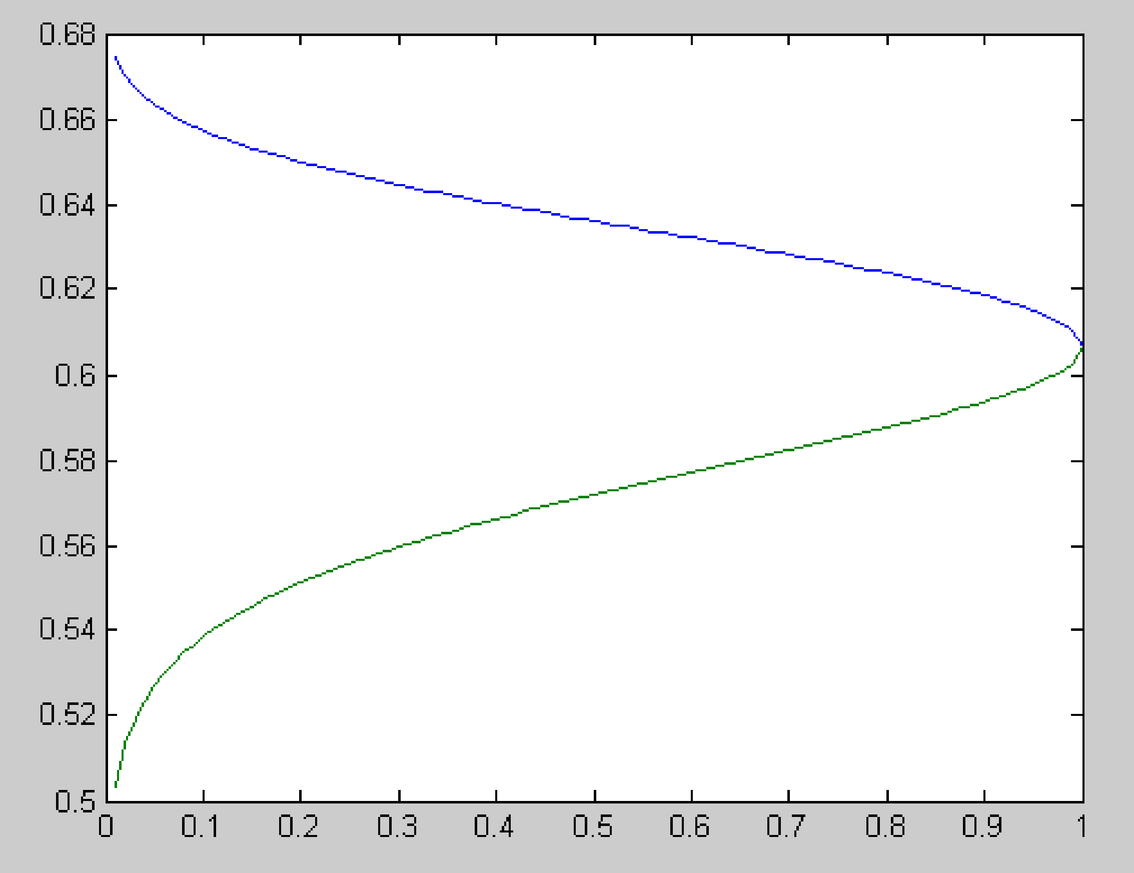


Рис. 1.

На рис. 1 показаны пределы, в которых лежит часть больных с уровнем возможности больше ε (параметр ε на горизонтальной оси, количество больных – на вертикальной).

Заключение

В работе проведена формализация нового класса нечетких дифференциальных уравнений и разработан метод для их решения. Этот метод может стать полезным при моделировании широкого круга реальных процессов и явлений, неопределенность в которых имеет невероятностную природу. Приведен пример приложения приведенной теории к моделированию динамики развития эпидемии.

Литература

[Cobb,1981] L.Cobb. Sochastic differential equations for the social sciences // Mathematical frontiers of social sciences, Westview Press, 1981. - p.37-68.

[Medlock, 2008] J.Medlock. Mathematical Modeling of Epidemics // http://www.amath.washington.edu/~medlock/other/epidemiology_intro_talk.pdf.

[Song, 2000] S.Song, C.Wu. Existence and uniqueness of solutions to Cauchy problem // Fuzzy Sets and Systems, 110, 2000. - p. 55-67.

[Zadeh, 1978] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility // Fuzzy Sets and Systems, 1978. Vol. 1, pp. 3-28.

[Бычков, 2005] А.С. Бычков. Построение интеграла по процессу нечеткого блуждания // Вестник Киевского университета, Серия: физико-математические науки, 2005. №4, с. 125-133.

[Бычков, 2007] А.С. Бычков, К.С.Колесников. Построение (PN) –модели теории возможностей // Вестник Киевского университета, Серия: физико-математические науки, 2007. №1, с.134-138.

[Пытьев, 1990] Ю.Пытьев. Возможность. Элементы теории и применение. УРСС,1990. -190 С.

Authors' Information

Алексей Бычков *– к.ф.-м.н., заведующий кафедрой программирования и компьютерной техники факультета информационных технологий Киевского национального университета имени Тараса Шевченка; ул. Ломоносова 81А, 03022, Киев, Украина; e-mail: bos.knu@gmail.com*

Основная область научных интересов: исследование гибридных автоматов как моделей непрерыво-дискретных процессов; построение согласованной теории возможностей, нечетких персептивных величин и процессов; математические основы моделирования нечетких сложных систем; применение математических методов в биологии, медицине и экономике

Евгений Иванов *– к.ф.-м.н., ассистент кафедры программирования и компьютерной техники факультета информационных технологий Киевского национального университета имени Тараса Шевченка; ул. Ломоносова 81А, 03022, Киев, Украина; e-mail: ivanov.eugen@gmail.com*

Основная область научных интересов: семантика языков программирования; формальные методы; математическая теория систем; гибридные (дискретно-неперервыei) системы

Ольга Супрун *– к.ф.-м.н., доцент кафедры программирования и компьютерной техники факультета информационных технологий Киевского национального университета имени Тараса Шевченка; ул. Ломоносова 81А, 03022, Киев, Украина; e-mail: o.n.suprun@gmail.com*

Основная область научных интересов: математическое моделирование и вычислительные методы; нечеткие величины и процессы; гибридные модели непрерывно-дискретных процессов

About estimate of fuzzy differential equations distribution

Alexei Bychkov, Eugene Ivanov, Olha Suprun

Abstract*: In the article the methods of estimation of solution's distribution for a new class of uncertain differential equations which contain a possibilistic process in right part are developed. The example of the use of developed methods is given.*

Keywords*: fuzzy logic, theory of possibility, fuzzy equation, estimate of distribution.*

AUTOMATIZATION OF COMPUTER BUSINESS GAME AUTOMATON MODEL CONSTRUCTION

Olga Vikentyeva, Alexander Deryabin, Dmitrij Kozhevnikov, Lidiia Shestakova

Abstract: *This paper suggests an approach, which enables automatization of computer business game automaton model construction. The processes of business game design and conduction occur within the Competence-based Business Game Studio source environment. This fact provides the universality of the domain. Separation of information system into operating and automaton models causes control logic to be concentrated within the automaton model. Due to this fact the control object (operating model) is to have unsophisticated behavior: accepting commands from the automaton and then executing stated commands. This paper verifies the need for automaton model, provides analysis of requirements and describes the design of corresponding program module (interactive visual model editor) of the Competence-based Business Game Studio source environment. This module is used to construct and edit business process models interactively during the stage of business game design.*

Keywords: *competencies, active learning methods, business-game, business-process, control automat, operating automat, business process modeling.*

ACM Classification Keywords: *K.3 COMPUTERS AND EDUCATION: K.3.2 Computer and Information Science Education – Information systems education. I.2 ARTIFICIAL INTELLIGENCE: I.2.1 Applications and Expert Systems – Games.*

Conference topic: *Technology-Facilitated Learning in Complex Domains.*

Introduction

Educational technologies in professional sphere tend to leverage increasingly high number of active training methods. One of the most wide-spread methods is represented by business games (BG). A computer business game (CBG) differs from the conventional one by a considerably lower time required for training, as well as a significantly higher number of business simulations. Currently, CBGs are vastly employed in corporate training, higher and secondary education due to their ability to simulate real business cases and help building approaches to address professional problems. The relevance of this research is evidenced by numerous publications of both Russian and foreign authors [Biggs, 1990], [Draganidis, 2006], [Girev, 2010], [Vikentjeva, 2013], [Bazhenov, 2014].

At the moment the market of computer business games is represented by a large variety of software titles, SimulTrain being amongst the most known and sophisticated, for example.

Commonly, a computer business game is limited to a certain domain (project management, marketing, general management, financing, etc.), thus, making it highly specific. This paper suggest an approach, that enables designing and conducting business games based on the business processes of an organization, which makes the business game universal in regard to the domain. A formalized description of business game domain is acquired with the help of consecutive transformation of business processes models, which makes the automatization of business game design possible.

A project called “Competence-based Business Game Studio” (CBGS) [Vikentjeva, 2013] has been initiated to implement the model of production and management activity. This project is composed of several intertwined subsystems (including design subsystem) and serves the purpose of creating and assessing competences through business games, built in real business processes.

The design subsystem consists of many modules, including the one that automates the transformation of business process models into business game scenarios. This paper considers design and development of such program module.

Leveraging automaton model during business game design

Thus far many of the mathematical models related to automaton programming have found successful implementations across different domains. These models may vary in details but still have much in common.

Automaton models owe their diversity to the broad variety of implementation domains. The latter include mathematical linguistics, logic control, human behavior simulation, communication protocols, formal language, computability and computational complexity theories.

It is a common practice to distinguish control devices and control subjects. Following this concept, the system may be divided into following parts:

- control part (control system) is responsible for behavior logic: transition to the new state of the system, choice of actions to be executed (which depends on current state and incoming signals);
- controlled part (control subject) is responsible for execution of actions, determined by control system, and, possibly, for the creation of certain incoming signals for control part (feedback).

Therefore, the logic of system behavior is concentrated in the control automaton. Control subject is characterized by unsophisticated behavior, i.e. it does not process input signals from the outer environment, rather it merely receives commands from the control automaton. Each command stands for one and only action of the subject [Polikarpova, 2008].

Automaton model is described with the use of algorithm logic scheme language (ALS). A sequence of operators written down in the language of ALS implements and algorithm of business game control.

ALS expression may be represented as following: $L = \{ H, A, P, \omega, \uparrow, \downarrow, K \}$, where H – algorithm start operator, K – algorithm finish operator, P – conditional transition, A – controlling action, ω – unconditional transition, ↑ – transition start, ↓ – transition end.

Each operator of the ALS expression implies a command, interpretable by the automaton module (for instance, transition from one state of the business game to another, or conveying control signals from the operating model).

Automatization scheme of the automaton model acquisition may be seen on Fig. 1.

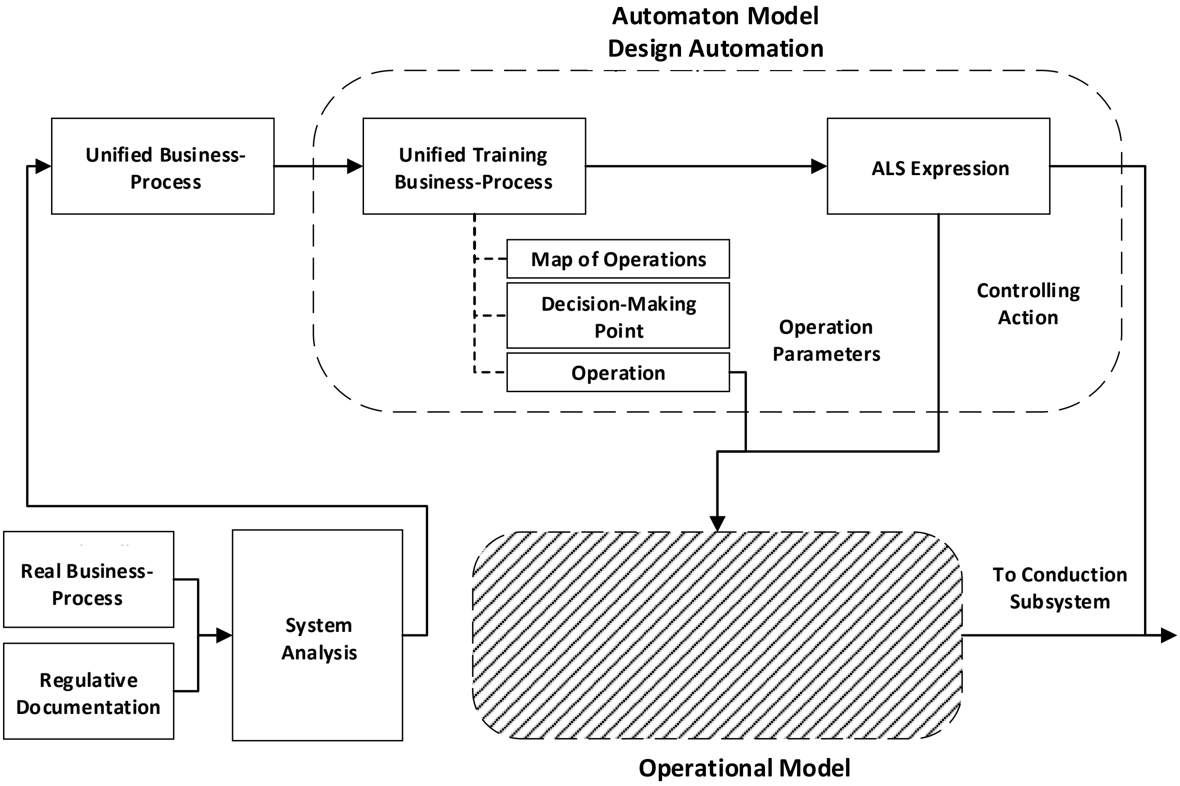


Figure 1. Automatization scheme of the automaton model acquisition

Input data for the model is represented by poorly-formalized data sources (graphical business process models, text descriptions, regulative documentation, etc.). During system analysis of the domain real business process model is transformed into unified business process model (UBP). The latter is comparatively more generalized, due to the fact, that process of UBP construction disregards business operations, specific only to certain companies. The correctness of the models is ensured by compliance with regulative documentation and industry standard business process models.

Unified business process may happen to be considerably sophisticated and include conditional and repetitive business operations besides sequential operations. To convert such model into sequential representation all the cyclic and conditional structures are suggested to be replaced with generalized blocks. Each of these blocks receives a separate Map of Operations of training UBP.

Unified business process requires to be transformed into unified training business process. This is needed to implement additional changes to the original UBP model, related to the process of competence establishment on the trainee's side.

Unified training business process description consists of three metamodels:

- Map of Operations (MO) represents a tree-like structure with all the possible sequences of business operations;
- Operation contains information about operation's parameters (inputs, outputs, controlling information, regulations, mechanisms, time limits, costs, etc.);
- Decision-Making Point (DMP) implements interaction with the player by enabling the latter to choose which operation should be performed next.

When the user makes a decision at this point, the system performs an operation and proceeds to the state, when the user is to choose next operation once again. The amount of possible BG states is finite, due to the fact that repetitive states are not allowed. I.e. once an operation is performed, it becomes unavailable for the user to repeat it.

The task of DMP number minimization is solved by business process decomposition. This enables breakdown of Map of Operations into complementing parts, which contain relatively small amount of Decision-Making Points. Therefore, the automaton model of the business game is going to be represented as a set of strings that store the algorithms of business game control in the form of ALS expressions.

The need for BG automaton model construction automatization is caused by following factors:

- Business game design leverages multimodel representation of business processes [Vikentjeva, 2015].

- Unified business process may happen to be fairly sophisticated, perplexing the derived models and impeding manual business game design.
- All the models are well-formalized, making the transition to automated process possible.

Thus, a contemporary CASE tool needs to be developed, allowing user to interactively build and edit UTBP models in real-time during business game design (interactive visual model editor).

Interactive Visual Model Editor Requirements Analysis

The process of business game design is considered in [Vikentjeva, 2015]. Analysis of TO BE model has enabled formulating a list of functional specifications. According to the latter the editor’s purpose is to provide transformation of UBP description into automaton model represented in the form of ALS.

Interactive visual model editor is to provide BG designer with following functionality:

- UBP model creation, utilizing a set of elements, defined in previously developed notation.
- UBP model creation occurs via “drag & drop” technique, or via selecting an element from the context menu, called by right clicking on the work area.
- Each element of the notation has a set of editable attributes.
- Resources of an operation can be set and modified on a separate work screen for operating model editing.
- Generated model complies with syntactic rules of the notation.
- Models can be modified, saved and loaded.
- Syntactically correct UBP model enables generation of UTBP model and grants access to Map of Operations metamodel.
- Generated UTBP models can be modified, saved and loaded.
- Decision-Making Point element can be edited.
- UTBP model is complying with the syntactic rules.
- Syntactically correct UTBP model enables generation of ALS expressions.
- ALS expressions can be modified, saved and loaded.
- Generated ALS expressions are syntactically correct.

Functional specification formulated above have enabled Use Case Diagram (Fig. 2) to be built as well as description of use cases and Activity Diagrams (Fig. 3) to be provided.

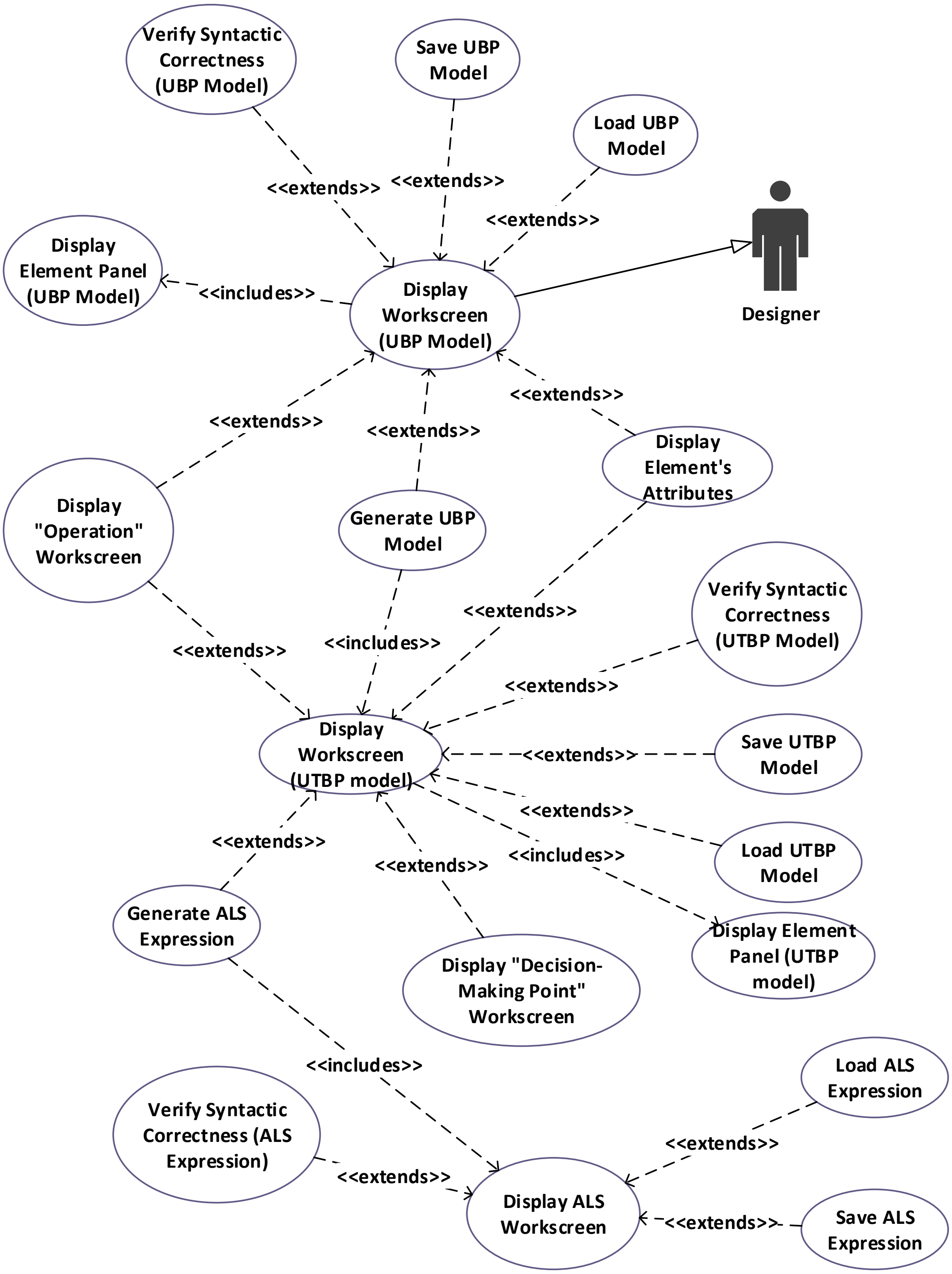

Verify Syntactic Correctness (UBP Model)
Save UBP Model
Load UBP Model
Display Element Panel (UBP Model)
Display Workscreen (UBP Model)
Designer
Display "Operation" Workscreen
Generate UBP Model
Display Element's Attributes
Verify Syntactic Correctness (UTBP Model)
Display Workscreen (UTBP model)
Save UTBP Model
Load UTBP Model
Display Element Panel (UTBP model)
Generate ALS Expression
Display "Decision-Making Point" Workscreen
Load ALS Expression
Verify Syntactic Correctness (ALS Expression)
Display ALS Workscreen
Save ALS Expression
<<extends>>
<<includes>>

Figure 2. Use Case Diagram

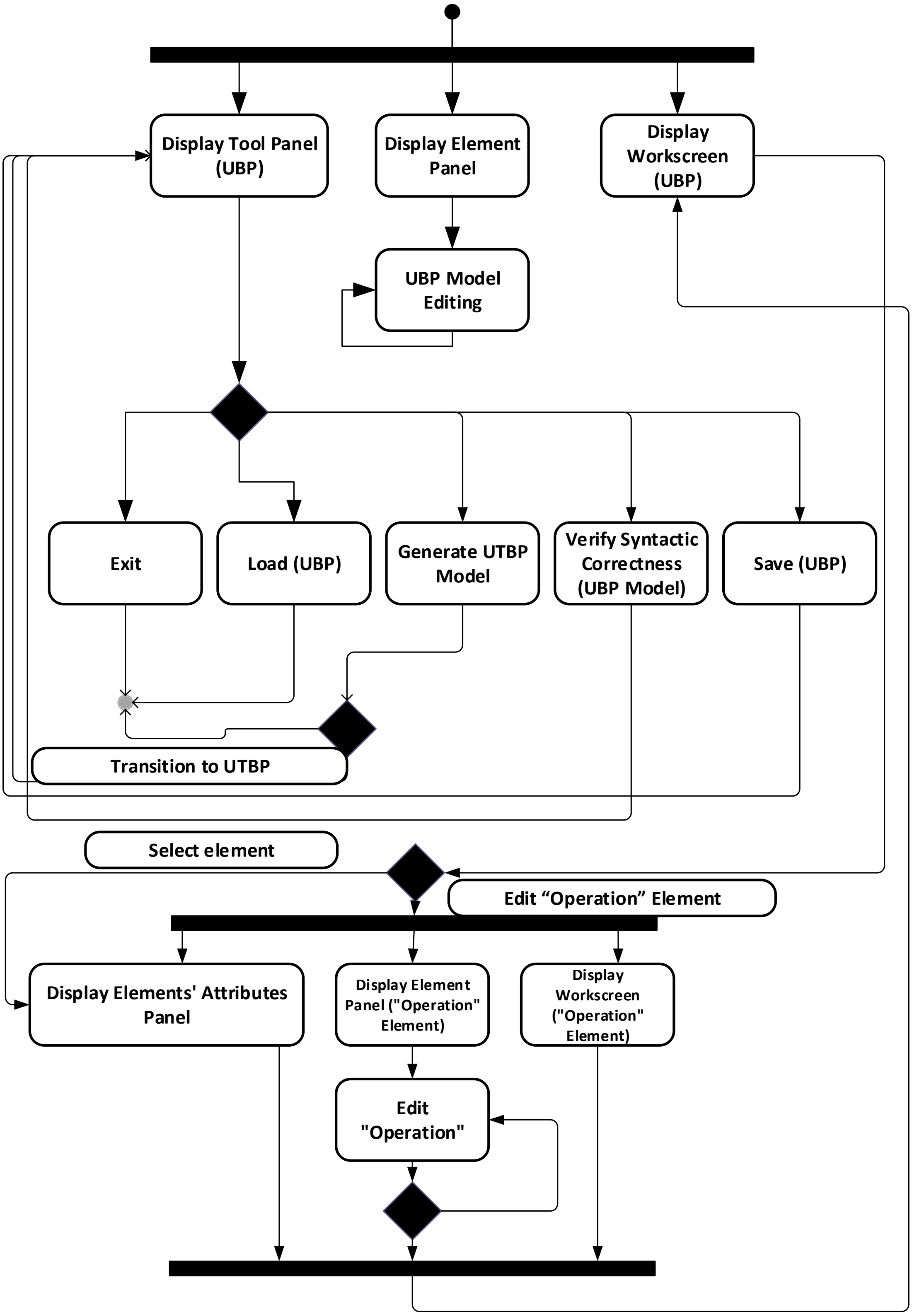


Figure 3. UBP Model Interaction Activity Diagram

Interaction with the UBP models begins with displaying a screen for model editing. Four major areas of this screen may be distinguished:

– *Main Work Area*. Serves the purpose of UBP model construction and display.

– *Toolbar Area*. Contains a set of main tools for model editing.

– *Element Panel Area*. Hosts a set of UBP metamodel elements, which are used for model construction.

– *Element Attributes Panel Area*. Displays and enables editing of selected element's attributes.

After the screen has been displayed, user may proceed constructing or modifying the model by adding notation's elements from the Element Panel.

Modification of the Operation element happens on a separate work screen. This is stipulated by the fact that Operation element possesses a considerable amount of attributes that require to be displayed on the screen. Thus, the stated operation of modification was decided to be carried out on a separate screen.

Modification of the remaining elements does not require additional screens, rather the Attribute Panel is leveraged. The latter is displayed only when an element is in user's focus, once the focus is lost the panel is hidden away.

The user may verify the syntactic correctness of built model at any moment of time. If the verification is successful, the user is given an option to generate the UTBP model from the current UBP model. Algorithm of UTBP model generation may be found in [Vikentjeva, 2014].

The user is also able to save his current model or load a previously built one.

The process of UTBP model construction is very much identical to that one of the UBP model. A special screen for UTBP models is displayed, which contains the same areas: Work Area, Toolbar, Element Panel, and Attribute Panel.

The user modifies the UTBP model in the Work Area using the same Toolbar and respective Panels.

When an element is selected, his attributes are displayed in the Attribute Panel. Operation element modification happens similarly to UBP model. Modification of DMP element is also carried out on a separate screen for the purpose of maintaining readability and convenience of the model.

The user may verify the syntactic correctness of built model at any moment of time. If the verification is successful, the user is given an option to generate the ALS expression from the current UTBP model.

The user is also able to save his current model or load a previously built one.

Interaction with ALS also occurs on an individual screen, which consists of Work Area and Toolbar Panel. The user can modify ALS, then save, or load another expression. The syntactic correctness of an ALS can also be verified.

Interactive Visual Model Editor Design

The analysis of requirements has shown that specific functions of model constructions (UTBP generation, DMP modification, ALS generation, resource modification of an operation, etc.) should be implemented as well as some of the generic functions of visual editors: element rendering and focusing, element alignment, layering, deletion, dragging etc. Due to this fact, it was decided to prepare a prototype using a premade editor with open source code and complete it with necessary changes and additions. Below are listed the prototype requirements:

- Open source code (the code must be enhanced to meet the CBGS requirements);
- Editor should focus in diagram creation (the editor is to be used for business process model creation);
- Editor should be written in C# (this is the main programming language of the development);
- Generated models (diagrams) should be XML-exportable.

“WPF Diagram Designer” has been selected as an editor prototype. This editor provides the necessary generic functionality and uses WPF-technology. Designer’s license permits source code modification and use in own applications. Moreover, according to the license, if the proportion of own code outweighs the proportion of source code, the application is allowed to be used for commercial purposes.

It is generally viable to use architectural patterns (design patterns) during the process of application development. Object-oriented patterns commonly provide a concept of classes and their interactions with each other [Gamma, 2001].

MVVM pattern was considered to be feasible for editor development, since the specificity of the latter implies a direct linkage between data model and data representation.

MVVM (Model – View – View model) consists of three elements:

- Model – reflect application’s business logic;
- View – visualizes data model;
- View model – is a view abstraction, alters the view upon suffering any changes.

Windows Presentation Foundation (WPF) is believed to be the most popular technology that supports MVVM [Gossman, 2005]. It is also used as a basis for the considered editor.

WPF can be successfully implemented when constructing both autonomous desktop applications and web-based browser applications.

The basic editor is quite limited in terms of required functionality, supporting only generic operations. Thus alterations to basic classes as well as completely new classes will be required for application development.

Below are given the limitations of basic “WPF Diagram Designer”:

- Naturally, the required layer of business logic is completely missing (elements and attributes of developed notation are not implemented);
- Database interface is not present (mechanisms of database uploading and downloading for both data model and visual representations) must be implemented;
- Element’s attributes cannot be edited;
- Metamodel transitions are not implemented (the user should be able to edit, generate and switch between relevant models).

Thus the graphical user interface of the editor should undergo following changes (Fig. 4):

- Add database interaction dialogues;
- Develop sets of elements for each of the models;
- Add buttons for UTBP and ALS generation;
- Implement Attribute Panel.

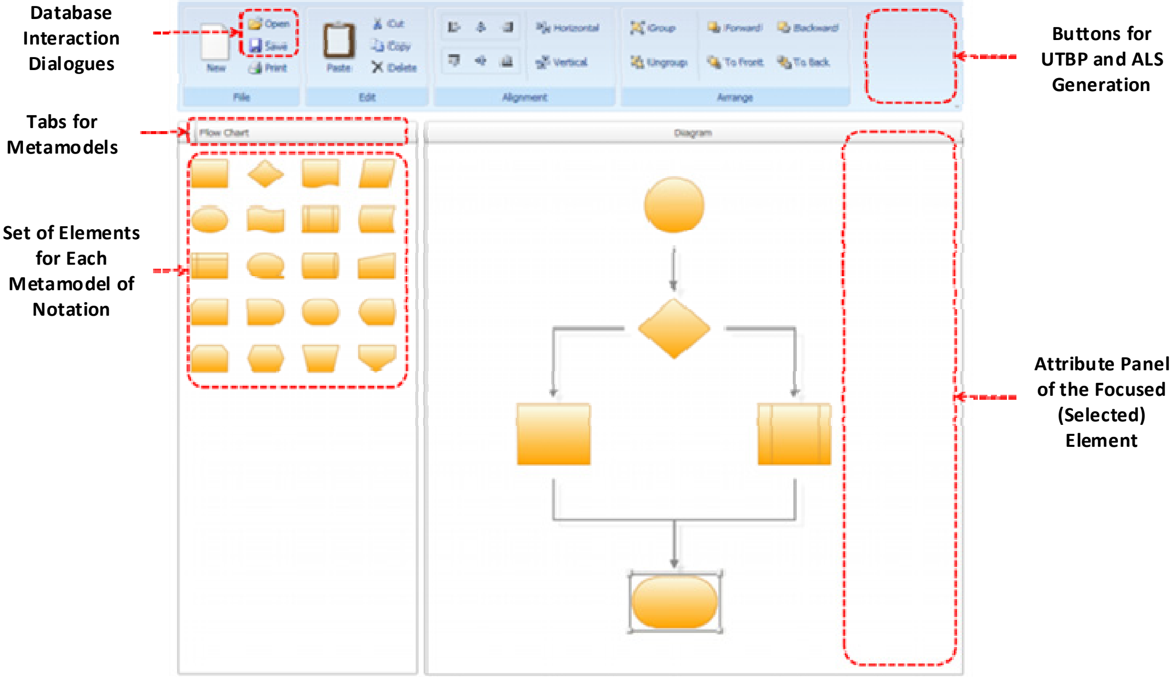

Database Interaction Dialogues
Tabs for Metamodels
Set of Elements for Each Metamodel of Notation
Buttons for UTBP and ALS Generation
Attribute Panel of the Focused (Selected) Element
New
Open
Save
Print
File
Paste
Cut
Copy
Delete
Edit
Horizontal
Vertical
Alignment
Group
Ungroup
Forward
Backward
To Front
To Back
Arrange
Flow Chart
Diagram

Figure 4. Changes to the Basic GUI

This screen stencil has been used during the development of own visual model editor. Taking stated requirements into account, the GUI of the editor has received the following form (Fig. 5).

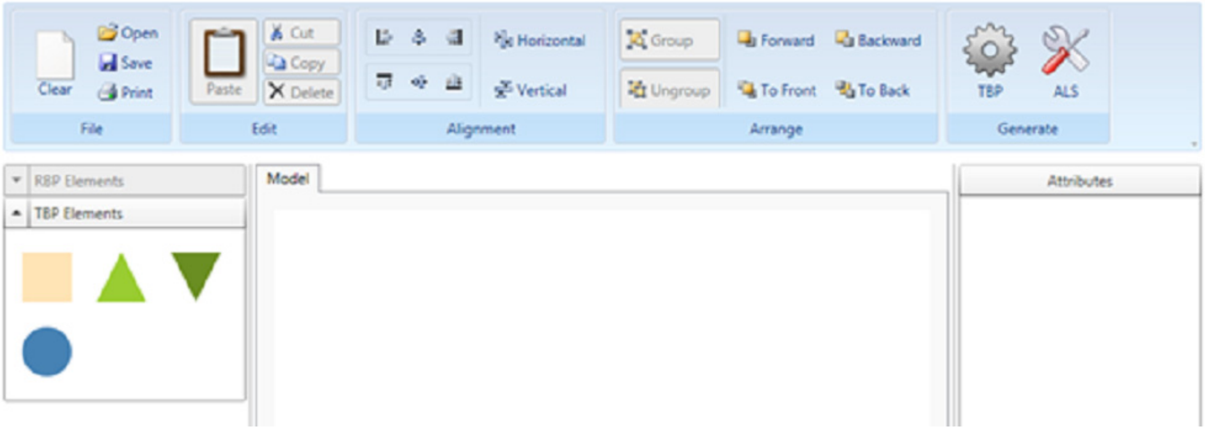


Figure 5. Visual Model Editor GUI

Fig. 6 shows Class Diagram of the editor. The diagram contains unaltered basic classes, modified and completely new classes.

Each class of the model corresponds to an element of the notation. All data model classes are inherited from the parent class BaseLogic. This has been done in order to implement common methods of attribute displaying.

Class Window1 contains XAML markup of the editor, accompanied by the resource and style dictionaries. This class is required for data model visualization and, basically, implements GUI of the application. It also hosts linkages to view model and data model. XAML markup implements the previously described screen stencil. Window1 is one of the basic classes, which belongs to the original assembly of the editor. The XAML markup was changed during the development, own style and resource dictionaries were implemented as well.

Class App is inherited from the standard class Application, it implements enumeration of all used in Window1 XAML markup dictionary resources.

Class ProgressBarPopup is called to visualize processing of slow procedures. The main purpose of this class is displaying a progress bar during procedure execution.

Table 1 contains all the classes of view model with description and relevant commentaries.

Considered classes have been used in the latest version of interactive visual model editor to implement attribute panel, element panel, graphical elements, UTBP and automaton model generation.

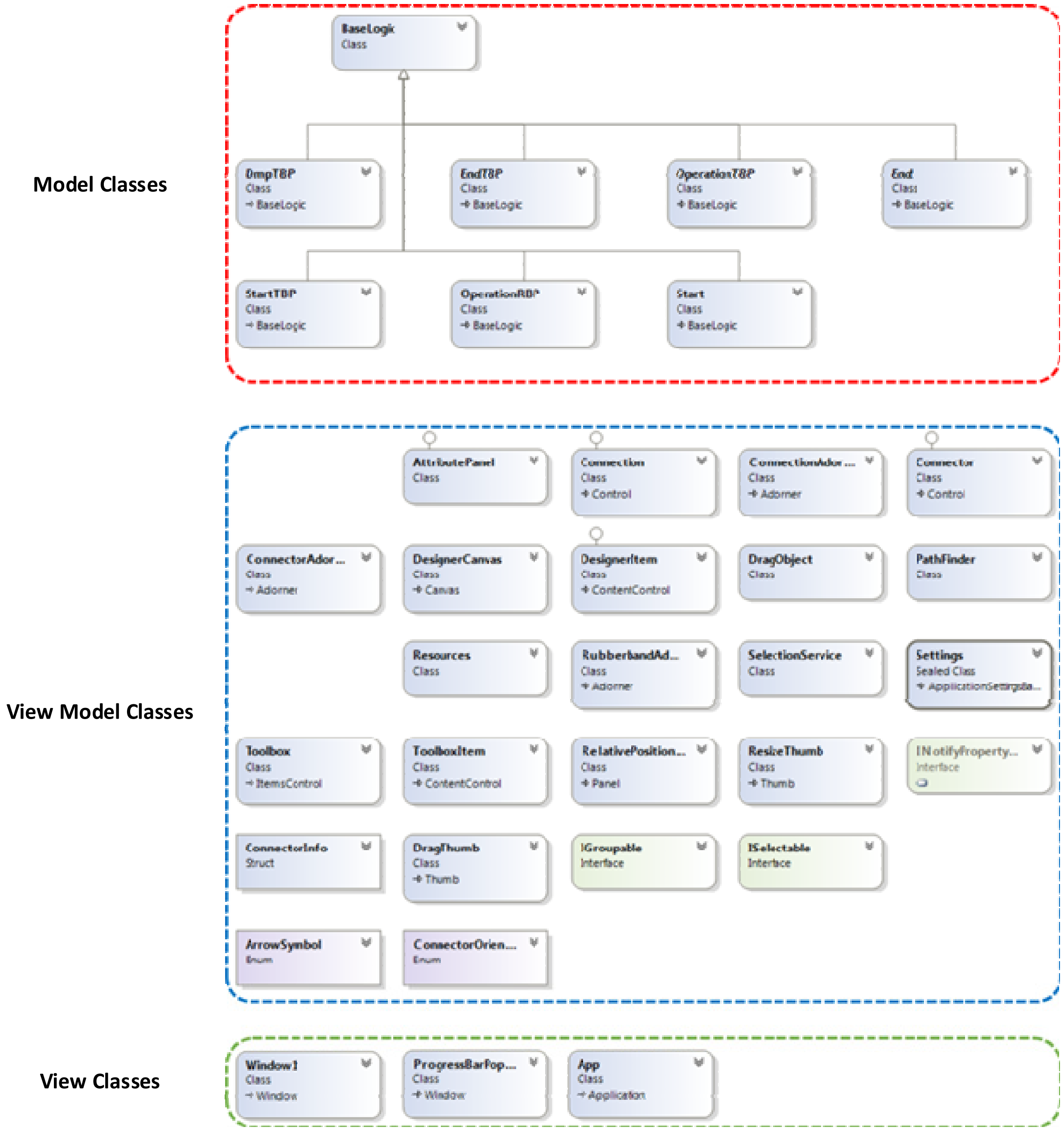


Figure 6. Class Diagram

Table 1. List of New and Modified Classes

Class Name	Description	Changes
AttributePanel	Displays and implements the logic of Attribute Panel.	This class does not belong to the basic assembly of the designer and has been composed from scratch. The class displays attributes of a logic element, which is bound to the selected graphical element on work screen. It also implements messaging mechanism and, therefore, conveys changes of interface to the data model.
ToolboxItem	Implements interaction with Element Panel items.	This basic class had to be modified so that Element Panel items contain logic object, besides graphical object.
DesignerItem	One of the main classes of the application. Displays and implements the logic of editor's graphical elements.	Changes to this class include following: – Linkage between graphical objects and logic objects. – Display of logic object's name on top of graphical object. – Notification mechanism that warns view and data model upon. Modification of class intance's attributes.
DesingerCanvas	Implements logic of interaction with editor's graphical elements.	Modifications to this class mainly relate to invocation of class methods, responsible for element focusing.
SelectionService	Implements the mechanism of element focusing.	Changes of this class include developments in invocation methods of Attribute Panel class.
DesignerCanvas.Commands	Implements interface commands of class DesignerCanvas, which are called via Toolbox Panel.	Modifications of this class are related to the Save and Load procedures as well as UTBP and ALS generation. All commands had to be set up to include cooperation with logic objects, commands of UTBP and ALS generation also had to be developed from scratch.

Conclusion

The paper suggests an approach, which enables automatic acquisition of UTBP model and automaton model of computer business game based upon multimodel representation of business processes.

The need for BG automaton model construction automatization is stipulated by following factors:

- mulimodel representation of business processes;
- complexity of process models impede manual design of business game;
- well-formalized models, which are used for business game design.

"WPF Diagram Designer" has been chosen to be used as a prototype of model editor. The Designer provides generic functionality of working with model elements and utilizes WPF technology. The license also permits source code to be leveraged in own developments. The GUI of the basic editor has received following modifications:

- database interaction dialogues have been added;
- for each metamodel a set of elements has been developed;
- buttons for UTBP and ALS generation have been added, corresponding algorithms have also been implemented;
- Attribute Panel has been added.

Thus, a contemporary CASE-tool has been developed, which enables construction and modification of UTBP model in real-time during the process of BG design (interactive visual model editor).

Bibliography

[Biggs, 1990] Biggs William D. Introduction to Computerized Business Management Simulations // Guide to Business Gaming and Experiential Learning – London : Nichols/GP Publishsing, 1990.

[Draganidis, 2006] F. Draganidis. Chamopoulou P and Mentzas G An Ontology Based Tool for Competency Management and Learning Paths 6th International Conference on Knowledge Management I-KNOW 06, Special track on integrating Working and Learning, 6th September 2006, Graz, (2006).

[Gossman, 2005] Gossman John. Introduction to Model/View/ViewModel pattern for building WPF apps // MSDN Blogs. [URL: http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx] [26.04.2015].

[Vikentyeva, 2014] Vikentyeva O.L., Deryabin A.I., Shestakova L.V. Algorithms of Automate Model Constriction for Business Game Execution Subsystem // International Journal "Information Models and Analysis". 2014. Vol. 3. No. 3. P. 271-279.

[Bazhenov, 2014] Баженов Р.И. Об организации деловых игр в курсе «Управление проектами информационных систем, Научный аспект, 2014. – т.1, №1, С. 101-102.

[Vikentyeva, 2013] Викентьева, О.Л. Концепция студии компетентностных деловых игр [Электронный ресур] / О.Л. Викентьева, А.И. Дерябин, Л.В. Шестакова // Современные проблемы науки и образования. – 2013. – № 2; [URL: http://www.science-education.ru/108-8746] (дата обращения: 03.04.2013).

[Vikentyeva, 2015] Викентьева О.Л., Дерябин А.И., Кожевников Д.Д., Красилич Н.В., Шестакова Л.В. Подсистема проектирования информационной системы для проведения деловых игр // В кн.: Технологии разработки информационных систем: сборник статей международной научно-практической конференции. Таганрог : Издательство ЮФУ, 2015. С. 27-32.

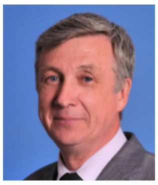
[Vikentyeva, 2015] Викентьева О.Л., Дерябин А.И., Шестакова Л.В., Лебедев В.В. Многомодельный подход к формализации предметной области // Информатизация и связь. 2015. № 3. С. 51-56.

[Gamma, 2001] Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Дждонсонс, Дж. Влиссидес. СПб: Питер, 2001.

[Girev, 2010] Гирев, П.Е. Инновационные подходы к использованию интерактивных моделей в обучении / П.Е. Гирев, О.И. Мухин, О.А. Полякова // Дистанционное и виртуальное обучение, 2010. – С.84.

[Polikarpova, 2008] Поликарпова Н. И., Шалыто А. А. Автоматное программирование. СПб, НИУ ИТМО, 2008.

Authors' Information



Alexander Deryabin – National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: paid2@yandex.ru.

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

Olga Vikentyeva – National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: oleovic@rambler.ru.

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

Lidiia Shestakova – National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: L.V.Shestakova@gmail.com.

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

Dmitrij Kozhevnikov – National Research University Higher School of Economics, City of Perm, Perm, Russia, e-mail: mefaze@yandex.ru.

Major Fields of Scientific Research: General theoretical information research, Multi-dimensional information systems

METHODS AND ALGORITHMS OF LOAD BALANCING

Igor Ivanisenko

***Abstract**: In this paper the classification of the most used load balancing methods in distributed systems (including cloud technology, cluster systems, grid systems) is described. Load balancing is represented on four levels of network model OSI: channel, network, transport, application. Features, advantages and shortcomings are presented for each level. Also strengths and weaknesses of network, transport and application levels are described. Basics of hardware based load balancing in Network Packet Broker and Application Delivery Controllers, that working on OSI layers 2-7, are described. In this work strengths and weaknesses of hardware based load balancing are shown. Basics of software based load balancing are carried out. Differences between software based load balancing and hardware based one are described too. In the work characterizations of the most used dynamic load balancing algorithms in distributed systems is described. Advantages and shortcomings of each algorithm are carried out. Load balancing uses a variety of methods and algorithms for balancing. In the work methods that are used on channel, network, transport, application levels of OSI model and available in balancers and/or can be configured on the servers are presented and analyzed. Employment, effectiveness, strengths and weaknesses of each type of the methods are described in accordance with analysis. Following methods are carried out: direct Routing, Network Address Translation, Source Network Address Translation, Transparent SNAT, SSL Termination or Acceleration, TCP/IP server load balancing, Hashing, Caching, DNS load balancing, Network Load Balancing, Proxy method, Load balancing by using redirection.*

***Keywords**: Keywords— load balancing, distributed system, hardware and software load balancing cloud, DNS, network level, Network Address Translation, proxy.*

***ACM Classification Keywords:** C.2.0 General – Open Systems Interconnection reference model (OSI), C.2.3 Network Operations - Network management, C.2.4 Distributed Systems - Client/server, Distributed applications*

Introduction

Evolutionary processes occurring in communication networks are inevitably reflected in amount and internal traffic structure. According to numerous studies the total amount of data transferred over the WAN, shows a steady exponential growth despite the fact that this trend will continue in the coming years [Kopparapu, 2008; Erl, 2013].

With increasing amount of data, the behavior of traffic in today's global network shows such a negative feature as the instability of the load, which is characterized by the possibility of the emergence of unpredictable surges of transmission intensity. There are many causes of such instability. A prime example can conduct multiple users on the network caused by the viral nature of the spread of the popular media. Avalanche-like process of attracting new users, new ways of collective communication, based on the widespread use of social services, mass online broadcast - all this makes talking about the new nature of emerging overloads.

The researchers note that today's networks suffer from a lack of bandwidth. According to research about 20-30% of WAN links are routed through congested areas. At the same time there is considerable nonuniformity of load distribution channel resources, indicating procedures are inefficient traffic management in the current environment [Erl, 2015].

The way out of this situation is the use of special methods of balancing traffic to effectively distribute the load in accordance with the existing untapped resources.

The issue of capacity planning should be addressed in the early stages of building a network or project. Initially, the problem of insufficient capacity nodes due to increased loads can be solved by increasing their capacities, or the optimization of the algorithms, software code, and so on [Erl, 2013; Кириченко, 2011]. But sooner or later the moment comes when these measures are insufficient. And then it is necessary to use load balancing methods.

Load balancing is implemented using hardware, software instruments, or a combination of both. Previously, it was clear delineation of hardware and software load balancing. Now, in connection with the development and improvement of both hardware and software load balancers, the boundaries between them are deleted [Roth, 2008; Natario, 2011]. Assume that if hub, switch, Application Delivery Controllers (ADC) are used it is hardware base balancing.

When using the server (computer), we assume that the software load balancing occurs. Hardware load balancing is often used at a channel, network and transport layers. In general, hardware load balancing, faster than software solutions, but its drawback is the cost.

Software based load balancing as opposed to hardware load balancer operates on a standard operating system and standard hardware components such as a PC. Software solutions operate in dedicated hardware load balancing node or directly in the appendix. The hardware load balancing devices are used called Network Packet Broker (NPB) (or Network Monitoring Switch), which have 100GbE interfaces and work on 2 and 3 levels of the network model OSI [Laviol, 2014].

NPB is embedded into a rack network device, receiving and aggregating network traffic from the ports, which it manipulates in the future. The primary and most important function of NPB is load balancing. For load balancing 3-7 levels used ADC [Gurevich 2010]. The procedure of load balancing is carried out using complex algorithms and methods that conform to following levels of the network model OSI: channel; network; transport; applied [Бажин, 2010].

The purpose of this paper is to review the existing basic algorithms and methods for hardware and software based load balancing at different levels of the network model OSI, analyze of their strengths and weaknesses.

In the first section of this paper client based load balancing and server based one are carried out. In the second section the load balancing is described on four levels of network model OSI, their features, advantages and shortcomings are analyzed. In the third section the basics of hardware load balancing and its differences from the software load balancing are described. In the fourth section of this work the survey of few existing load balancing algorithms are carried out. The basic methods of hardware and software load balancing, which are used at different levels of model OSI, are described in the fifth section, their advantages and shortcomings and range of application are described too.

1. Client and server based load balancing

Schematically load balancing in distributed systems can be represented as a structure (Figure 1).

Briefly describe the main components of the scheme. Client based load balancing usually is much worse than server based load balancing [Cardellini, 1999; Koteswaramma, 2012; Pavan Kumar, 2012]. The reason is that clients often can not monitor server availability or load rate. If the server is overloaded or offline, the client waits for a timeout before he tries to connect to another server. The dissemination of information to the client about servers load creates an additional network load and the delay of dissemination information is added to the total time of service. Availability of servers also is very dynamic, so the client can not use this information for a long time period. Also, the balancer can amortize the cost of querying server availability over many requests.

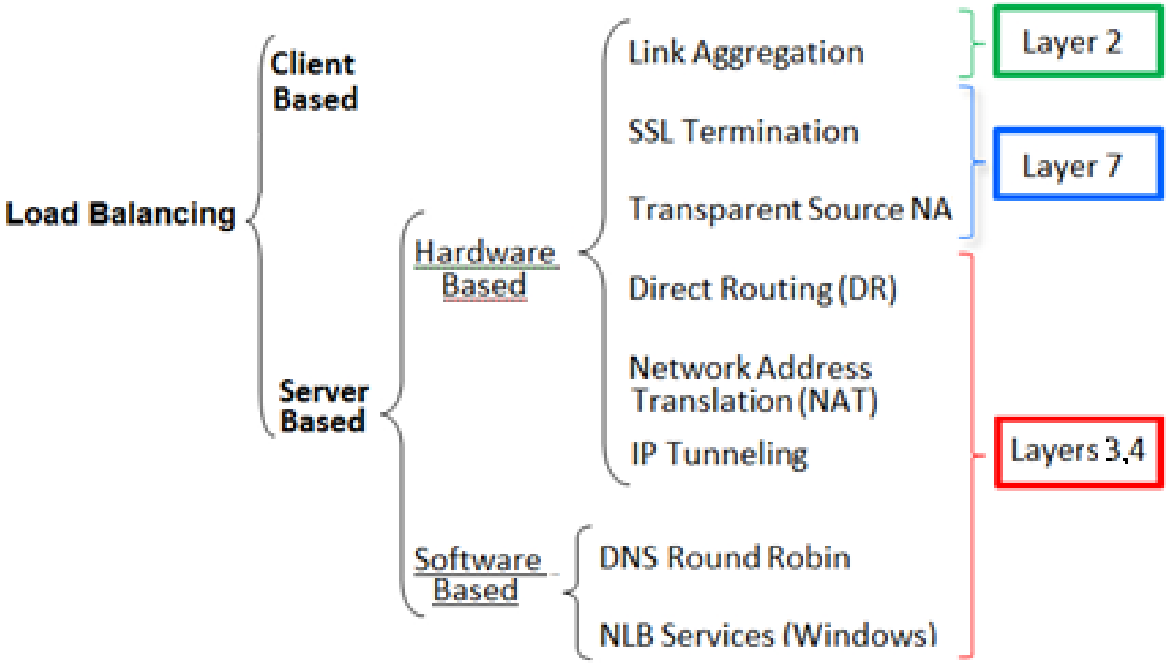


Figure 1. The methods of load balancing in OSI model

Server based load balancing, therefore, incurs no latency penalty to the client. Client based load balancing (Figure 2) typically uses more bandwidth than a server based load balancing. This is because the network path for each route client-server could potentially take different routes.

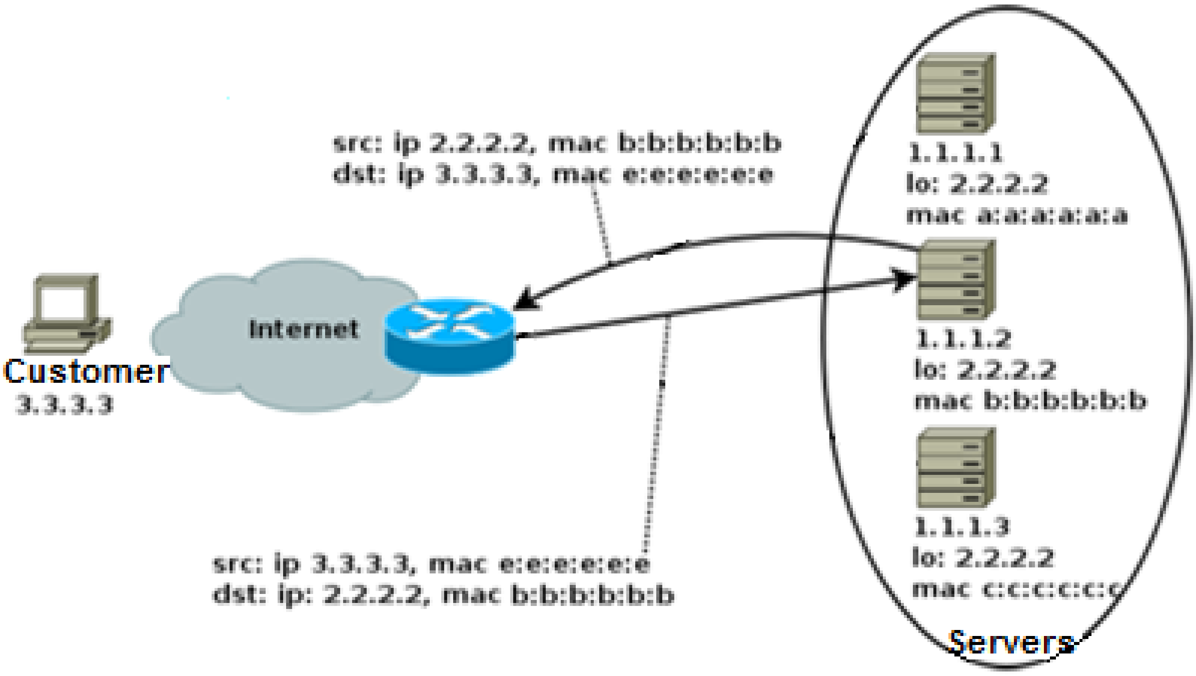

src: ip 2.2.2.2, mac b:b:b:b:b:b
dst: ip 3.3.3.3, mac e:e:e:e:e:e
Internet
Customer
3.3.3.3
1.1.1.1
lo: 2.2.2.2
mac a:a:a:a:a:a
1.1.1.2
lo: 2.2.2.2
mac b:b:b:b:b:b
1.1.1.3
lo: 2.2.2.2
mac c:c:c:c:c:c
Servers
src: ip 3.3.3.3, mac e:e:e:e:e:e
dst: ip: 2.2.2.2, mac b:b:b:b:b:b

Figure 2. Client based load balancing

In this example, the user directly connects to a server address 1.1.1.2 without any balancing. If the server does not respond, then the user can not get access to the necessary resources. Also, if a very large number of users access the server, many of them will be answered with a big time delay, and some have not it. Another variant of client based load balancing: a list of application servers can be inserted into client code. I.e. on the client side there is a list of available servers to which the client tries to communicate until he finds one that responds.

When server based load balancing is possible to do that multiple servers are reflected as a single server - single virtual service - transparent distribution of user requests between servers. The distribution of load to the server prevents the disabling of hardware and software services to end users, and can also provide disaster recovery services by redirecting service requests to a backup copy when the main resource is disable [Cardellini, 2001; Roth, 2008; Jiao, 2010; Tuncer, 2011; Zhihao, 2013]. There are two categories of implementation of server based load balancing:

• **software based load balancing** consists of a special software installed on the servers in the load balancing cluster. The software sends and receives requests from the client to the server based on various algorithms. For example, Microsoft Network Load Balancing is a software load balancing for Web farms, and Microsoft Component Load Balancing is a software load balancing for applications in the farm.

• **hardware based load balancing** consists of a special switch or router with software to provide load balancing functionality. This solution combines switching and load balancing into a single device, resulting in reduce the amount of additional equipment necessary for the realization of load balancing. Modern equipment load balancing devices are known as NPB.

2. Load balancing on levels of OSI model

2.1. Load balancing on the second (channel) level

Load balancing on the second level of the protocol stack there are two options: balancing using a separate dedicated balancer and without it (Figures 3, 4) [Бажин, 2010].

In both cases, some IP-address of the service is set for all servers or to other specialized interface. This is done to ensure that these servers can accept connections on this IP-address and respond from it, but do not respond to ARP-requests (Address Resolution Protocol - the protocol definition addresses) belonging to this address.

This balance is performed as follows: on the balancer which has IP-address and responds to ARP, first packet connection comes. Balancer determines that packet was first. This packet is sent to the server using needed algorithm, changing the MAC-address to destination, it is written in connection table. If this is not the first packet, it looks at which server processes this connection using connection table, and the packet is sent there. Considering that the headers of the third and higher levels are not modified, the response from the server can be sent past the balancer directly over the Internet to the client (to the necessary gateway).

The most common currently solution from software implementations of this method is called the Linux Virtual Server. In the URL-terminology this load balancing method is called direct routing.

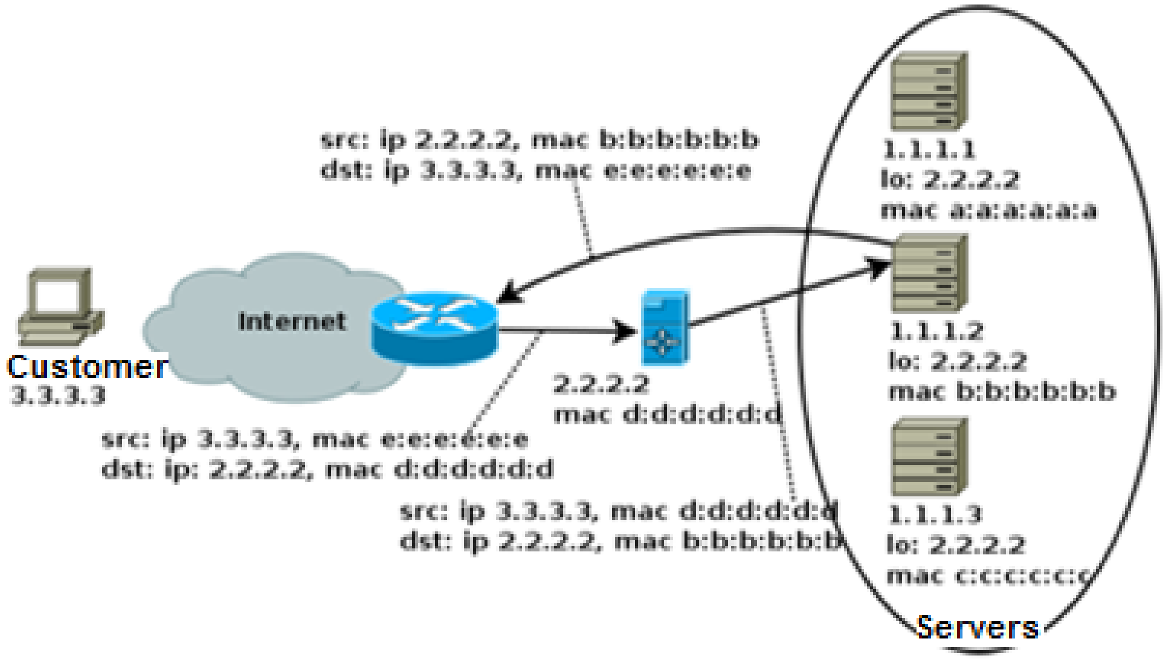

src: ip 2.2.2.2, mac b:b:b:b:b:b
dst: ip 3.3.3.3, mac e:e:e:e:e:e
1.1.1.1
lo: 2.2.2.2
mac a:a:a:a:a:a
Internet
Customer
3.3.3.3
1.1.1.2
lo: 2.2.2.2
mac b:b:b:b:b:b
2.2.2.2
mac d:d:d:d:d:d
src: ip 3.3.3.3, mac e:e:e:e:e:e
dst: ip: 2.2.2.2, mac d:d:d:d:d:d
src: ip 3.3.3.3, mac d:d:d:d:d:d
dst: ip 2.2.2.2, mac b:b:b:b:b:b
1.1.1.3
lo: 2.2.2.2
mac c:c:c:c:c:c
Servers

Figure 3. Load balancing on the channel level

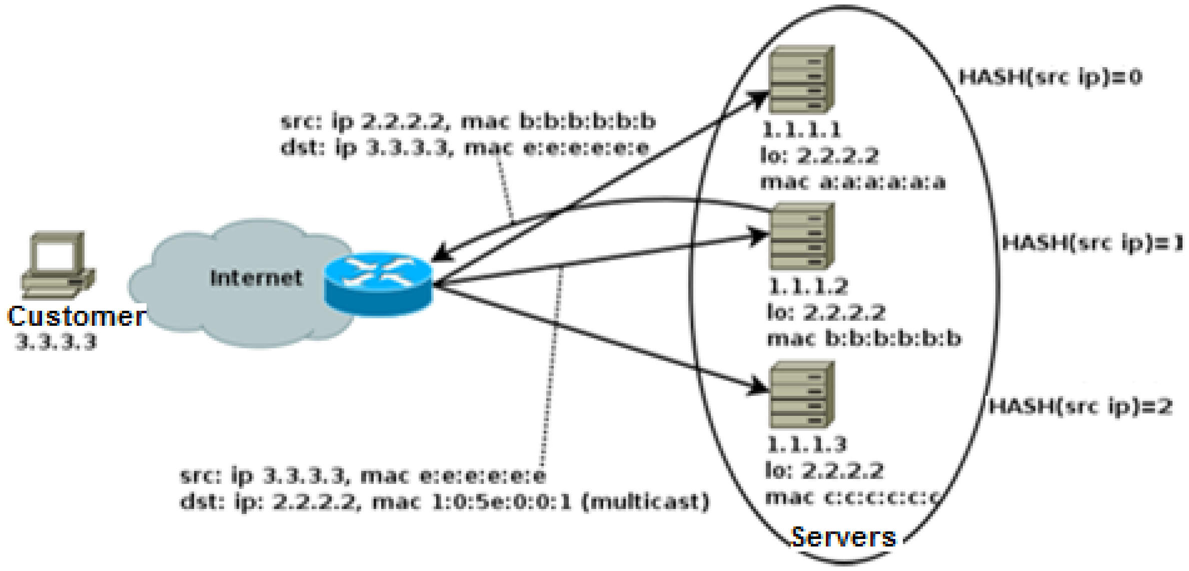


Figure 4. Load balancing on the channel level without a dedicated load balancer

Load balancing without a dedicated load balancer. In this case, the IP-address of the service is prescribed as a static ARP-record at the gateway to some of the multicast MAC-address. The switches are configured in a way that frames that coming in the MAC-address, delivered with all the necessary servers. The hash is calculated on some servers, for example the clients IP-address. According to the value server determines whether it should respond to these requests. If HASH = 0, the first server must answer. It responds. Other servers "know" that they do not have to answer.

Advantages of load balancing on the channel level.

- Independence from the high level protocol. Relatively low resource consumption. It is possible to balance the HTTP, FTP or SMTP - the difference will not be. There is a method of balancing without a dedicated load balancer. With low number of servers it can be actual. It is possible to send answers past balancer. Taking into account that, for example, in the HTTP protocol the size of response is typically larger than the size of the request, then the increase of resource economy is take place.

Shortcomings of load balancing on the channel level.

- All servers must be in the same network segment. Specific configuration of servers and network equipment are necessary. Therefore, this method is not always convenient and applied. As the implementation of this scheme the cluster IP on the firewall IP-tables for Linux can be used.

2.2. Load balancing on the third (network) level

Load balancing on the network layer (Figure 5) surmise the following tasks: to do so for one particular server IP-address corresponding to different physical machines [Hong, 2006; Roth, 2008; Бажин, 2010]. Balancer is assigned the same IP-address service. When a request comes to it, destination NAT is applied, i.e. a destination IP-address is substituted in the packet: IP-address of current server is changed to necessary IP-address of the server that will handle the request by selected algorithm. The difference between this method and the previous one is that headers of the third level are modified. Sender IP-address should be changed from the server IP-address that handles the request to the service IP-address, which is used in the balancer.

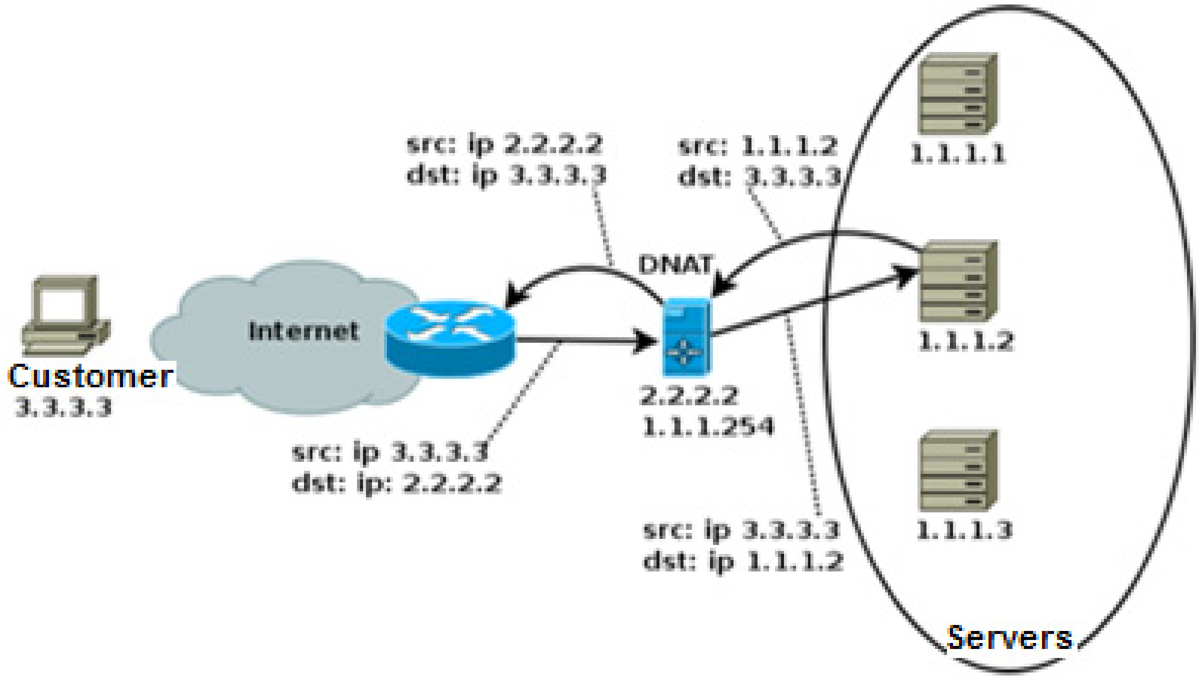


Figure 5. Load balancing on the network level

There are many realizations of this method.

- DNS-balancing. For the one domain name is allocated several IP-addresses. The server, on which a client request will be sent, is typically determined by a load balancing algorithm, e.g. Round Robin.

- Construction of NLB-cluster. By using this method, the servers combined into clusters, which consist of input and computing nodes. Load balancing is performed by using a special algorithm. It is used in Microsoft solutions.

- Load balancing IP network using additional router.

- Load balancing on a territorial basis is performed by placing the same services with the same address in geographically different regions of the Internet.

Advantages of load balancing on the network level.

• Independence from the high level protocol. Complete transparency for servers.

Shortcomings of load balancing on the network level:

• Reverse server traffic should pass through the balancer. Accordingly, the load on it will be higher than when using the balancer on the second level.

2.3. Load balancing on the fourth (transport) level

This type of balancing is the easiest: the client address to the balancer, that forwards the request to one of the servers, which will process it (Figure 6) [Hong, 2006; Roth, 2008; Бажин, 2010]. The choice of the server that will process the request can be carried out in accordance with the different algorithms: sorting by simple circular, by selecting the least loaded server from a pool, etc.

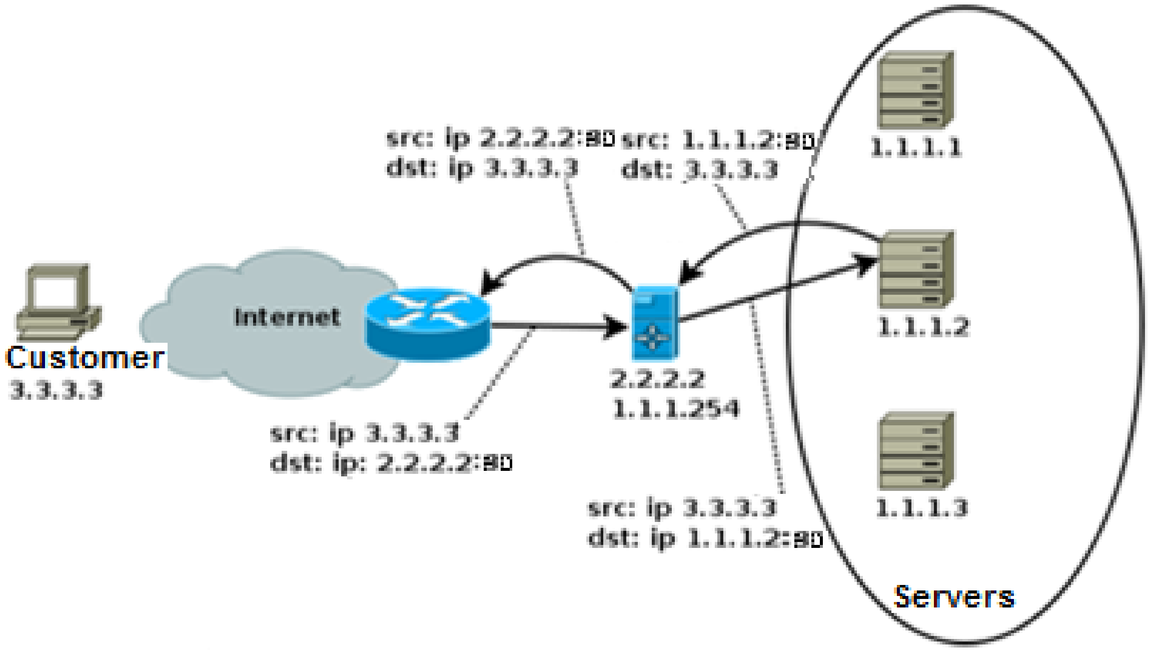


Figure 6. Load balancing on the transport level

Sometimes the balance on the transport layer is difficult to distinguish from the balance on the network level. When balancing of outgoing traffic at the network level takes place, specific port or specific communication protocol are indicated. The difference between the levels of balance can be explained as follows. Solutions of the network layer are not terminates the user session on themselves. They simply direct traffic and do not work in proxy mode. On the network level load balancer decides on which server to transmit packets. A server provides a client session. On the transport layer a communication with the client becomes isolated on balancer, which acts as a proxy. It communicates with the server on its own behalf by passing client information in additional data and headers. The popular software balancer HAProxy works in this way.

2.4. Load balancing on the seventh (application) level

When balancing on the application level load balancer works as "smart proxy" [Vlaeminck, 2004; Hong, 2006; Roth, 2008; Mendonca, 2014]. It analyzes client requests and forwards it to different servers depending on content of requests. Web server Nginx works by distributing requests between frontend and backend. In contrast to the low-level load balancing solutions, load balancing in the seventh level works with knowledge of application. One of the popular architectures load balancing is shown in Figure 7, it includes load-balancing on the application and transport layers.

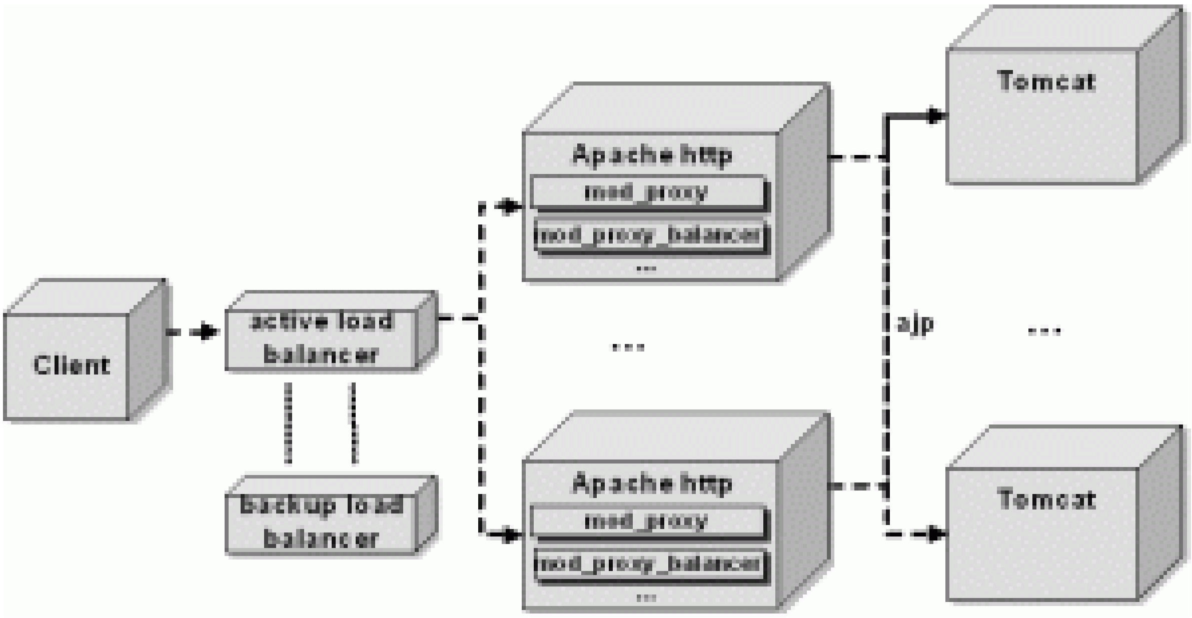


Figure 7. Load balancing on application and transport levels

Load balancing on application level serve as a normal server for load balancing on transport layer. Incoming TCP connection are directed to balancer at the application level. When it receives a request on the application level, it determines the destination server based on application layer data and forwards the request to that server.

In this example (figure 8) user visits a high-loaded site. During the session, the user can request a static content (images or videos), dynamic content (news channel, transactional information - the order status). Load balancing on 7-th level allows to route requests using information of the requested content. So now the request of image or video can be routed to the servers, which store it, and it is possible to optimize the service for multimedia content.

Request for transaction information, such as payment, can be directed to the application server, responsible for managing pricing.

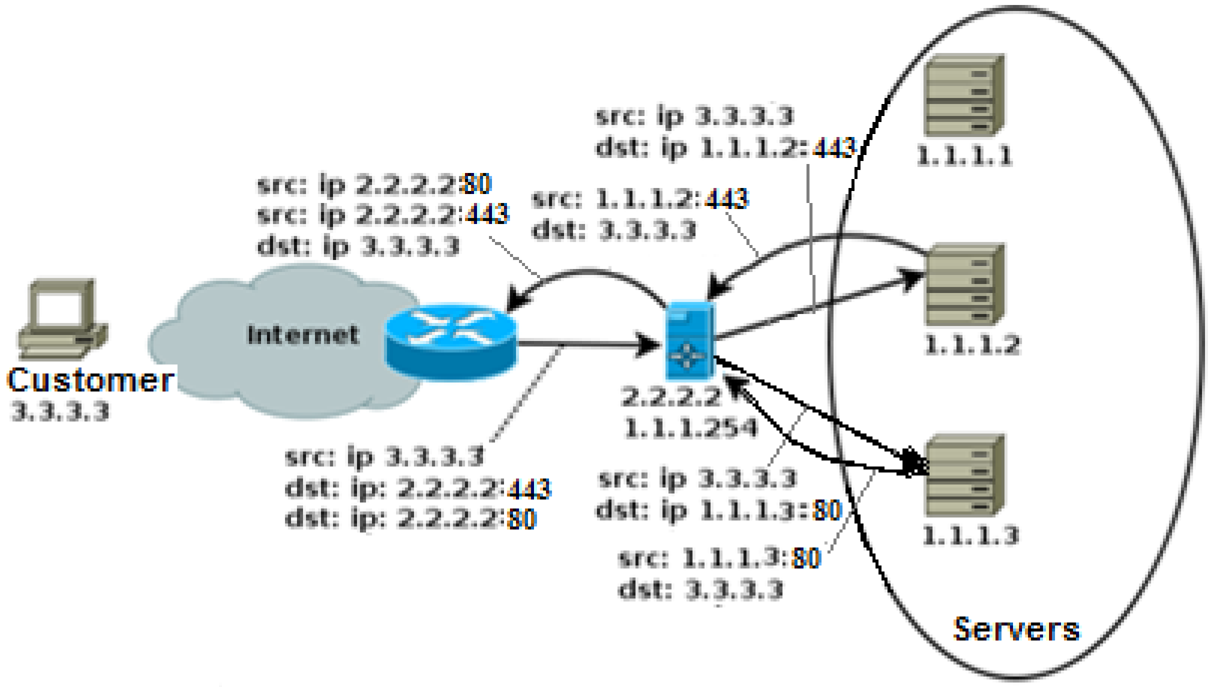


Figure 8. Load balancing on application level

Load Balancing Layer 7 also allows to increase the efficiency of the application infrastructure, because different types of content have different requirements in terms of CPU usage, bandwidth, etc. Thus it is possible to obtain higher efficiency servers classify them in such a way that some of them are treated with a transaction, while others simply act as a massive storage systems for serving static pages or optimized for downloading streaming video, for example.

Application delivery controllers perform load balancing on 7-th layer called (ADC), and they combine the features of traditional load balancing with advanced application switching of 7 level to provide design a scalable, optimized application delivery network.

Advantages of load balancing on the application level:

• rarely reduce performance in the modern server; make more informed decisions of load balancing; it is possible to apply the optimization and content changes (such as compression and encryption); buffering is used to unload the slow connections from superior server, that improves performance.

Shortcomings of load balancing on the application level:

- overhead on the analysis of requests is high; limited scalability as compared with load balancing on the other levels.

2.5. Differences between 4-th and 7-th levels load balancing

Load balancing level 4 deals with the message delivery, without regard to the content of the message. Transmission Control Protocol (TCP) is a protocol of level 4 for hypertext transfer protocol (HTTP) traffic on the Internet. The transport layer load balancing only transmits network packets to and from the superior server without checking packages contents. At this level, it can be making limited decisions to route by checking the first few packets in a stream of TCP.

Load balancing layer 7 works at the application level, which has to deal with the actual content of each message. HTTP is the dominant protocol level 7. Load balancing on the application-level directs network traffic using much more complex mechanisms than the choice of the path on the transport level load balancing.

Load Balancing layer 7 pauses network traffic for reading messages. Thus, the load balancing decision made using message content (URL or cookie, for example). Then new TCP connection is created to selected upstream server (or to recurring, via HTTP-support activity) and request is sent to server.

3. HARDWARE BASED LOAD BALANCING

Load balancing hardware devices working on OSI layers 2-7 and used for splitting the network load among multiple servers in terms of factors such as utilization of processor CPU, number of connections, total server performance [Natario, 2011; Laviol, 2014].

Use of this type of technology minimizes the probability that any particular server will be overloaded, and optimizes the throughput for each computer or terminal. In addition, by using balancer it can be minimized network downtime, to ease traffic prioritization, to monitor applications from end to end, to provide user authentication, and help defend against harmful activity such as denial of service (DoS) attacks.

Router LVS uses low-level filtering, that has advantages compared to redirect requests on the application level, because of the load balancing on the transport level does not cause significant computational costs and can be scaled [Red Hat, 2015].

The basic principle is that the network traffic is routed to a common IP called virtual IP (VIP), or listening IP, and this address is assigned to the balancer. After the load balancer receives a request on this VIP, and it will need to decide where to send request, and that decision is usually controlled by the load balancing algorithm and set of rules. Then the request is sent to corresponding server and server will generate a response which depending on the type of load balancing. Response will be sent back to either the balancer in the case of the device layer 7 or, as a rule, from the device level 4 directly back to the end user (typically using default gateway). In the case of proxy based load balancer, a request from a Web server can be returned to the balancer and processed before being sent back to the user. This processing can include replacement of content or compression or other scenarios. A more detailed look at the processes taking place inside the NPB at the level of software solutions and basic methods and algorithms that used for load balancing.

Load balancing in NPB is division process input stream from one or more interfaces to multiple output interfaces by certain rules or criteria. The following functions are used with a balancing:

- filtering - rules to identify flows for their subsequent balancing and reducing the amount of data in these flows;
- aggregation - merge of flows from multiple input interfaces into a united before the operation of balancing;
- connections - most NPB can work as a switch.

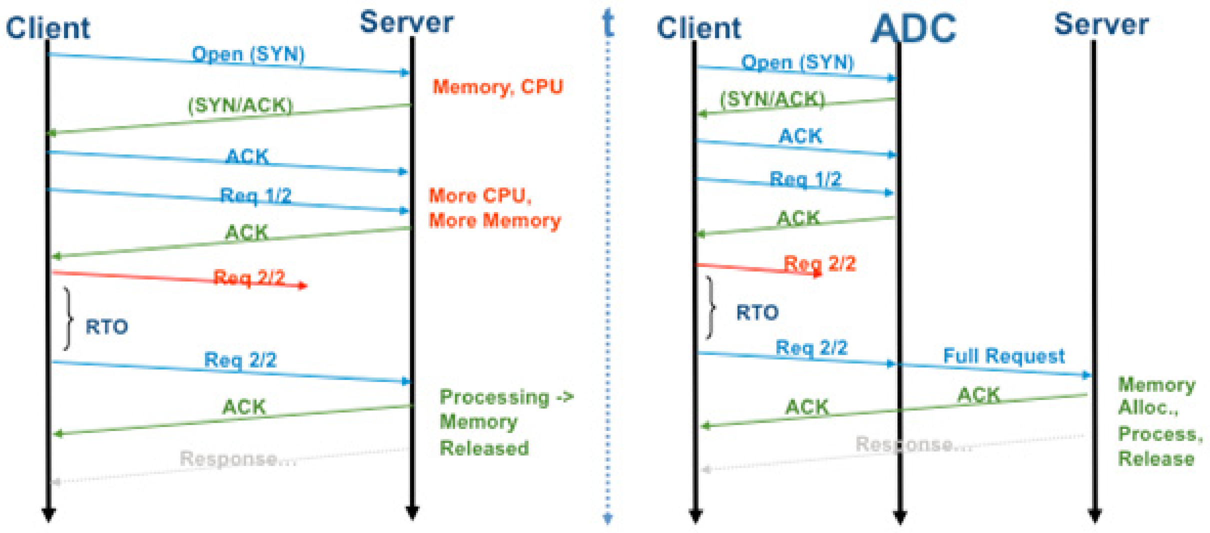
The main application of NPB is a selection of large necessary data flows and their division into smaller ones. This problem occurs quite often. Traffic coming from the several ports is aggregated and comes to the NPB. Accordingly the first condition, one of the main require is made if client equipment works with a flow on the session level, then NPB does not have these sessions to break, then packages of one session should always come to the same output interface. This property is called Flow Coherence. As discussed in [Laviol, 2014] in the NPB traffic are filtered, which allows to filter incoming packets using the defined rules by various network protocols, cut and analyze part of the package, insert ports labels, VLAN, MPLS into packets, mark and delete duplicate packets.

Usually flows coming in NPB are determined by header (src/dst IP, src/dst port, protocol). But the structure of flows may change, and NPB must be able to adapt to them. For example, if the IP-address is fixed, and the ports vary depending from the path of a package or vice versa, the ports are fixed and IP-addresses can vary.

The NPB has functions that allow to monitor the status of channels, to reserve and manage channels. NPB also uses different algorithms for traffic management, which will be discussed later in this paper.

3.1 Differences software and hardware load balancing

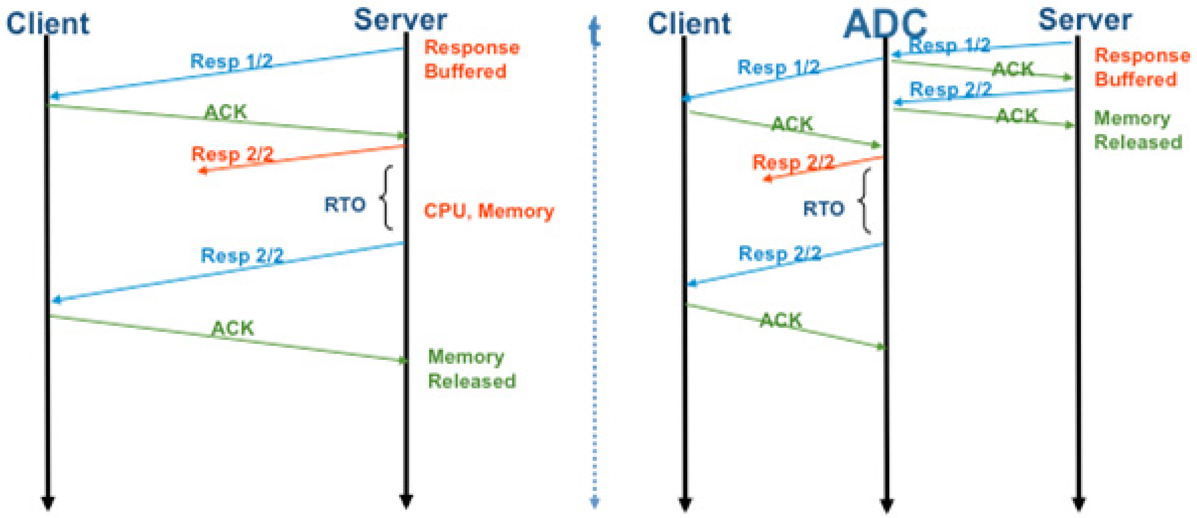
The preliminary program approval communication conditions (handshake) is made separately, multiplexes - separately. The server is not required to deal with such problems. It does not deal with inquiries, transactions, and sends HTTP-page. As a result, processor, bus and memory are unloaded (Figure 9) [Гуревич, 2010].



a) b)

Figure 9. Handshakes for software and hardware load balancing

Consider example of buffering requests (Figure 10).



a) b)

Figure 10. Example of buffering requests for software and hardware load balancing

Buffering looks as follows on the server: open (SYN), (SYN/ACK), etc. Processor and memory are unloaded. If the application delivery controller (ADC) is placed between the server and the client, all realized using ADC. The server does not know about it. Only at the end, it begins to engage to return pages.

An answer is buffering by other way. A server communicates directly with the client. It works with Crescendo or other application delivery controllers, it behaves as if the computer is in the local network. A server does not work in the global computer network and from it side, retransmission do not occur, and the process proceeds normally. Response time is much less. After using the application delivery controller a load on the processor decrease significantly. The processor is completely unloaded. Response time decreases only due to the fact that the server return deals exclusively with web-pages.

4. SURVEY OF FEW EXISTING LOAD BALANCING ALGORITHMS

Load balancer uses different algorithms for traffic management for the purpose of load balancing and/or the maximum use of all the servers in distributes systems. Higher bandwidth and improves response time in a distributed system are achieved because of algorithms. Each algorithm has advantages and shortcomings.

1. **Task Scheduling based on LB** [Singhal, 2011; Ghanbari, 2012; Ghuge, 2014; Rajwinder, 2014] is dynamic algorithm, based on load balancing, consists of a two-level scheduling mechanism. It provides a high efficiency of resources using. This algorithm balances the load by the primary tasks distribution in virtual machine, and then all virtual machines distribute on the host resources, in this way to improving the tasks response time, resource utilization and efficiency of cloud computing environment. This algorithm ensures the satisfaction of the dynamic users requirement and high ratio of resource utilization.

2. **Opportunistic Load Balancing (OLB)** [Singhal, 2011; Ghuge, 2014; Rajwinder, 2014] is static algorithm, trying to occupy each node, so it do not take into account the current load of each node or its suitability for the task. In other words, OLB sends unfulfilled tasks to currently available nodes in a random order, regardless of the current nodes load. The advantage is the simplicity, achieving load balancing, but its disadvantage is that it does not address the expected execution time for each task, resulting in a higher average completion time (total cycle time).

3. **Round Robin** [Keshav, 1997; Ghuge, 2014; Rajwinder, 2014] is a listing of the circular cycle – the first task is transferred to a node, then the next task is transferred to another and so on until it reaches

the last node, and then it all starts again. In this algorithm, all tasks are divided equally among all the processors, but various problems have a different run time, that is absolutely not taken into account load of nodes in the cluster. In Round Robin Scheduling (task management in systems with a time distribution), the algorithm identifies ring as a queue, and the fixed time slot. Each task can be done only in the time slot and queue. If the task can not be completed within one time slot, it will return to the queue for waiting a next round. However, it is difficult to determine the appropriate time slot. When the time slot is very high, the RR scheduling algorithm works the same as FCFS Scheduling. When a time slot is too small, the Round Robin Scheduling is known as Processor Sharing algorithm. Balancing Method Round Robin DNS does not require communication between servers, so it can be used for local and global balancing, and solutions, based on the Round Robin algorithm, are low cost.

4. **Weighted Round Robin** [Keshav, 1997; Gupta, 2013; Rajwinder, 2014] is improved version of Round Robin algorithm: each node is assigned a weight in accordance with its performance and capacity. This helps distribute the load more flexibly: nodes with more weight process more requests.

5. **Randomized** [Ray, 2012; Rajwinder, 2014] is static algorithm that randomly distributes the load across the available nodes, selecting one of them using a random number generator and sending a current task to it. This algorithm works well when all processes have the same load, but when the load is different computational complexity there are problems. This algorithm does not support the deterministic approach.

6. **Min-Min Algorithm** [Elzeki, 2012; Chen, 2013; Ghuge, 2014; Kashyap, 2014; Rajwinder, 2014] is a static load balancing algorithm, so that the parameters relating to the work are previously known. The algorithm as soon as possible provides resources for tasks that can be done as soon as possible. Minimum execution time of each task is found. The minimum value of time is searching among minimum execution time and the task with that value is sent for execution. Until all tasks have been assigned for execution a queued task will be updated and completed and executed tasks will be removed from the queue. Tasks with the maximum wait time should a non-specific period of time. The main problem with this algorithm is that it can lead to acute shortage. It works best when most of tasks have minimal execution time.

7. **Max-Min Algorithm** [Elzeki, 2012; Katyal, 2013; Ghuge, 2014; Kashyap, 2014; Rajwinder, 2014] is algorithm works almost the same way as in the algorithm min-min. The main difference is: in this

algorithm first finding out the minimum time perform tasks, select the maximum value that is the maximum time of all the tasks on all resources. Further task with founded maximum time is assigned for execution specifically on the selected node. Then execution time of all tasks is calculated on that node by adding the execution time of task to execution time of other tasks on that node. Then, set task is removed from the system.

8. **Honeybee Foraging Behavior** [Ghuge, 2014; Rajwinder, 2014] is decentralized algorithm that helps to achieve increased throughput and global load balancing by using of local actions server. Actual VM load calculate, after that a VM condition is solved: or it is overloaded or underloaded or balanced. VM are grouped in accordance with the current load. The priority of the task that awaits in the VM, is taken into account after removing it from the overloaded VM. Then, the task is assigned to underloaded VM. Previously taken task is useful for finding underloaded VM. These problems are known as bee intelligence agents in the next step. The algorithm reduces response time of VM and waiting time of task. System performance is enhanced with raise the heterogeneity of the system. The main problem is that the bandwidth does not increase with raising system size. Algorithm is most appropriate when is necessary diversity of species services.

9. **Active Clustering** [Gupta, 2013; Rajwinder, 2014]. In this algorithm the same components of the system are grouped together and they work in groups. It works as in the technique of self-assembled load balancing, where the network is rewired for load balancing system. The system is optimized using a similar assignments work by connecting these services. System performance is enhanced with improved resources. Bandwidth is improved by effectively utilizing all the resources.

10. **Compare and Balance** [Hu, 1998; Gupta, 2013; Rajwinder, 2014] is used to reaching the equilibrium state and managing of load balanced system. In this algorithm, based on the probability (the number of virtual machines running on the current host, and the whole cloud system), the current host randomly selects the host and compares their loads. If the load of current host is more than one of selected host, it sends an additional load on this particular node. Then, each node of the system carries out the same procedure. This load balancing algorithm is also designed and implemented to reduce the time migration of VM. Shared memory is used to reduce the time migration of VMs.

11. **Lock-free multiprocessing solution for LB** [Liu, 2013; Ghuge, 2014; Rajwinder, 2014] it offered unblocking multiprocessor load balancing solution that excludes the use of shared memory in contrast

to other multiprocessor load balancing solutions that use shared memory and locking to maintain the user's session. This is achieved by modifying the kernel. This solution helps to improve the overall performance load balancing in multicore environments by running multiple processes of load balancing in a single load balancer.

12. **Ant Colony Optimization** [Mishra, 2012; Dhinesh, 2013; Katyal, 2013; Rajwinder, 2014] is distribution algorithm. In this algorithm, resource information is dynamically updated with every move of ants. Multiple ant colonies are described so that a node sends colored colonies throughout the network. Painted ant colonies are used to prevent movement of ants from the same slot following by one route, and to ensure their distribution across all nodes in a system where each ant acts as a mobile agent, which carries an update load balancing information in the next node.

13. **Shortest Response Time First** [Singhal, 2011; Liu, 2013; Kashyap, 2014]. The idea of this algorithm is a direct forwarding. A priority is assigned to each process for run it. In processes with equal priorities planned FIFO order. SRTF algorithm is special case of the general priority scheduling algorithm. The priority is the reverse of the next burst of processor (CPU) in the SRTF algorithm. This means that if the burst processor increases, the priority is lowered. SRTF policy selects the task with the shortest processing time. In this algorithm, short tasks are done before long one. The SRTF is very important to know or estimate the processing time of each job and this is the main problem of SRTF.

14. **Based Random Sampling** [Singhal, 2011; Raghava, 2014] has approach of scalable and distributed load balancing, which uses a random sample of the system domain to achieve self-organization, in this way the load is balancing between all nodes in the system. System performance is improved by increasing the amount and similarity of resources, which leads to increased throughput by efficient using more scope of system resources. However, the algorithm gets worse with increasing variety resources.

15. **The two phase scheduling load balancing algorithm** [Singhal, 2011; Ghanbari, 2012; Katyal, 2013]. This combination OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms, which uses a high performance implementation and system support load balancing. OLB keeps every node in operating condition to achieve the purpose of load balancing. LBMM scheduling algorithm is used to minimize the execution time of each task on the node, thereby

minimizing the total completion time. This algorithm is used to improve the efficiency of resource utilization and increases efficiency.

16. **Active Clustering load balancing Algorithm** [Hu, 1998; Singhal, 2011; Katyal, 2013]. The algorithm optimizes task by connecting similar services using the local rewiring. Work by grouping similar nodes. The process of grouping is based on the concept referee node. Referee node defines connection between the neighbors, which is similar to the initiating node. Then, a referee node breaks the connection between itself and a primary node. Next set of processes repeates again and again. System performance is increased by high availability of resources, because of this bandwidth also increases.

17. **ACCLB** [Singhal, 2011] is load balancing method based on ant colony and the theory of complex network (ACCLB) in open cloud computing. It uses low-level specifications and scaleless complex network to gain a better load distribution. This technique allows to overcome nonuniformity, is adaptive to a dynamic environment, is excellent in fault tolerance and has good scalability, consequently, helps to improve system performance.

18. **Decentralized content aware** [Randles, 2010; Singhal, 2011] is load balancing method, referred to as the workload and client notification policy (WCAP). This method uses a parameter called the USP for guidance unique and special properties of requests, as well as compute nodes. USP helps the planner to decide about the most appropriate node for processing tasks. This strategy is implemented on a decentralized basis with low costs. Using the content information to narrow the search, the search performance overall system is improved and downtime of computing nodes is reduce, thus improving their utilization.

19. **Server-based LB for Internet distributed services** [Randles, 2010; Singhal, 2011] is load balancing method for web services, distributed throughout the world. It helps in reducing the response time of the service by using protocol, which limits the forwarding of requests to the closest remote servers without overloading. For the implementation of this protocol is typically of middleware. It also uses heuristics to help web servers to withstand overload.

20. **Join-Idle-Queue** [Singhal, 2011, Gupta, 2013] is load balancing algorithm for dynamically scalable web services, provides a large-scale load balancing at distributed senders. At first it calculates the

availability of idle processors in each sender, and then assigns the task to processors to reduce the average length of the queue for each processor. When removing load balancing task from the critical path request process the algorithm effectively reduces the system load, it does not assume any communication load on the newly arrived task and does not increase the actual response time.

21. **Token Routing** [Ray, 2012] has the main goal to minimize the cost of the system by moving the markers within the system. But in a scalable a cloud system, agents can not have enough information to spread the workload due to communication bottlenecks. So load distribution between agents is not fixed. The disadvantage of this algorithm can be removed using a heuristic approach load balancing based marker. This algorithm provides a quick and effective solution to the routing. In this algorithm, agents do not need to have knowledge about the global state and workload neighbor. In order to make decisions on marker transfer agents actually build their own knowledge base. This knowledge base is derived from previously received tokens. Thus, in this approach there are no communications overhead.

22. **Central queuing** [Rajwinder, 2014] operates on the principle of dynamic allocation. Each new task comes to the queue manager and is queued. When a queue manager is received a request to perform a task, it removes the first task from the queue and sends it to the requester. If a queue does not have ready tasks, the request is buffered until the new task will not be available. But in the case of recording a new task in a queue until there are unanswered questions in the queue, the first such request is removed from the queue, and a new task put before it. When the CPU usage falls below the threshold value then the local boot manager sends a request for a new task to the central download manager. Then, the manager responds to the request, if finished task is found, otherwise complied with the order request before the new task.

23. **Connection mechanism** [Ray, 2012; Gupta, 2013; Liu, 2013]: load balancing algorithm may also be based on the mechanism of smallest amount connections, which is part of the dynamic scheduling algorithm. It is required for counting the number of connections for each server of dynamic load estimation. The load balancer writes the number of connections per server. Number of ports is increased when the new connection is sent to the server, and decreases when the connection is terminated or interrupted.

24. **Least connections** [Mishra, 2012; Kashyap, 2014] algorithm sends requests to the server, which is currently served by the smallest number of connections. The load balancer will monitor the number of connections of server and send the following request to the server with a minimum of connections.

5. LOAD BALANCING METHODS

In this section are described main operation load balancing methods that are available in modern balancers and/or can be configured on the servers.

5.1 Direct Routing

Direct routing mode (DR) is a high-performance solution with a slight modification in the existing infrastructure, and it works by changing the MAC-address of incoming packet on-the-fly, which works very quickly (Figure 11) [Roth, 2008; Бажин, 2010; Turnbull, 2015].

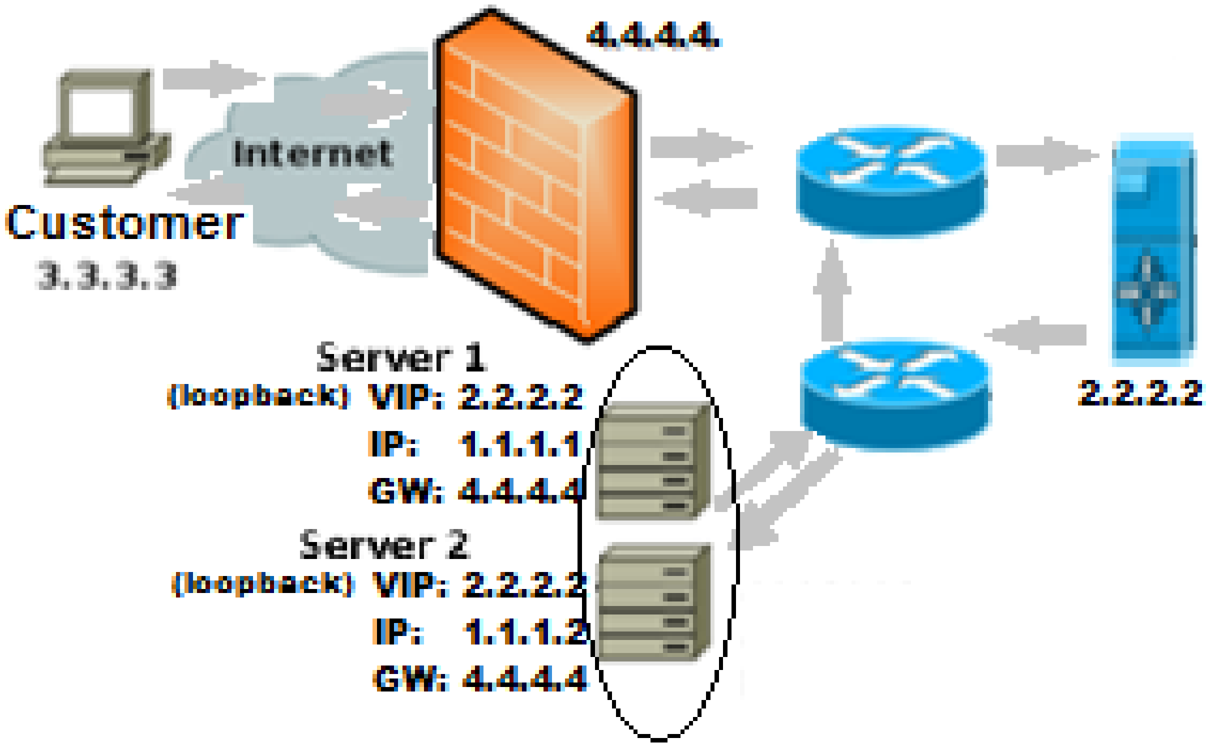


Figure 11. Example of direct routing load balancing method

I.e. it means that when the packet comes to the real server, it expects that server has a virtual IP, since it is necessary to assure the real server to respond to VIP, but do not respond to Address Resolution

Protocol (ARP) requests. Direct routing mode is enables for servers to access the network with VIP or real IP-address, without requiring any additional subnets or routes, but the real server must be configured to respond both VIP and it own IP-address.

5.2 Network Address Translation

Sometimes it is impossible to use the DR mode: either because the application can not communicate to RIP and VIP in the same time, or because the host operating system can not be modified to respond to ARP requests [Roth, 2008; Бажин, 2010; Turnbull, 2015]. In this case, it is possible to use the mode Network Address Translation (NAT) (Figure 12), which is also a high-performance, but requires infrastructure changes in internal and external subnet.

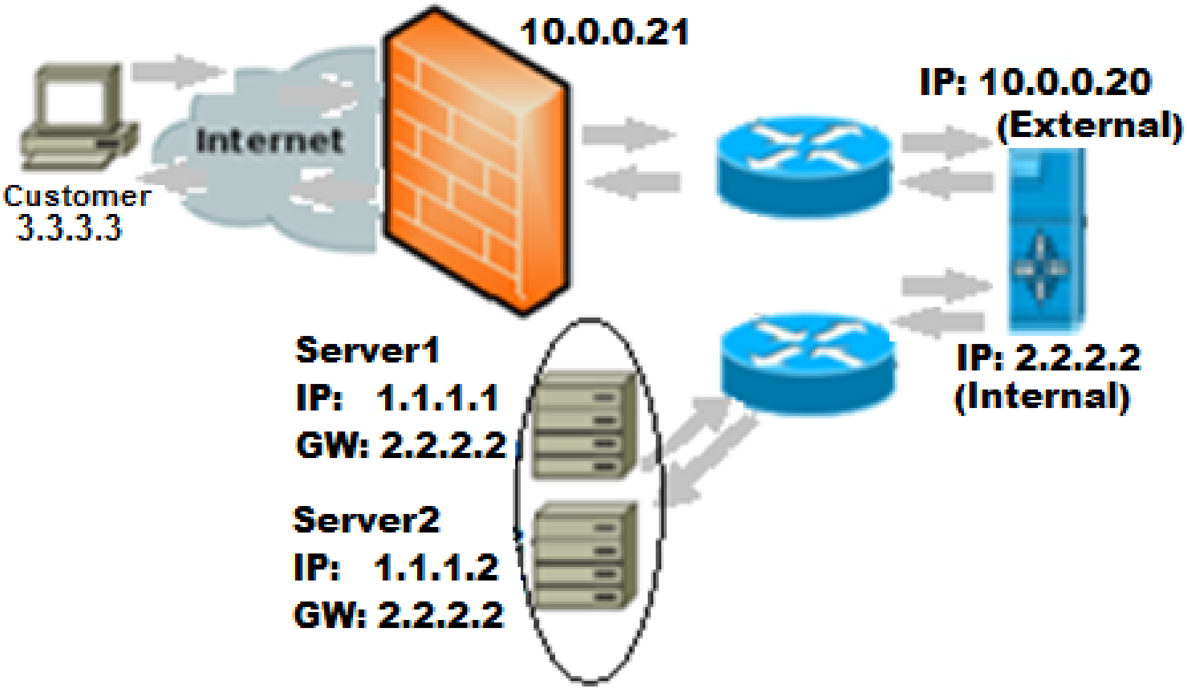


Figure 12. Example of Network address translation load balancing method

In this mode the load balancer translates all requests from external virtual server on internal real servers and real servers must have their default gateway, which are configured so that it points to load balancer. If real servers must be accessible by their own IP-address for a non-load balanced services, i.e. SMTP, it will be necessary to install individual firewall rules for each real server.

5.3 Source Network Address Translation

If the application requires the load balancer to handle cookie insertion, it is necessary to use Source Network Address Translation (SNAT) configuration, which does not require any changes to the application servers. However, since the load balancer acts as a full proxy, it does not have the same capacity as in the previous methods (Figure 13) [Roth, 2008; Turnbull, 2015].

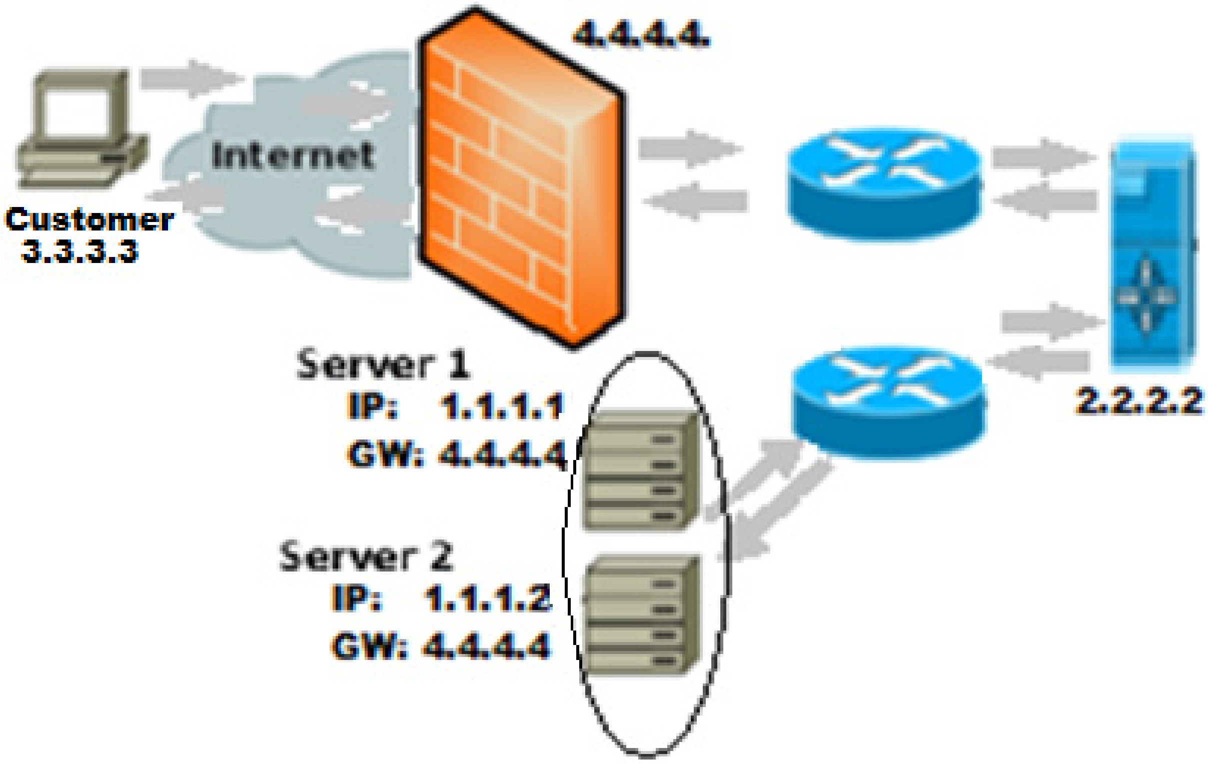


Figure 13. Example of Source Network address translation load balancing method

The load balancer proxies the application traffic to the servers so that the source of all traffic becomes the load balancer.

5.4 Transparent SNAT

If the source address of the client is a requirement then the balancer can be forced into transparent mode requiring that the real servers use the load balancer as the default gateway (as in NAT mode) and only operates for directly attached subnets (also as in NAT mode) [Roth, 2008; Turnbull, 2015].

5.5 SSL Termination or Acceleration

All load balancing methods of 4 and 7 levels can process SSL traffic on their level, i.e. internal servers do the decryption and encryption of traffic [Roth, 2008; Mendonca, 2014; Turnbull, 2015]. However, in order to check the HTTPS traffic, to read or insert cookies there is a necessity to decode (terminate) SSL traffic balancer, and this can be done by importing secure key and certificate to the load balancer, giving to it the right for decrypt traffic .

5.6 TCP/IP server load balancing

The tunnel mode looks like the Direct Server Return mode, except that traffic between the load-balancer and the server can be routed. The load-balancer encapsulates the request in IP tunnel to the server. The server recovers the client request from the load balancer, process it and forward the response directly to the client.

TCP/IP server load balancers operate on low-level layer switching [Roth, 2008]. The real servers appear to the outside world as a single "virtual" server. The incoming requests on a TCP connection are forwarded to the real servers by the load balancer, which runs a Linux kernel patched to include IP Virtual Server (IPVS) code.

To ensure high availability, a pair of load balancer nodes is set up, with one load balancer node in passive mode. If a load balancer fails, the heartbeat program that runs on both load balancers activates the passive load balancer node and initiates the takeover of the Virtual IP address (VIP). While the heartbeat is responsible for managing the failover between the load balancers, simple send/expect scripts are used to monitor the state of real servers.

Transparency to the client is achieved by using a VIP that is assigned to the load balancer. If the client issues a request, first the requested host name is translated into the VIP. When it receives the request packet, the load balancer decides which real server should handle the request packet. The target IP address of the request packet is rewritten into the Real IP of the real server.

5.7 Hashing

The load balancer on the 4-th level distributes a load across all available virtual machines by calculating the hash function of the traffic that has entered to this endpoint. This hash function is calculated such that all packets received within one connection (TCP or UDP) send to same server. Load balancer uses a set of 5 fields (IP address source, source port, IP destination address, destination port, protocol type) to calculate the hash used in the comparison of traffic and available server. Moreover, the hash function is selected in a way that distribution of servers connections are sufficiently random. However, depending on the type of traffic is acceptable that various connections are attached to the same server. Basic hash function allows get good distribution of requests at sufficiently large number of them from various sources [Roth, 2008; Mendonca, 2014; Turnbull, 2015].

Shortcoming of this algorithm is that it must be quite difficult to be able to distribute the load to other servers in the case of exclusion or inclusion of servers.

5.8 Caching

Caching is technology that introduces intelligence element to the concept of store and forward data to determine which information resources are copied from the core to the periphery of the network and how often they are updated. The caching technology is based on the fact that it is cheaper and more efficient to store data than to transmit. In fact, this idea is not new: in computers and other electronic computing devices, data is stored locally for reducing access time to them, the same principle is implemented in Web browsers, where the last viewed Web pages are cached on user hard drive [Meyer, 1998; Roth, 2008; Yucesan, 2011;Turnbull, 2015].

Caching removes some of the load from the overloaded Web sites and protects them from sharp traffic fluctuations.

Three configurations are most popular: the cache is placed near the router and processes traffic by Web cache control protocol (WCCP); cache combined with the switch Level 4 or above, controls the traffic; cache is embedded in a layer 2 switch or Layer 3, and traffic control is transferred to a separate load balancer.

Based on load-balancing scheduling algorithm user session requests are handled by different servers. If the cache is used on the server side, wandering requests will be a problem. In this situation, there is a method in which the cache is placed in a global space. Memcached is a popular solution of distributed caching, which provides a large cache on multiple machines. This is a distributed cache that uses sequential hashing to determine the cache server (daemon) for the cache entry. Based on the cache the hash code in the client library always displays the same hash code for the same address of the server cache. Then this address is used to store the cache entry. Figure 14 illustrates this approach cache.

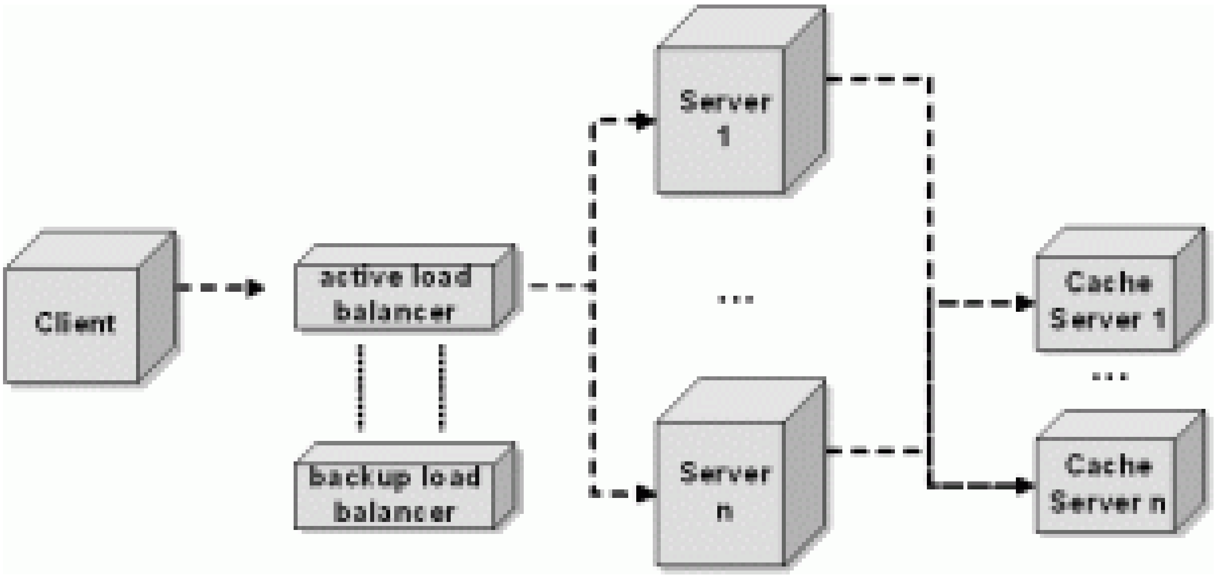


Figure 14. Load balancer architecture enhanced by a partitioned, distributed cache

Cache methods are divided into two types [Cao, 1996; Meyer, 1998; Forney, 2002; Angrish, 2011; Савчук, 2012]:

1) cache-unaware. This kind of strategy does not take into account the possible difference between the data that are cached locally on each server. Such strategies work well in two cases:

If the server does not cache anything locally (stateless servers). For example, when data load from a remote resource on the network requires less CPU resources and time of same data load from the local cache. Such situation is possible if the cache hit ratio for the local cache aspires to zero due to the large scope of data, which needs to be cached. It is also possible with the frequent updating data, so that the next request have out of date.

If local caches on all servers contain the same data. In this case, no matter where are sent the next requests. For cache-unaware strategies the best is distribution of load on the least loaded server at this moment. This strategy allows to achieve the lowest latency in the request queue.

2) cache-aware. This kind of strategy sends requests so as to maximize the amount of cache hits in local caches of servers.

The most famous of these strategies is sticky load balancing (SLB). It is based on fair assumption that the requests from the one user are needed of the common data which are amenable to local cache on the server. Such data may include user settings and data that have meaning only for a particular user. It is obvious that the direction of requests from the one user on the one server allows to maximize the cache hit ratio.

This strategy works properly with the following conditions:

- If the user performs more than one request to the server for a short time.
- When processing requests from one user a server needs the one data that are specific to that user, and these data require large computing resources or consume a lot of network traffic when pull them out of the remote services.

Comparing cache-aware and cache-unaware distribution strategies requests.

Advantages of the cache-aware strategy:

- Well-optimized design (i.e. actively using local caches to minimize the cost of CPU time and network traffic) working on a single server is much easier to migrate to multiple servers by using cache-aware load balancing.
- It reduces server load and increases the number of requests that can be processed by each server per unit time, as no need to waste CPU time to generate the data, if they are already present in the local cache. Also, as opposite to shared cache, the local cache may contain a prepared data that do not need to waste the network traffic and CPU time on serialization before writing and deserialization before reading.
- It reduces the load on external data sources and the network between servers and external data sources, as it does not need to pull the data if they are already present in the local cache.
- It reduces the time of request processing, as it does not need to wait for a response from the external data sources if they are already present in the local cache.
- It reduces the total amount of memory required on the local caches, as data cached on different servers almost are not doubled. On the other hand, it gives opportunity to cache more different data in a fixed amount of local cache, thereby increasing the effective cache size.

Shortcomings of a cache-aware strategy:

- Higher average waiting time in the queue requests compare to the round robin and least loaded at the same average server load. This shortcoming is offset by the fact that cache-aware strategy usually can process more requests per unit of time at a comparable server load compared to the cache-unaware strategies due to the above advantages.
- More complex synchronization of local caches in comparison with a shared cache. If data is cached only in the shared cache, the current request can be processed on an arbitrary server, as current data for the user is always possible to try to pull out of the total cache.

In cache-aware strategists a same data may be out of sync if the group of requests gets for a short time on the "foreign" server, and then extend to "own" server. This is possible in the case of short-term false "failure" of one of servers, which quickly returns to operation without loss of the local cache.

It should not rely on the safety of data in the cache, because at any moment they can be lost. This can happen in various ways - for example, the cache has grown to enormous size and its need to compress one by removing out any data. Or service responsible for caching is fail. Critical data of user sessions are not recommended to keep in the local cache without placing them in storage, guaranteeing their safety.

5.9 DNS Load Balancing

This is the easiest load balancing method, the essence of which is that it creates multiple DNS-record type A for the domain record on DNS-server [Roth, 2008; Бажин, 2010; Natario, 2011; Turnbull, 2015]. DNS server issues recording-type A in an alternating cyclic basis.

Usually clients' resolver is programmed in a way that a customer's caching DNS-server does not affect to balance. Also the client chooses random entries from received one (Figure 15). Accordingly, there is a connection to the appropriate server.

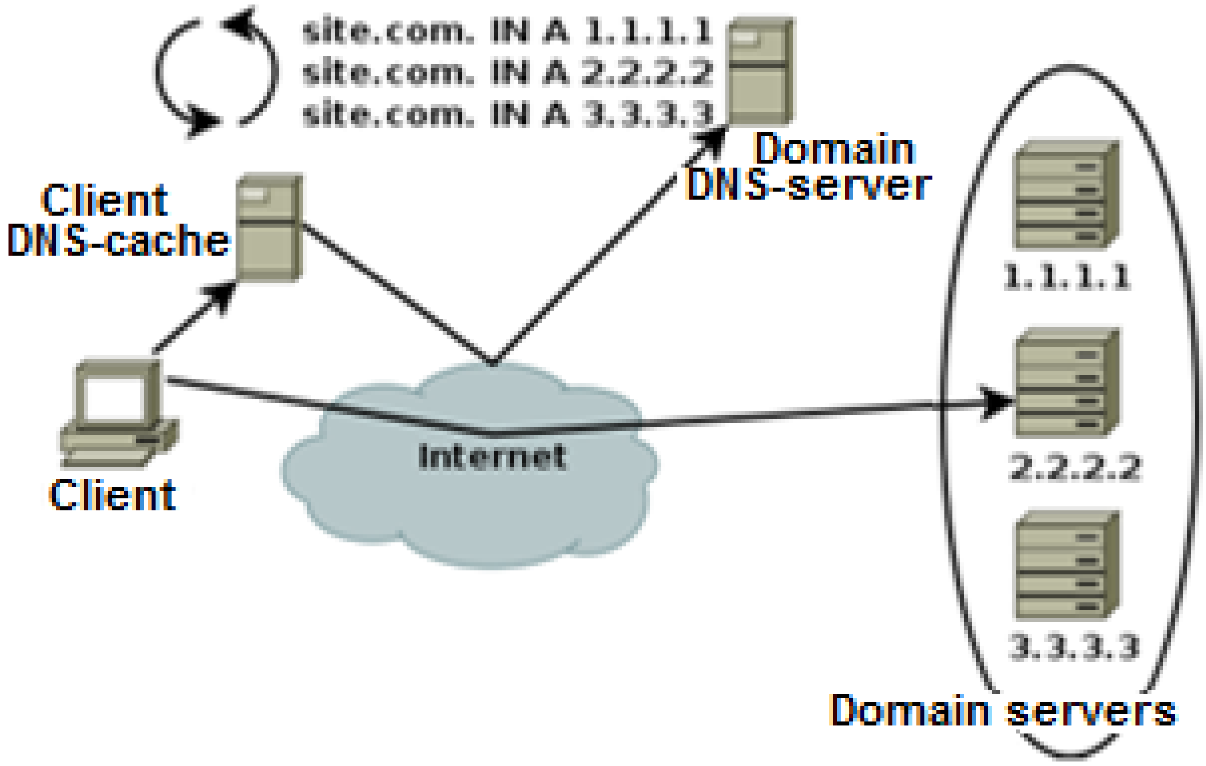


Figure 15. DNS load balancing

To implement this method any DNS-server is suites perfectly.

Advantages of the method:

• It does not depend on a high-level protocol, i.e. this method can be used by any protocol, which handle to the server by name.

• It does not depend on the server load: because there is caching DNS-server, it is no difference how many customers are: one or millions.

• No communication between servers, so it can be used for local balancing (this balancing of servers within the same data center), and for the global balancing when there are multiple data centers, where servers are not connected with each other.

• Low cost solutions.

Shortcomings of the method:

• It is difficult to disconnect the servers that do not respond or have failed: because of the cache entry is deleted only after the time that is specified by TTL (Time To Live), or when forced caching is more longer.

• Load balancing between servers in the correct proportions is difficult.

• Clients, which concentrated in certain areas, provide the load on one server. This can give a large nonuniformity in the distribution of the load among servers.

• Maximum numbers of IP-addresses that can be balanced is limited.

•Calls of TCP from clients are not open on all DNS-servers.

5.10 Network Load Balancing

Microsoft's Network Load Balancing (NLB) – implementation is based on software that runs on each node in the cluster using a hashing algorithm that takes IP-address, or IP-address and port from incoming request and determines which node (host) cluster will handle the request. Each node in the cluster receives each packet traffic and determines which node is responsible for the reaction perform by applying a filter to each packet, in this way, only one node will eventually have service the request [Natario, 2011; Turnbull, 2015].

NLB concept is quite simple: each server in the cluster of load balancing is configured with a virtual IP-address and this address is configured on all servers that take part in the cluster. Whenever a request is made to the virtual IP, the network driver on each of these machines intercepts the request for the IP-address and forwards the request to one of the machines in the cluster of load balancing on the basis of rules that can be configured for each server in the cluster.

NLB acts as a virtual network device with a private IP-address and a real device (the actual port Ethernet) associated with load balancing. Instead of using and the ports' IP addresses, the system will use the NLB software's IP address and this will ultimately result in the NLB software and its ports looks like a single device to the clients (Figure 16).

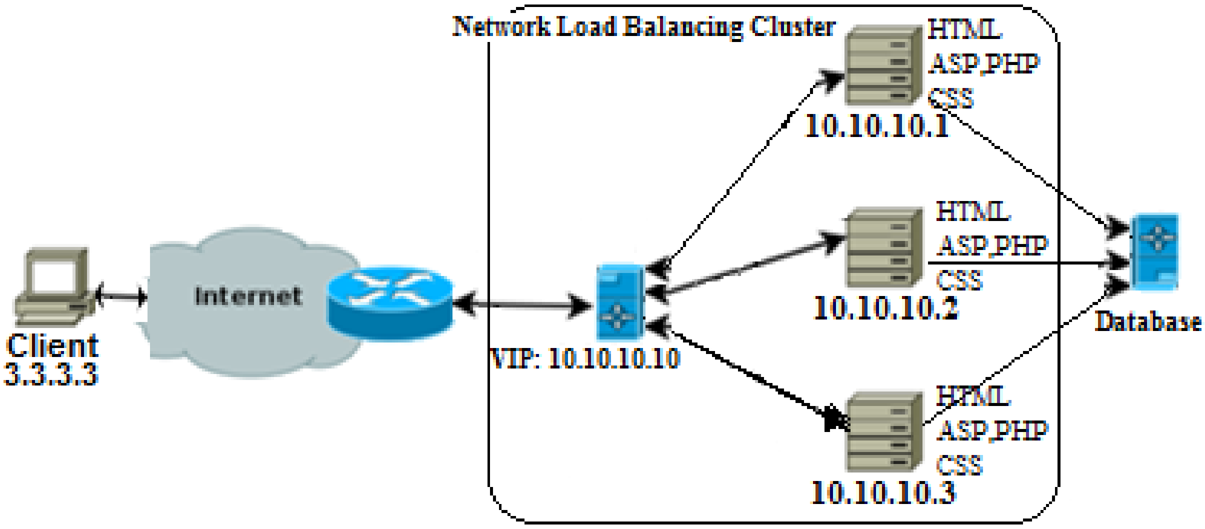


Figure 16. The usage of Network Load Balancing

5.11 Proxy method of load balancing

There is some proxy server that uses IP-address of our service, it receives a request, does something with it if it necessary and forwards it from his own to a necessary server (Figure 17) [Бажин, 2010; Red Hat, 2015]. This method is not transparent for the server, as it sees that it had been approached proxy [Roth, 2008; Red Hat, 2015]. To do this inside a high-level protocol we have to somehow send information about which client approaches us. For the HTTP protocol, we can add the header X-Real-IP with the client IP-address.

Advantages of proxy method:

- Because the proxy works at the protocol level, a proxy server can analyze and change our questions and answers. It allows to do bind a client to the server. For example, at value a specific setup cookie. This can be useful if we use local storage of the client session on the server.

- Proxy allows to distribute different types of requests to different servers. We can identify individual servers, which will "give" statically generated homepage, and do it with much less latency than the servers that handle "difficult" questions.
- It is possible to modify the response request. There can be done addition the title, for example, conversion. It is possible to cache the responses to this proxy.

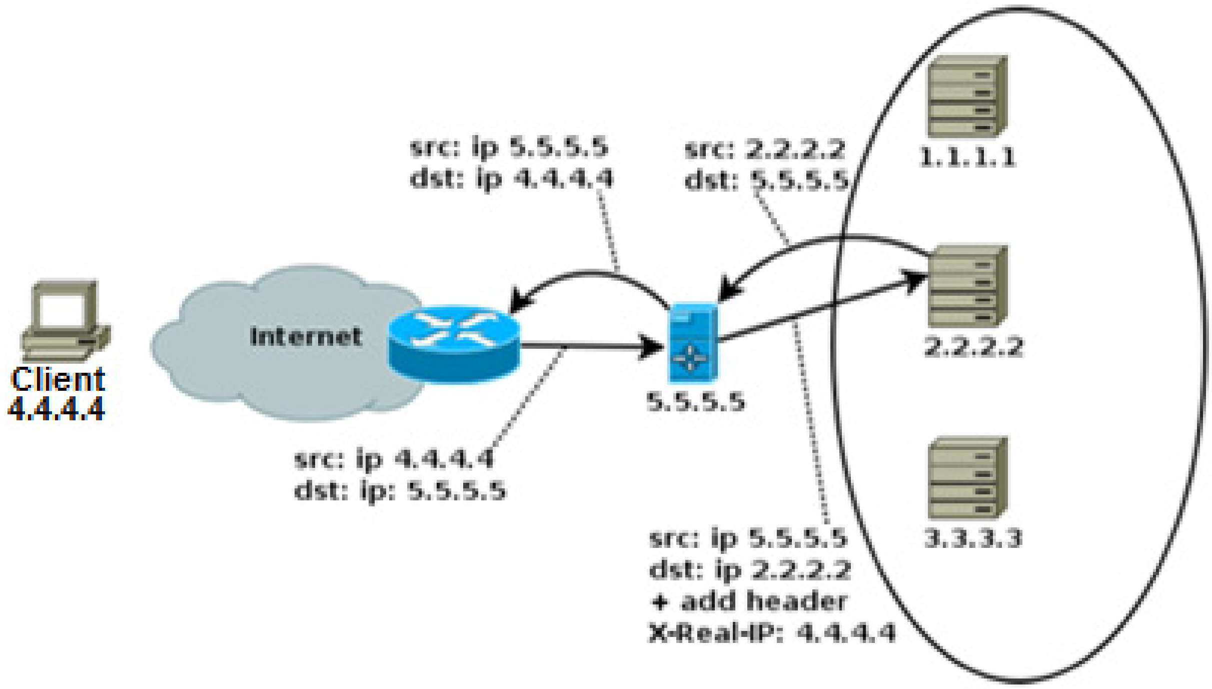


Figure 17. The usage of Proxy method

Shortcomings of the method:

- Proxy method has greatest resource consumption, as it involves higher-level protocols in comparison with other methods.
- Each protocol must have their own proxy type. Proxy cannot be implemented for everyone protocols in a way that it decides the required tasks.

5.12 Load balancing by using redirection

Another method of balancing is balancing redirects. Redirect is applicable to quite low number protocols [Red Hat, 2015]. Fortunately, it is applicable for HTTP.

There is some balancer that by referring to our service gives to client redirect to a specific server. In the case of HTTP this will look like a "HTTP redirect 302", the redirect code will appear as "Moved temporary".

Advantages of proxy method:

• If the request is enough "difficult", sometimes it makes sense to use a redirection even for global balancing.

• The method also allows to distribute various types of requests to different servers. Requests may be analyzed well.

Shortcomings of the method:

• It is applicable to a very low number of high-level protocols.

• On every client request two requests are made. One is to the redirector, one is to the server that handles the connection. This increases the time during which the client will receive the final answer to his request.

Conclusion

In this paper, a review of the main existing algorithms and methods for load balancing of distributed systems were carried out and the structure of load balancing methods for levels of OSI model are shown. The analysis of load balancing methods on different levels of the network model is carried out, the advantages and shortcomings of each method were given. The comparative analysis of hardware and software load balancing is dissected. Their differences, strengths and weaknesses are dissected. Also, description of the main features of load balancing algorithms, their advantages and shortcomings were presented.

Bibliography

[Angrish, 2011] R. Angrish, D. Garg. Efficient String Sorting Algorithms: Cache-aware and Cache-Oblivious. International Journal of Soft Computing and Engineering (IJSCE). – Vol1(2). – 2011. – P.12-16.

[Cao, 1996] P.Cao, E.W.Felten, A.R.Karlin, K.Li. Implementation and Performance of Integrated Application-Controlled File Caching, Prefetching, and Disk Scheduling. ACM Transactions on Computer Systems. – Vol.14(4). – 1996. – P.311-343.

[Cardellini, 1999] Valeria Cardellini, Michele Colajanni, Philip S. Yu. Dynamic Load Balancing on Web-server Systems. IEEE Internet Computing. - Vol.3. - No.3. – 1999.– P.28-39.

[Cardellini, 2001] V. Cardellini. A performance study of distributed architectures for the quality of web services. Proceedings of the 34th Conference on System Sciences. – Vol.10. – 2001. P.213-217.

[Chen, 2013] H. Chen, F. Wang, N. Helian, G. Akanmu. User-priority guided min-min scheduling algorithm for load balancing in cloud computing. National Conference Parallel Computing Technologies. - 2013. – P.1-8.

[Dhinesh, 2013] Dhinesh Babu L.D., P. Venkata Krishna, Honey bee behavior inspired load balancing of tasks in cloud computing environments. Applied Soft Computing. – Vol 13(5). - 2013. - P.2292–2303.

[Elzeki, 2012] O. Elzeki, M. Reshad, M. Elsoud. Improved max-min algorithm in cloud computing. International Journal of Computer Applications. - Vol. 50(12). – 2012. - P. 22–27.

[Erl, 2013] Thomas Erl, Ricardo Puttini, Zaigham Mahmood. Cloud Computing: Concepts, Technology & Architecture. Prentice Hall. Ed.1st. – 2013. – P.528.

[Erl, 2015] Thomas Erl, Robert Cope, Amin Naserpour. Cloud Computing Design Patterns. Prentice Hall. Ed.1st. – 2015. – P.592.

[Forney, 2002] Brian C. Forney, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau. Storage-Aware Caching: Revisiting Caching for Heterogeneous Storage Systems. Conference on file and storage technologies.– 2002. – P.61-74.

[Ghanbari, 2012] Shamsollah Ghanbari, Mohamed Othman. A Priority based Job Scheduling Algorithm in Cloud Computing. International Conference on Advances Science and Contemporary Engineering (ICASCE). –V. 50. - 2012. – P.778-785.

[Ghuge, 2014] Kalyani Ghuge, Minaxi Doorwar. A Survey of Various Load Balancing Techniques and Enhanced Load Balancing Approach in Cloud Computing. International Journal of Emerging Technology and Advanced Engineering. - Volume 4(10). 2014. – P.410-414.

[Gupta, 2013] Rohit O. Gupta, Tushar Champaneria. A Survey of Proposed Job Scheduling Algorithms in Cloud Computing Environment. International Journal of Advanced Research in Computer Science and Software Engineering. – Vol.3(11). – 2013. – P.782-790.

[Hong, 2006] Y.S. Hong, J.H. No, S.Y. Kim. DNS-based load-balancing in distributed web-server systems. Proceeding, in: Fourth IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WCCIA 2006). – 2006. – P.251-254.

[Hu, 1998] Y. Hu, R. Blake, D. Emerson. An optimal migration algorithm for dynamic load balancing. Concurrency: Practice and Experience. – V.10(6). – 1998. P. 467–483.

[Jiao, 2010] Yang Jiao, Wei Wang. Design and Implementation of Load Balancing of Distributed-system-based Web Server. Electronic Commerce and security. – 2010. – P.337 – 342.

[Kashyap, 2014] Dharmesh Kashyap, Jaydeep Viradiya. A Survey Of Various Load Balancing Algorithms In Cloud Computing. International Journal Of Scientific & Technology Research. - Vol.3(11). – 2014. - P.115-119.

[Katyal, 2013] Mayanka Katyal, Atul Mishra. A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. International Journal of Distributed and Cloud Computing. – Vol.1(2). – 2013. – P.6-14.

[Keshav, 1997] S.Keshav. An Engineering Approach to Computer Networking. Addison-Wesley, Reading, MA. – 1997. - P. 215-217.

[Kopparapu, 2008] Kopparapu Chandra. Load balancing servers, firewalls, and caches. Published by John Wiley & Sons, Inc. – 2002. – P.208.

[Koteswaramma, 2012] Rudra Koteswaramma. Client-Side Load Balancing and Resource Monitoring in Cloud. International Journal of Engineering Research and Applications (IJERA). - Vol.2(6). – 2012. - P.167-171.

[Laviol, 2014] Vitalij Laviol. Приборы с балансировкой нагрузки в системах сетевого мониторинга или «что такое Network Packet Broker». . – 2014. - URL: http://m.habrahabr.ru/company/metrotek/blog/259633/

[Liu, 2013] Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang, Bai-Nan Li. Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm. IJCSI International Journal of Computer Science. – V.10(1). - № 3. - 2013. – P.134-139.

[Mendonca, 2014] M. Mendonca, B.A.A. Nunes, X.-N. Nguyen, K.Obraczka, T. Turletti. A Survey of software-defined networking: past, present, and future of programmable networks. Communications Surveys & Tutorials, IEEE. – Vol.16(3). – 2013. – P.1617-1634.

[Meyer, 1998] Richard A. Meyer, Rajive Bagrodia Parsec. User Manual. Release 1.1. UCLA Parallel Computing Laboratory. - 1998. – URL: pcl.cs.ucla.edu/projects/parsec.

[Mishra, 2012] Ratan Mishra, Anant Jaiswal. Ant colony Optimization: A Solution of Load balancing in Cloud. International Journal of Web & Semantic Technology (IJWesT). - Vol.3. - No.2. - 2012. - P.335-338.

[Natario, 2011] Rui Natario. Load Balancing. – 2011. - URL: http://networksandservers.blogspot.com/2011/03/load-balancing-iv.html

[Pavan Kumar, 2012] Illa Pavan Kumar, Subrahmanyam Kodukula. A Generalized Framework for Building Scalable Load Balancing Architectures in the Cloud. International Journal of Computer Science and Information Technologies. - Vol.3(1). – 2012. – P.3015 – 3021.

[Raghava, 2014] N. S. Raghava, Deepti Singh. Comparative Study on Load Balancing Techniques in Cloud Computing. Open journal of mobile computing and cloud computing. – Vol.1. – No.1. – 2014. – P.18-25.

[Rajwinder, 2014] Rajwinder Kaur, Pawan Luthra. Load Balancing in Cloud Computing. Association of Computer Electronics and Electrical Engineers. – 2014. - P.374-381.

[Randles, 2010] Martin Randles, David Lamb, A. Taleb-Bendiab. A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops. – 2010. –P.551-556.

[Ray, 2012] Soumya Ray, Ajanta De Sarkar. Execution analysis of load balancing algorithms in cloud computing environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA). - Vol.2. - No.5. – 2012. - P.2657-2664.

[Red Hat, 2015] Red Hat Enterprise Linux. Обзор планирования распределения нагрузки. – 2015. - URL: https://access.redhat.com/documentation/ru-RU/Red_Hat_Enterprise_Linux/6/html/Virtual_Server_Administration/s1-lvs-scheduling-VSA.html.

[Roth, 2008] Gregor Roth. Server load balancing architectures, Part 1: Transport-level load balancing. – 2008. - URL: http://www.javaworld.com/article/2077921/architecture-scalability/server-load-balancing-architectures--part-1--transport-level-load-balancing.html.

[Singhal, 2011] Priyank Singhal, Sumiran Shah. Load Balancing Algorithm over a Distributed Cloud Network. 3rd IEEE International Conference on Machine Learning and Computing. – Singapore. – 2011. - P.37-42.

[Tuncer, 2011] D. Tuncer, M. Charalambides, G. Pavlou, N. Wang. Towards decentralized and adaptive network resource management. Network and Service Management (CNSM), IEEE. – 2011. – P.1–6.

[Turnbull, 2015] Malcolm Turnbull. Load Balancing Methods. – 2015. - URL: http://www.loadbalancer.org/blog/load-balancing-methods

[Vlaeminck, 2004] Vlaeminck K., Van Hoecke S., De Turck F., Dhoedt B., Demeester P. Design and implementation of an application server load balancing architecture supporting the end-to-end provisioning of value-added services. Telecommunications Network Strategy and Planning Symposium. - 2004. - P.345 – 350.

[Yucesan, 2011] Enver Yucesan, Yah Chuyn Luo, Chun-Hung Chen, Insup Lee. Distributed Web-based Experiments for optimization. Simulation Practice and Theory. – Vol.9. – 2001. - P.73-90.

[Zhihao, 2013] Zhihao Shang, Wenbo Chen, Qiang Ma, Bin Wu. Design and implementation of server cluster dynamic load balancing based on OpenFlow. Awareness Science and Technology and Ubi-Media Computing (iCAST-UMEDIA). - 2013. – P.691 – 697.

[Бажин, 2010] Алексей Бажин. Принципы балансировки Компании Mail.ru. – 2010. - URL: http://profyclub.ru/docs/21.

[Гуревич, 2010] Григорий Гуревич. Crescendo Networks - эволюция в мире WEB балансировки. – 2010. - URL: http://profyclub.ru/docs/99.

[Кириченко, 2011] Л.О. Кириченко, Э. Кайали, Т.А. Радивилова. Анализ методов повышения QoS в сетях MPLS с учетом самоподобия трафика. Системні технології. – 2011. – Вип. 3. – С. 52–59.

[Савчук, 2012] Игорь Савчук. Балансировка нагрузки сервера по методу SLB. – 2012. – URL: http://blogerator.ru/page/high-load-balansirovka-nagruzki-servera-po-metodu-sticky-load-balancing

Authors' Information

Igor Ivanisenko *– post graduate student, department of Applied Mathematics, Kharkiv National University of Radioelectronics; 14 Lenin Ave., 61166 Kharkiv, Ukraine;*

e-mail: ivanisenko79@yahoo.com.

Major Fields of Scientific Research: Modeling of Network traffic behavior, Self-similar traffic properties in Cloud technologies, Modeling of Queuing Systems, Computer networks

TECHNIQUE FOR ROAD AUTOMATED TRACKING WITH UNMANNED AERIAL VEHICLES

Samvel Hovsepyan

Abstract: *One of the most popular areas of unmanned aerial vehicles (UAV) use is the monitoring of roads and highways. UAVs are considered to be a low-cost and rapidly growing platform that can provide effective mechanisms for data collection and processing, especially in case of long distances. In this paper a new method for automated road monitoring with the help of UAV is offered. The method is suitable for determining image areas, where the heterogeneity compared with the general road structure is spotted. Also, algorithms for finding road cover from video shots and for determining the direction of the road are offered. The method is applicable for automated control of UAVs in order to find and track roads, as well as registration of various types of objects on the road. All the methods and algorithms were tested on a model and the results are shown.*

Keywords: *image processing, road tracking, road monitoring, similarity measure, UAV*

ACM Classification Keywords: *Image Processing and Computer Vision*

Introduction

Unmanned aerial vehicles (UAV) are widely being used in many fields. One of the most popular areas is the monitoring of roads and highways. The objectives of such monitoring are safety supervision, traffic and road conditions monitoring, road construction inspection, etc. For example appropriate researches had been done in the 2000s in the transport department of Ohio [Coifman 2004], and California [Frew 2004]. They used UAVs equipped with cameras and autonomous navigation systems.

Currently the UAVs are considered to be a low-cost platform that can provide effective mechanisms for data collection and processing. In particular, these data may be used in road traffic monitoring, which is important for companies that are engaged in transportation. Conventional traffic data collection methods [Zhou 2013] are based on a fixed infrastructure, which controls the local area. Therefore, control over wide areas becomes an expensive and time consuming problem. For comparison, the UAV has the following advantages: (1) low cost of monitoring over long distances; (2) by changing only the sensors it is possible to carry out different types of monitoring.

This article examines two issues related to road monitoring: (1) identification of road sections from a video stream; (2) anomalous situations detection on the road, related to obstacles occurrences, the existence of cars, their clusters, etc.

There are many approaches to solve the problems of detection and tracking of roads in the literature. Most of these methods are based on color and / or structural (geometric) properties of a road. Among them more effective and reliable methods use combination of both characteristics [Wang 2004], [Siogkas 2007]. Thus, we find more prudent use of both types of information.

In [Frew 2004], the proposed method is based on representation of color characteristics of road surface by using the Gaussian mixture models (GMM). Then, in order to determine the road pixels in each frame, the probability of pixels are checked for compliance to etalon GMM. In [Rathinam 2007] there is suggested a method, that is based on learning of color and gradient characteristics of river areas. Accordingly, to represent etalon models Gaussian and gamma distributions are used. In [Rathinam 2008], the method relies on the study of boundary structures of road. Those parts of the road, that are not being recognized automatically by using primary analysis, are being filled, if there are two sections of the road, that can bridge with straight line towards the road direction.

Most advanced tracking methods, such as the mean shift [Comaniciu 2000], particle filter [Nummiaro 2003] and optical flow [Schunck 1981] are based on the appearance of structures that were originally described. Hence, these methods are for special classes of objects, such as faces, cars or pedestrians etc, where objects have common attributes. In our case, the problem is more general, more simplified. The goal is to find any type of object on road without further specification. Also the listed algorithms require a lot of time and resources for calculations that leads to difficulties in their use in real mode.

Based on the above analysis of the literature, it seems appropriate to bring the problem to a new methodology development for road tracking and monitoring with the help of UAVs, which will meet the following requirements;

- Be fast enough for the calculations during the flight time of the UAV
- Give accurate results about the road tracking.

These goals can be divided into three separate subtasks.

- Finding road parts from an input image
- Automatically tracking of the road by a given direction
- Finding of all objects on the road.

It is assumed, that as an input method receives a video stream taken from a UAV. Video stream may come as a consecutive set of scenes or through short videos intervals, which again can be represented as a set of consecutive scenes by using already written existing application libraries. Therefore, it can be assumed, that the work is in consecutive scenes processing.

To determine road surface a method is proposed, which is based on road structural and color characteristics. It is quite fast and by effectiveness does not differ from commonly accepted models based on GMM. And for road monitoring, method tracks changes in consecutive frames and on changed parts runs a new algorithm, which is based on road structural characteristics.

Roadway Finding

In the proposed approach, finding the road surface from a given image is divided into two parts.

In the first part, based on the method, used in [Asatryan 2015], a complete image segmentation and simplification is carried out. Then, form the received segments, candidate sites on the pavement are identified based on the color characteristics of the web. Fig. 1 shows an example of an image before and after segmentation and simplification.

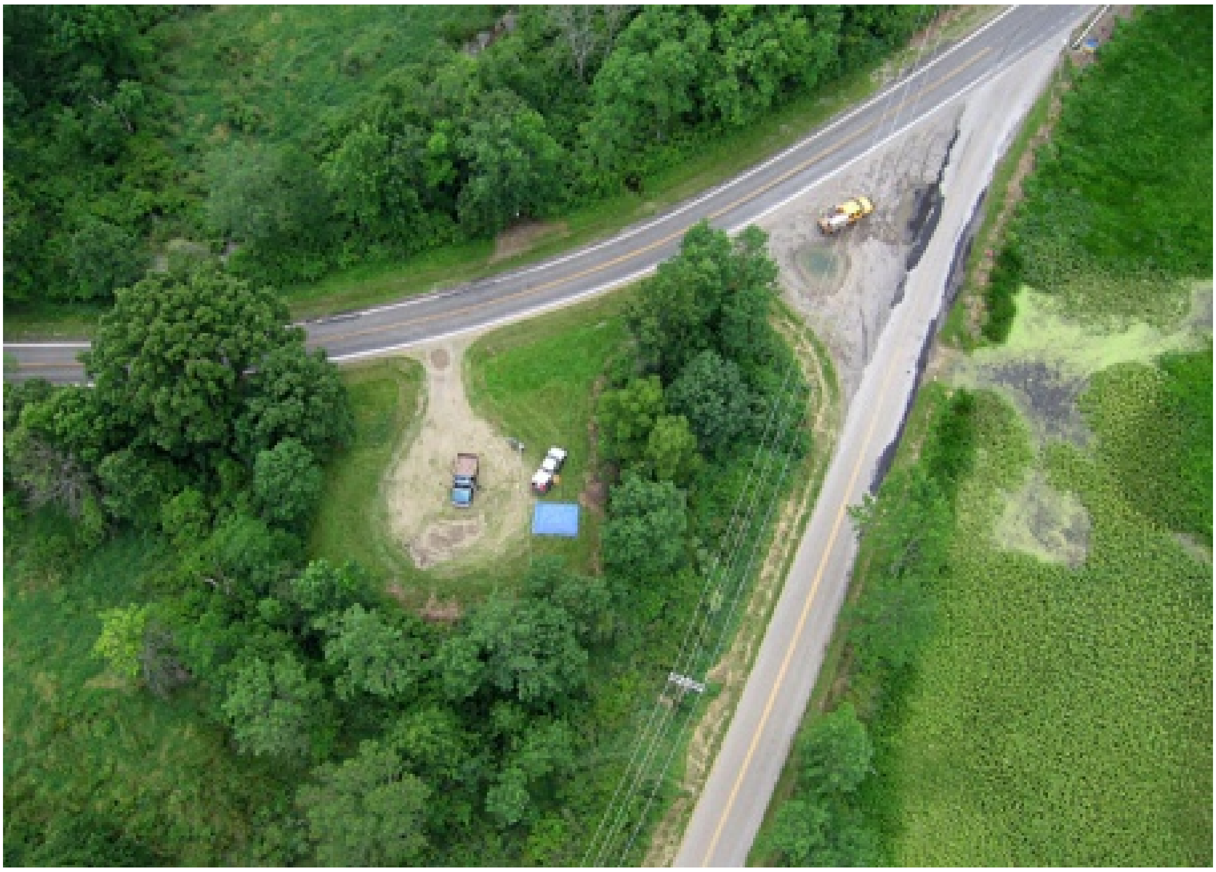


Fig. 1. (a)-original photo,

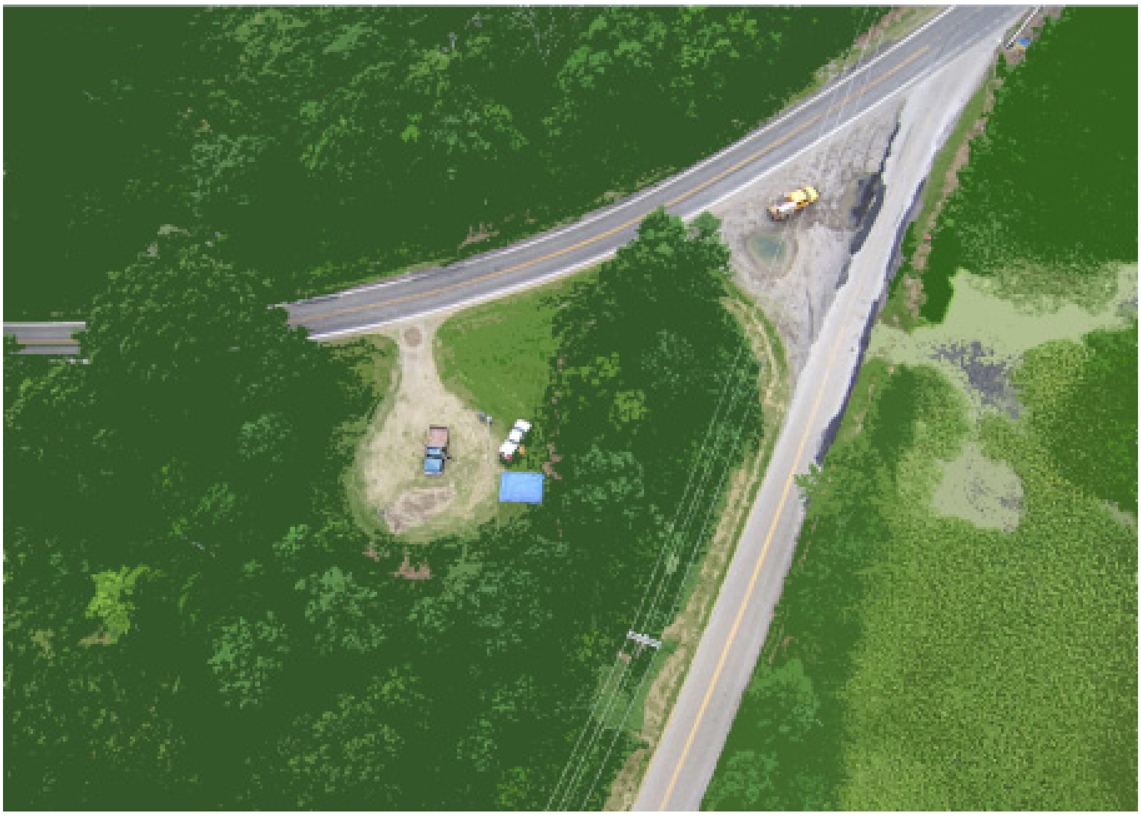


Fig. 1. (b) - photo after segmentation and simplification

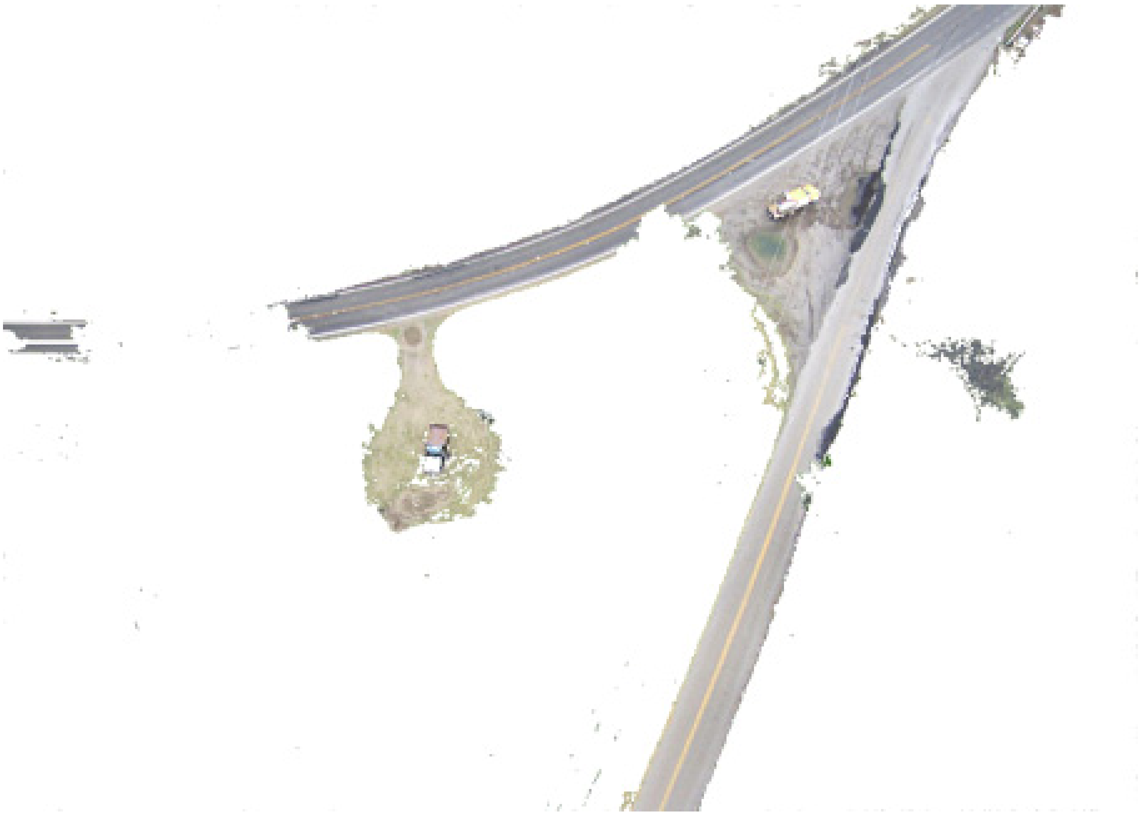
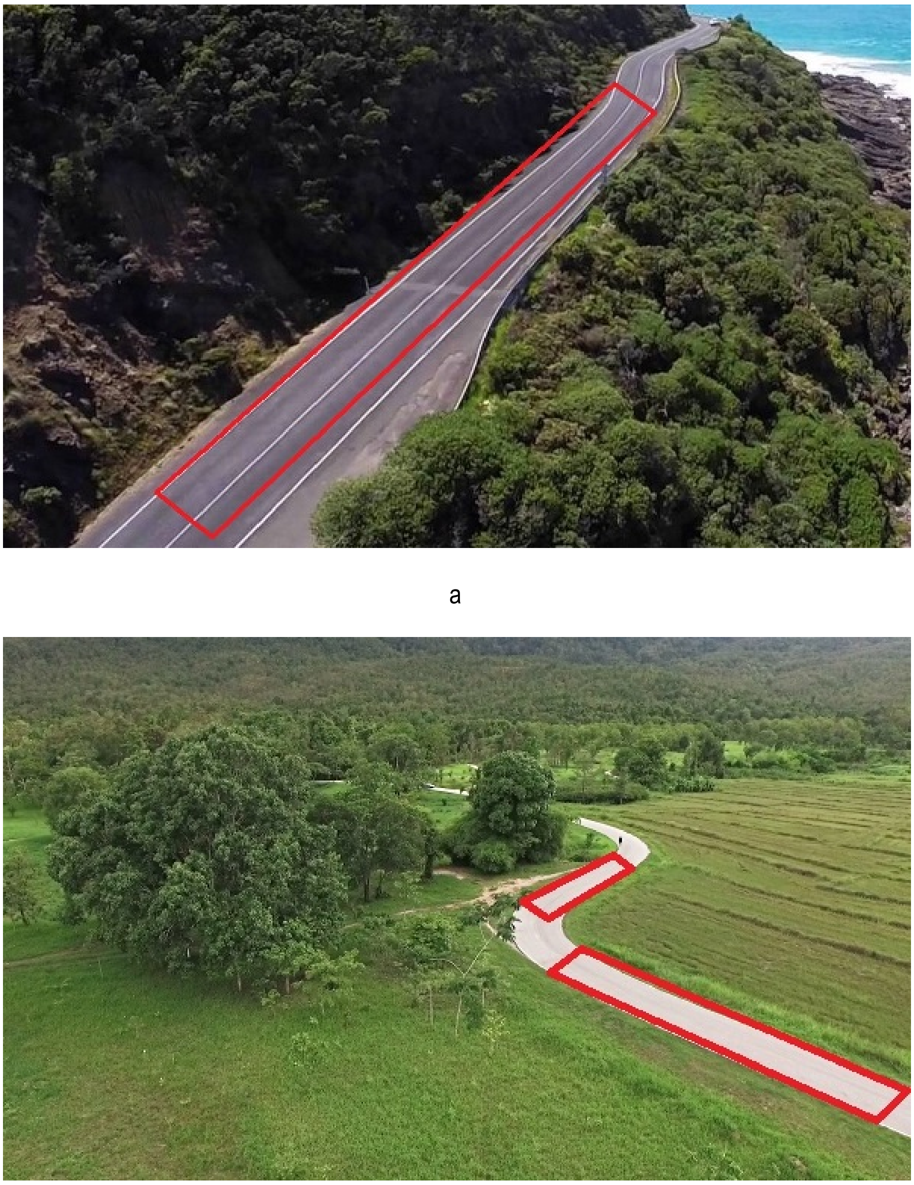


Fig. 1. (c) - only segments of asphalt

As can be seen form Figure 1c, by performing only color analysis, the image area, which has a color and texture resembling asphalt, can be confused as a road. To avoid these kinds of incidents, another analysis, which is connected to the structure of the road, is conducted. The segment of the image, having the form of a rectangle, the longest side of which coincides with the direction of the road, and the short coincides with its width, will be considered as a road (Figure 2). Therefore, from all the candidate segments, distinguished are those, that have a rectangular appearance.

It is important to estimate the width of the road, to detect smaller objects on the road (cars, animals) in the future.



a

b

Fig. 2. Images with emphasized rectangular roads

Automated Tracking of the Road

After road section detection, operator must direct the UAV to track the road in a chosen direction of the road. The task of the UAV is not to turn off from the road, and by tracking fly along it.

For simplicity let assume that a frame of the video stream is an image I = {I(m,n)} with the dimension M×N, where m=0,1,…,M; n=0,1,…,N and I(m,n) is the color characteristic of a pixel with coordinates (m,n) in RGB color space.

Let there are successive frames of the road sections, taken by a video camera of the UAV. Let this sequence through I1, I2, …, Ik. In Fig. 3 there is an example of such a sequence of video frames, where the time interval between frames is 1 second.

To enable the examination of the road it is necessary in each frame determine the direction of the road. In this work we follow the approach proposed in [Asatryan 2010] to calculate the dominant direction of the image, by using parameters of gradient field scattering ellipse components.

Let denote horizontal and vertical components of the gradient field through G_H(m,n) and G_V(m,n) respectively.

The equation of the scattering ellipse is being given by

$$\frac{1}{1-\rho_{HV}^2}\left[\frac{(g_H-\mu_H)^2}{\sigma_H^2}+\frac{(g_V-\mu_V)^2}{\sigma_V^2}-\frac{2\rho_{HV}(g_H-\mu_H)(g_V-\mu_V)}{\sigma_H\sigma_V}\right]=C^2 \tag{1}$$

where μ_H, μ_V, σ_H, σ_V are math expectations and MSE of $G_H(m,n)$ and $G_V(m,n)$ random variables, ρ_{HV} is the correlation coefficient between them, and C is a constant. Then, the angle α of an image orientation is being defined by the formula:

$$tg\alpha=\frac{2\rho_{HV}}{\sigma_H^2-\sigma_V^2+\sqrt{(\sigma_H^2-\sigma_V^2)^2+4\rho_{HV}^2}} \tag{2}$$

The components of the gradient field are measured using Sobel operator, and to calculate (1) and (2) formulas, the corresponding values of selected collections are being replaced.

The angle α which determined from the equation (2) is taken as the dominant orientation of the image.

For each frame, the dominant direction of the image is estimated and the UAV follows that direction. As the edges of the road are clearly distinguished in the image gradient magnitude, the discovered dominant direction of the image shows exactly the direction of the road.

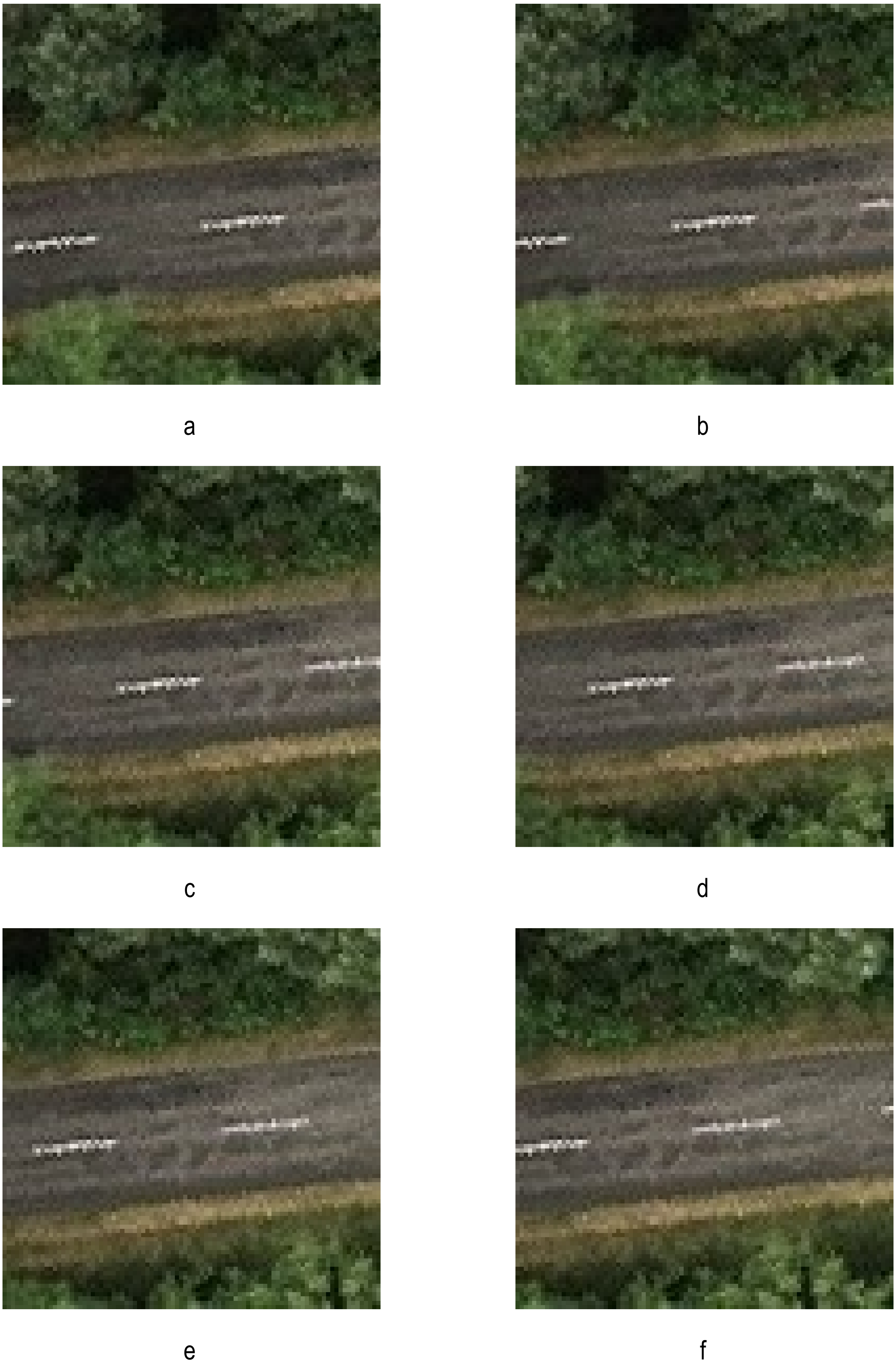


Fig. 3. The sequence of video frames of the road.

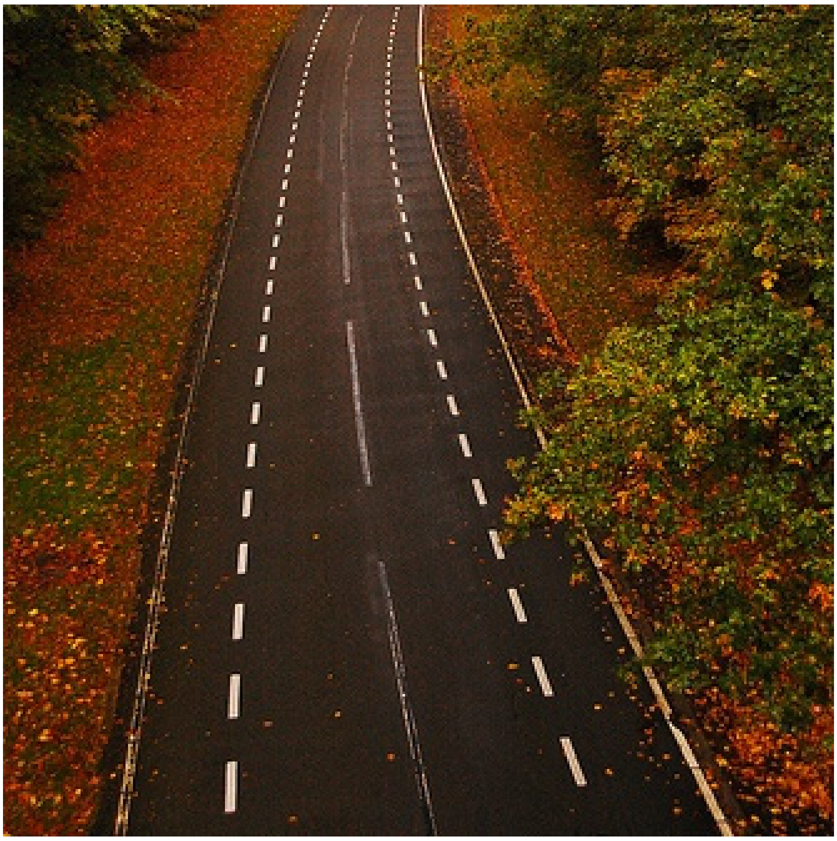


Fig. 4. (a) The image before applying the Sobel operator and the direction of the gradient field

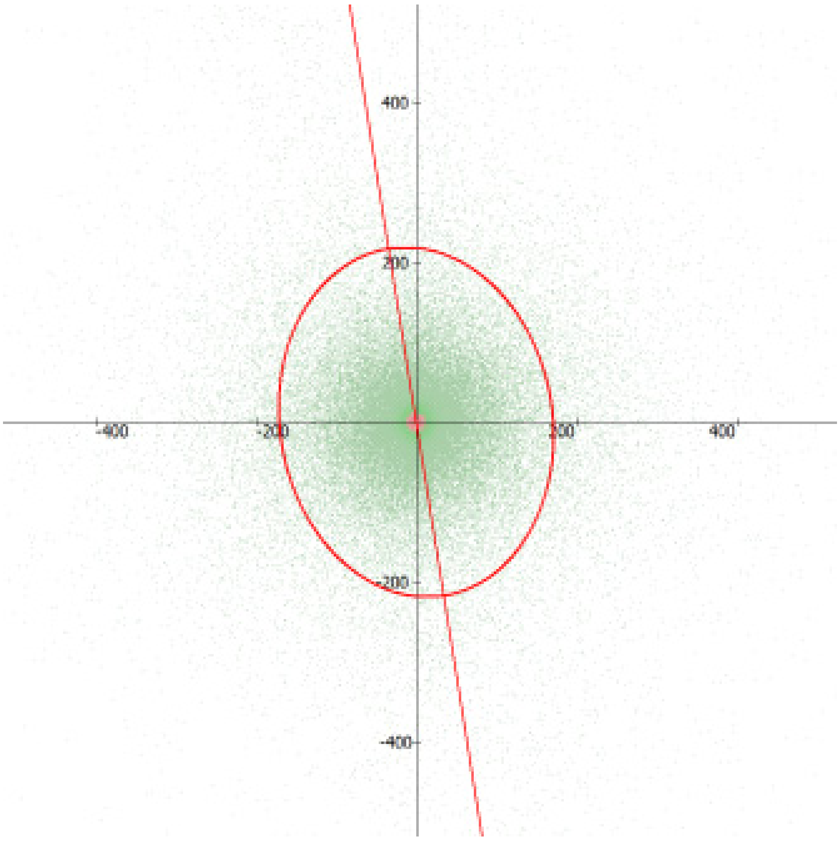


Fig. 4. (b) the Sobel operator and the direction of the gradient field.

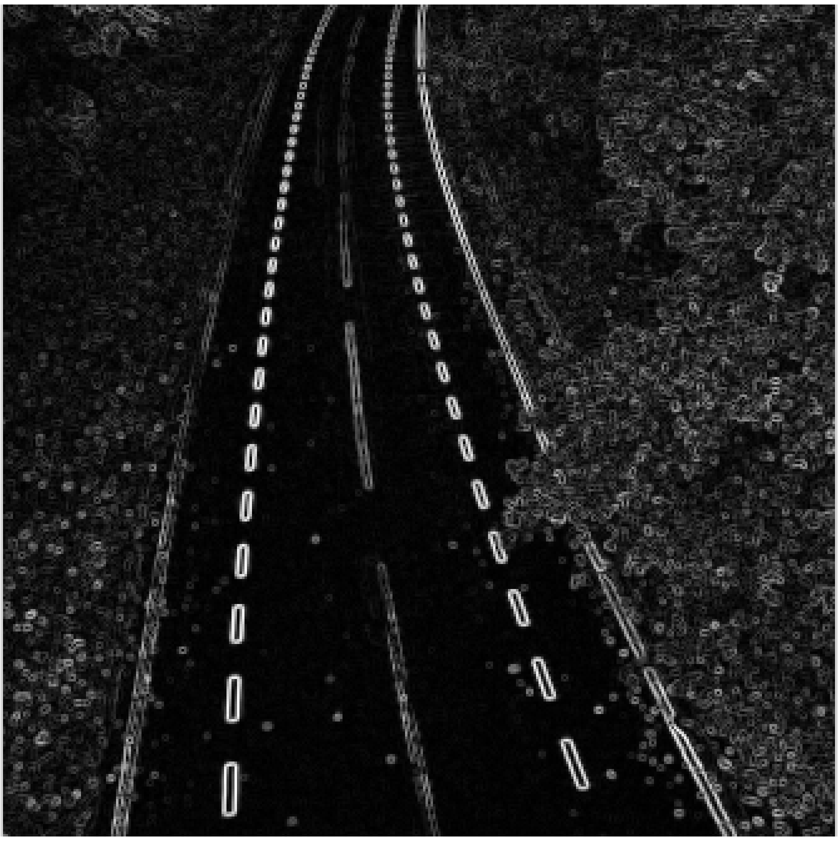


Fig. 4. (c) The image after applying the Sobel operator and the direction of the gradient field.
Fig. 4c shows an example of a gradient magnitude image, where the distinguished outskirts of the road are clearly seen.

Finding of Different Types of Objects on the Road

The proposed method for finding various objects on the road is based on a comparison of the current road segment with the etalon section of the road, which is automatically selected from those parts, where only the roadway is present. To control the correctness of the UAV course, for each road part the similarity of successive sections of the road is estimated. The similarity of the different sections of the road is estimated using measure W^2, proposed in [Asatryan 2009].

To identify whether there is an object on the road or is it other common part of the road, based on the value W^2, it is necessary to calculate such threshold value T, so that if $W^2 \geq T$, then we are on a different section of the road, otherwise there are another objects on the road, such as cars or animals. For this, initially three different sections of the same road are selected then between them W^2 values are being calculated, and the arithmetic mean of these values is being taken for T.

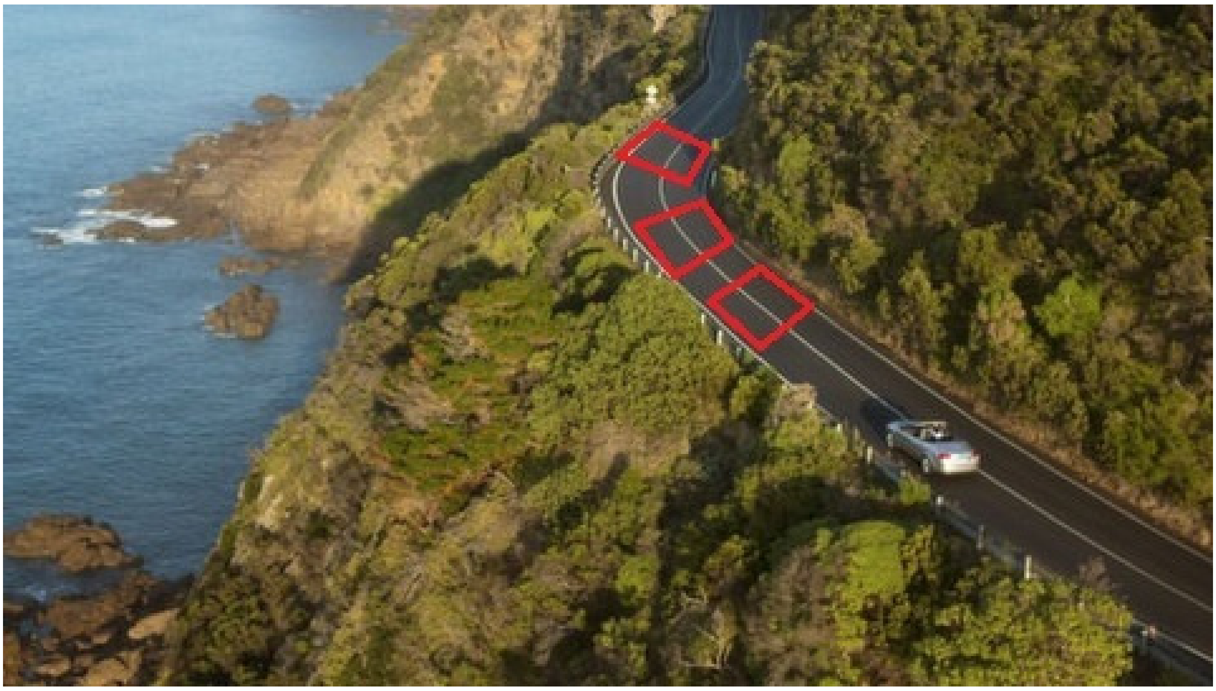


Fig. 5. Image with three etalon sections of road

Results

Testing of the proposed method was carried out on a model. First a picture with a road covering, photographed from above was selected. Then all three steps of the algorithm are carried out on the picture:

- Finding the road covering of the selected image
- Automatic tracking of the road for some direction
- Finding all kinds of different objects on the road

Pic 6 shows the result of the experiment. For a given pattern, the following steps are carried out.

After the first stage, the road sections for etalons are selected. For our case we have 2 etalons of this kind, as there are two intersecting roads in the photo. Then, the direction of tracking is given. The program's objective is to find objects on the road, continuously tracking it. For the given example, the program finds such areas very well. The picture below shows the areas that have been found by the program.

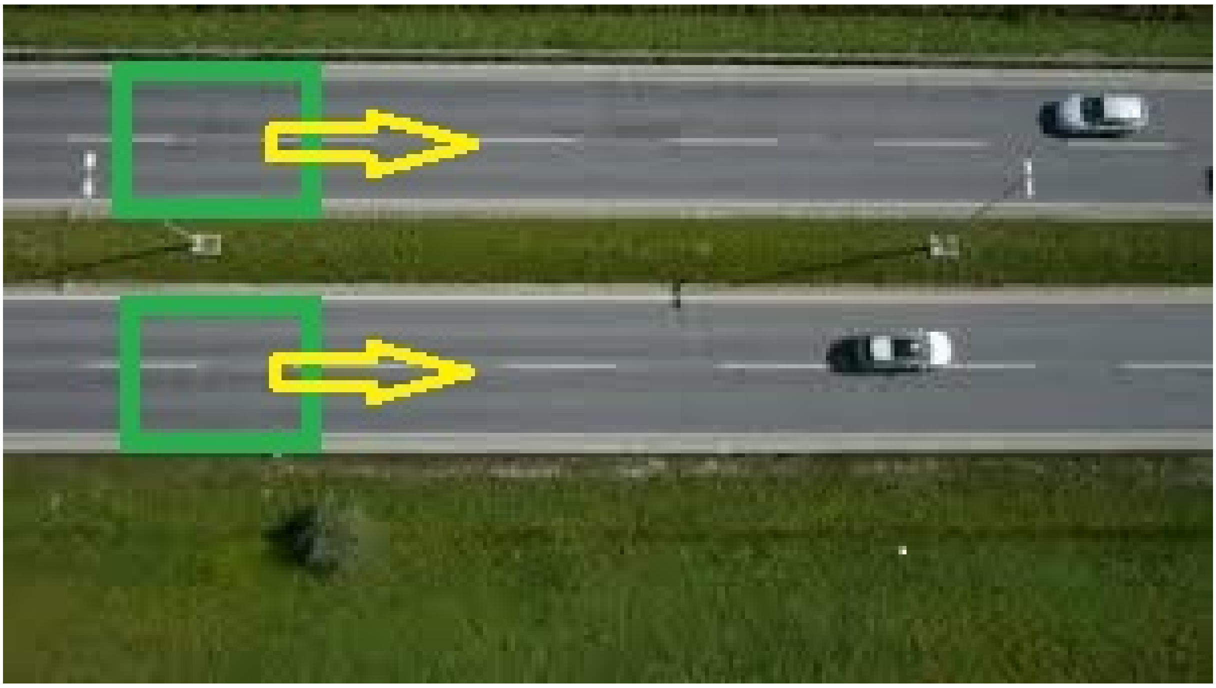


Fig. 6. By green squares the etalon pieces of the road are labeled and the direction of tracking is set.

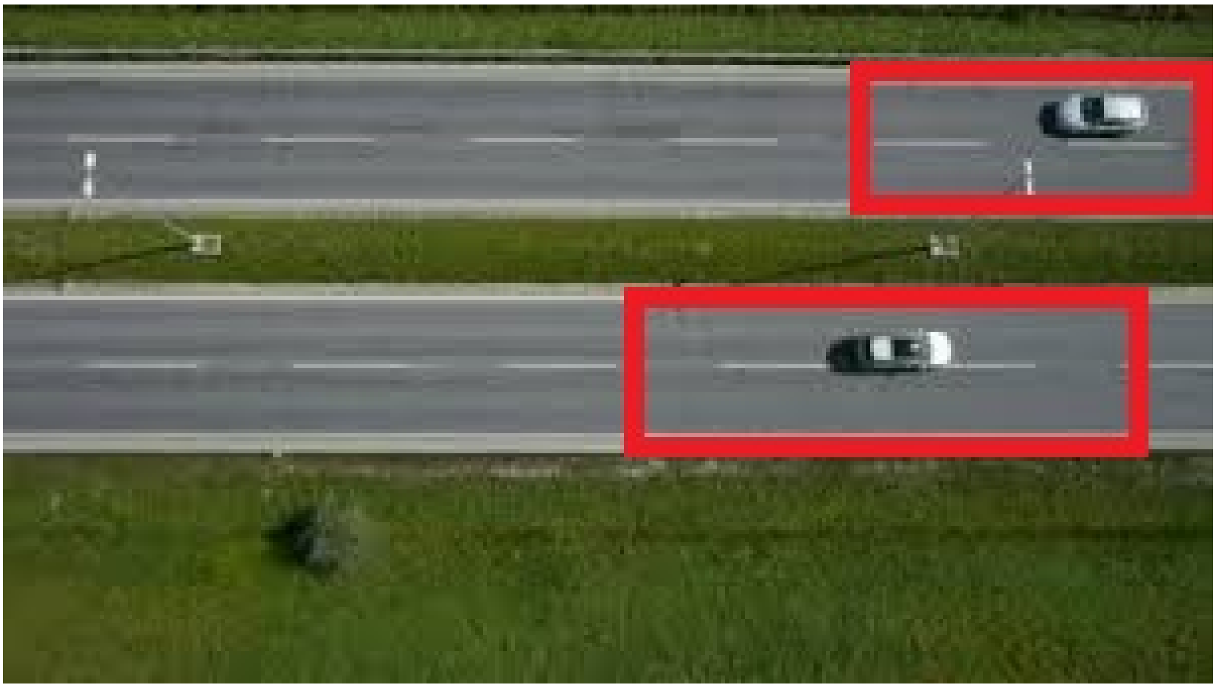


Fig. 7. The result of the algorithm.

The images, where the road is clearly allocated, the algorithm always gives quite good results.

Conclusion

A new method for automated road monitoring with the help of UAV is offered. The method is suitable for determining image areas, where the heterogeneity compared with the general picture is spotted. An algorithm for finding the road cover in pictures, as well as the algorithm for finding the direction of the road from the image is offered.

The methodology is applicable to the processing of automatic control of unmanned aerial vehicles in order to find and track roads, as well as registration of various types of objects on the road.

Bibliography

[Coifman 2004] B. Coifman, M. McCord, M. Mishalani, and K. Redmill, “Surface transportation surveillance from unmanned aerial vehicles,” in *Proc. 83rd Annu. Meet. Transp. Res. Board*, 2004, pp. 1–9.

[Frew 2004] E. Frew *et al.*, “Vision-based road-following using a small autonomous aircraft,” in *Proc. IEEE Aerosp. Conf.*, Mar. 2004, vol. 5, pp. 3006–3015.

[Zhou 2013] H. Zhou, D. Creighton, L. Wei, D. Y. Gao, and S. Nahavandi, “Video driven traffic modeling,” in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics*, Jul. 2013, pp. 506–511.

[Wang 2004] Y. He, H. Wang, and B. Zhang, “Color-based road detection in urban traffic scenes,” *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 309–318, Dec. 2004.

[Siogkas 2007] G. K. Siogkas and E. S. Dermatas, “Random-walker monocular road detection in adverse conditions using automated spatiotemporal seed selection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 527–538, Jun. 2013.

[Rathinam 2007] S. Rathinam *et al.*, “Autonomous searching and tracking of a river using an UAV,” in *Proc. IEEE Am. Control Conf.*, Jul. 2007, pp. 359–364.

[Rathinam 2008] S. Rathinam, Z. Kim, and R. Sengupta, “Vision-based monitoring of locally linear structures using an unmanned aerial vehicle 1,” *J. Infrastructure Syst.*, vol. 14, no. 1, pp. 52–63, Mar. 2008.

[Comaniciu 2000] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” in *Proc. IEEE Comput. Vis. Pattern Recog.*, 2000, pp. 142–149.

[Nummiaro 2003] K. Nummiaro, E. Koller-Meier, and L. Van Gool, “An adaptive colorbased particle filter,” *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.

[Schunck 1981] B. G. Schunck, “Determining optical flow,” *Artif. Intell.*, vol. 17, no. 1, pp. 185–203, Aug. 1981.

[Asatryan 2015] D. Asatryan, S. Hovsepyan, “Vision Based Technique for Smoke and Fire Detection in Monitored Forest Terrain”, Computer Science and Information Technologies, vol. 10, no. 1, pp. 129–132, Sep 2015.

[Asatryan 2010] D. Asatryan, K. Egiazarian, V. Kurkchiyan “Orientation estimation with applications to image analysis and registration” Information Theories and Applications, Vol. 17, Number 4, 2010 pp 303-311

[Asatryan 2009] D. Asatryan and K. Egiazarian. “Quality Assessment Measure Based on Image Structural Properties.”

International Workshop on Local and Non-Local Approximation in Image Processing, Finland, Helsinki, pp. 70-73, 2009.

Authors' Information

Samvel Hovsepyan *– Ph.D. student at Russian-Armenian (Slavonic) University, Senior software developer at OMD Armenia: Yerevan, Armenia; e-mail: samvel1207@gmail.com*

Major Fields of Scientific Research: Digital signal and image processing, Software development

INTEGRATION OF ONTOLOGY RESOURCES INTO OPEN FORMAT DOCUMENTS FOR SEMANTIC INDEXING

Viacheslav Lanin

***Abstract**: The article describes the development of a software library for ontological metadata inclusion into modern office documents formats. The model of the document used for indexing its content by ontology concepts is given. Existing projects addressed for similar problems are overviewed.*

***Keywords**: ontology; semantic indexing, document formats.*

***ACM Classification Keywords**: I.2 Artificial Intelligence: I.2.11 Distributed Artificial Intelligence; I.7 Document and Text Processing: I.7.2 Document Preparation; I.7.3 Index Generation.*

Introduction

In modern information systems (IS) there is a shift from the processing of structured data to unstructured data handling. For definiteness, we mean that unstructured data is the traditional electronic documents in different formats. This trend is noticeable in the corporate sector and among private users. Specialized software and formats for storing documents were developed and used throughout the history of information technologies for the processing of documents. Nowadays new classes of systems (social networks, corporate portals, wiki-resources, etc.) become the only important part of the information space. The key element of those classes is the concept of "content", which can be generalized to the "electronic paper”.

Across-the-board applicable technology WYSYWIG becomes a "time bomb" for electronic documents. Most of modern technologies that used for working with documents (text editors, language HTML) focus on organization of convenient work with information for the person because often ways to work with electronic information just copy the methods of work with a "paper" information. Text editor contains wide opportunities for text formatting (presentation in human readable format), but there are practically no opportunities for the transfer of the semantic content of the text, i.e. there are no tools for semantic indexing. Automatic intelligent processing of text is extremely difficult, because we usually have to deal with the "document for the people" and not to "document for the person and the system".

The modern approach to the definition of an electronic document requires metadata that describes the structure and semantics of the data presented in the document. Due to this approach, the processing of electronic documents can be organized in a qualitatively different level, since it is possible to automatically intelligent analysis of information. This concept is laid in the project Semantic Web, but the status of the "Semantic Web" for a variety of reasons is still far from implementation. However, the ideas of the Semantic Web [Berners-Lee, 2001] can be implemented within a single information system due to the smaller scale of its domain. Currently, the data required for processing the documents dispersed (stored in the document as well as in databases of IS for processing documents), and is specific to each of the tasks accomplished during the life cycle of the document in the IS. Therefore, it is necessary to use a single mechanism to provide information about the document. Another solution is the ontological resource that describes the various aspects of electronic documents that exist throughout its life cycle. This resource can be the basis for a wide range of tasks associated with the processing of electronic documents in the IS.

For the complex decision of tasks is necessary to develop a model of the electronic document that allows to include the meta and ontological resource (the basis for semantic indexing document contents). Also it’s needed to develop technology for the introduction of metadata in the document and to propose a mechanism for processing documents. This paper is devoted to discussion of possible approaches to solving the above problems.

Document Model

The electronic document is a set of structural elements called fragments in this paper. For example, they are table, header, details form, etc. Thus the document can be represented by four species:

$$d = (S\,(F, R), C, o, M).$$

Here *S (F, R)* is oriented hypergraph. Nodes of that hypergraph are compared elements of *F* (*F* is a set of document fragments, and *R* is the set of edges of corresponding relations between fragments); elements of *C* represent the information content of the document (its contents); o is document ontology, *M* is mapping of *F* on the ontology’s concepts. Let us examine the described components.

Hypergraph *S (F, R)* defines the relationship between the portions of the document. Direction of graph is needed, for example, to keep track of links "part-whole" between fragments. The nodes belonging to the edge are numbered that allows to set the order of the fragments in the text. Obviously the edge that includes all nodes corresponds to the document entirely.

There two types of fragments. The first type is basic simple fragments that are indivisible elements, such as the title or date of the document creating. The second type is compound fragments that contain other fragments.

Formally define fragment as a pair of the form:

$$f = (stat, inf),\ inf = \left[\begin{array}{l} F^*, F^* \subseteq F; \\ c, c \in C \end{array} \right. .$$

There *stat* is a static part of the fragment (it can be represented as a text, images, links, any special symbol. In addition, there information for representing the fragment may be contained); *inf* is a part of fragment that indicates the location for placement of element content *c* ($c \in C$), or contains a set of fragments *F**.

Traditionally, common graphs are used for document presentation. Usually trees are used (e.g., format XML). Tree structure of description is much easier to work with the document but at the same time, it has significant limitations. Selecting of hypergraph to represent document structure is substantiated of possibility of hypergraphs to present arbitrary connections between fragments of documents and their sets.

In the above notation, the document template can be defined as $t = (S\ (F, R), C_0)$, where C_0 is the *primary content* (for instance, standard headers that are included in the template, etc.).

In view of the specificity of solved problems in this paper, we specify the notion of ontology:

$$o = (C, R, A),$$

where *C* is a set of ontology concepts, *R* is the set of relations between concepts, *A* is a set of axioms, that are determined on ontology. Both classes and instances of these classes can be concepts. Axioms are used to set limits and rules that cannot be expressed in terms of the relationship.

For documents processing it's necessary to implement the operation of allocation an arbitrary part of the document (let us call it *operation of range getting*). The input parameter of that operation is arbitrary set of nodes, and the result is the subgraph generated by this set of nodes. *Operation of decoding* is "imposition" of the structure on the fragment (node of graph). In the majority of applications *visual layout of the document* and its presentation in a certain format are very important, so the *operation of document presentation in specific format* is necessary. This operation represents the function that sets a mapping between document fragments and some set of formats, the elements of which set the rules of fragments displaying. *The search operation* is applied to the various components of the document: the structure, content and presentation, and the result of the operation will be parts of the document matching search criteria.

Project Semantic Assistants

Semantic Assistants is an open source research project [Witte, 2008] developed by Canadian laboratory Semantic Software Lab. Semantic Assistants helps users to extract, analyze and development of content providing contextual services of NLP (Natural Language Processing). It directly integrates with desktop applications (word processors, email clients, web browsers), web information systems (e.g., wiki) and mobile applications based on Android. Semantic Assistants has an open service-oriented architecture and uses OWL ontologies Semantic Web.

Semantic Assistants architecture consists of four levels. On the first level, there are the client applications. On the second level, there are Web services and NLP Service Connector, which now wraps GATE framework for NLP and it is responsible for communication with customers, read requests, and the creating of the responses. The third level is NLP subsystem that is responsible for extracting, compiling and indexing of information as well as search. The fourth level is resource. It contains all the necessary external documents to that subsystem NLP should have access.

This work is of interest claiming a large number of supported client applications and offering good architecture. But at the moment the project is under construction and only three clients are implemented. Only one of them is word processor OpenOffice.org Writer. For introduction of semantic information in ODF documents Semantic Assistants does not use all the possibilities provided by the specification of ODF 1.2 and uses peer review mechanism adding to the document notes. Thereby it keeps the information in unstructured form and accessible for editing by the user, which is not always convenient.

Word Add-in for Ontology Recognition

Word Add-in for Ontology Recognition (Word Add-in) [Fink, 2010] is tool for the manual annotation of documents in Microsoft Word. Word Add-in is an application layer add-in for Microsoft Office and it is built on the .NET platform with using of VSTO technology. Word Add-in is an open source project that allows adapting it easily to needs of any interested user.

Using the Word Add-in begins with the selection of the base ontology. It’s an electronic catalog that contains ontology related to one subject domain. The user can select one of the ontology of the database and then starts working with Word Add-in in the background to analyze the input text. If the word matches one of the selected ontology’s concepts it will be specially marked (smart tags or custom actions). When the smart tag is activated or custom actions are selected in the menu there is a special context menu with which this concept can be viewed in browser of ontologies.

One of the main problems of Word Add-in is synonymous. It is also one of the problems is that a word may correspond to several concepts of different ontologies. In this case, the user has to select one of ontology the most satisfying sense of the text.

In spite of the above-mentioned disadvantages, Word Add-in is a completely finished product. Its main advantage is the high level of integration with one of the most popular office suite Microsoft Office, which allows using it of wide range of users and does not have specific requirements for their preparation.

An Infrastructure for Managing Semantic Documents

Infrastructure for Managing Semantic Documents (ISDM) is specialized industrial product [Lucas, 2010]. Main functions of Infrastructure for Managing Semantic Documents (ISDM) are:

- semi-automatic annotation of electronic documents based on ontologies using markup templates;
- version control of electronic documents;
- semantic search;
- notification of changes.

ISDM consists of two main modules. The first is a semantic document repository (Semantic Document Repository – SDR) for storing electronic documents. The second module is so-called ”*main module*” which in turn also has a complicated structure and can be divided into several sub-modules. They are module of semantic markup (Semantic Annotation Module – SAM), data extraction module and version control (Data Extraction and Versioning Module – DEVM), Search module (Search and Traceability Interface Module – STIM).

Semantic markup Module (SAM) allows you to add metadata to the corresponding subject ontology electronic document. Version ISDM described in provides a single electronic document format ODF 1.0. This format version is still lacking convenient and flexible metadata model and so the authors were forced to use the most appropriate means provided by the format. Instead of using the manual annotation of documents an approach is promoted based on the use of templates that allows reusing metadata.

Metadata is used to represent the so-called "instructions”. They are instance and property. To specify ontologies that are used in the annotation another hidden field with the name «Ontologies» is used in the sense of that, the URL of the ontology is indicated.

Although this project is still relevant to this day, its main part, namely the mechanism of semantic markup is significantly out of date, because it is designed in accordance with ODF 1.0, while the new ODF 1.2 specification provides substantial tools to add metadata to ODF documents.

The architecture of the OfficeMetadataLib component

In this section, we will describe the requirements for software library OfficeMetadataLib and will show the architecture designed to solve the problem.

The OfficeMetadataLib component should be implemented as a software library that provides a set of functions:

- creating new and opening existing textual electronic document of Office Open XML and OpenDocument formats;
- providing access (control) of textual content of documents;
- providing access (control) of preinstalled and user metadata documents;
- inculcation of ontologies in OWL format to the document metadata and providing access to them (management);
- automated search and binding of the document's fragments of text with ontology concepts;
- possibility of algorithm's expanding and replacement of implemented basic search algorithms and lemmatization.

The OfficeMetadataLib software library should have a modular architecture (schematically shown in Fig. 1) for providing unified access to the electronic document format Office Open XML and OpenDocument, and the possibility of expanding the basic search algorithms and lemmatization.

DocumentModel describes the generalized model of textual office document that consists of two levels:

- ContentModel is model of document content.
- MetadataModel is model of document metadata.

This model should be designed in accordance with ISO/IEC 29500 standard that is described in [ISO/IEC 29500-1; ISO/IEC 29500-2; Open, 2011] and OASIS ODF 1.2 specification.

LemmatiserModel describes a generalized model of lemmatizer.

SearchModel describes a generalized model of search engine

OOXMLPlugin and *ODFPlugin* implement a generalized model of document in accordance with specifics of Office Open XML and OpenDocument formats. Selection of the feature's implementation for each document format into a single plug-in allows to refine and modify the code for each plugin individually (e.g., in case of changing the specification of document format) without changing the overall model and without touching the source code of the main library and other plug-ins.

LemmatiserPlugin is a concrete implementation of lemmatizer.

Selection of lemmatizer implementation as a plugin will allow connecting to the library third-party lemmatizers that implement appropriate interfaces.

BaseSearchPlugin implements a basic search engine. Like a lemmatizer it can be changed by third-party developers.

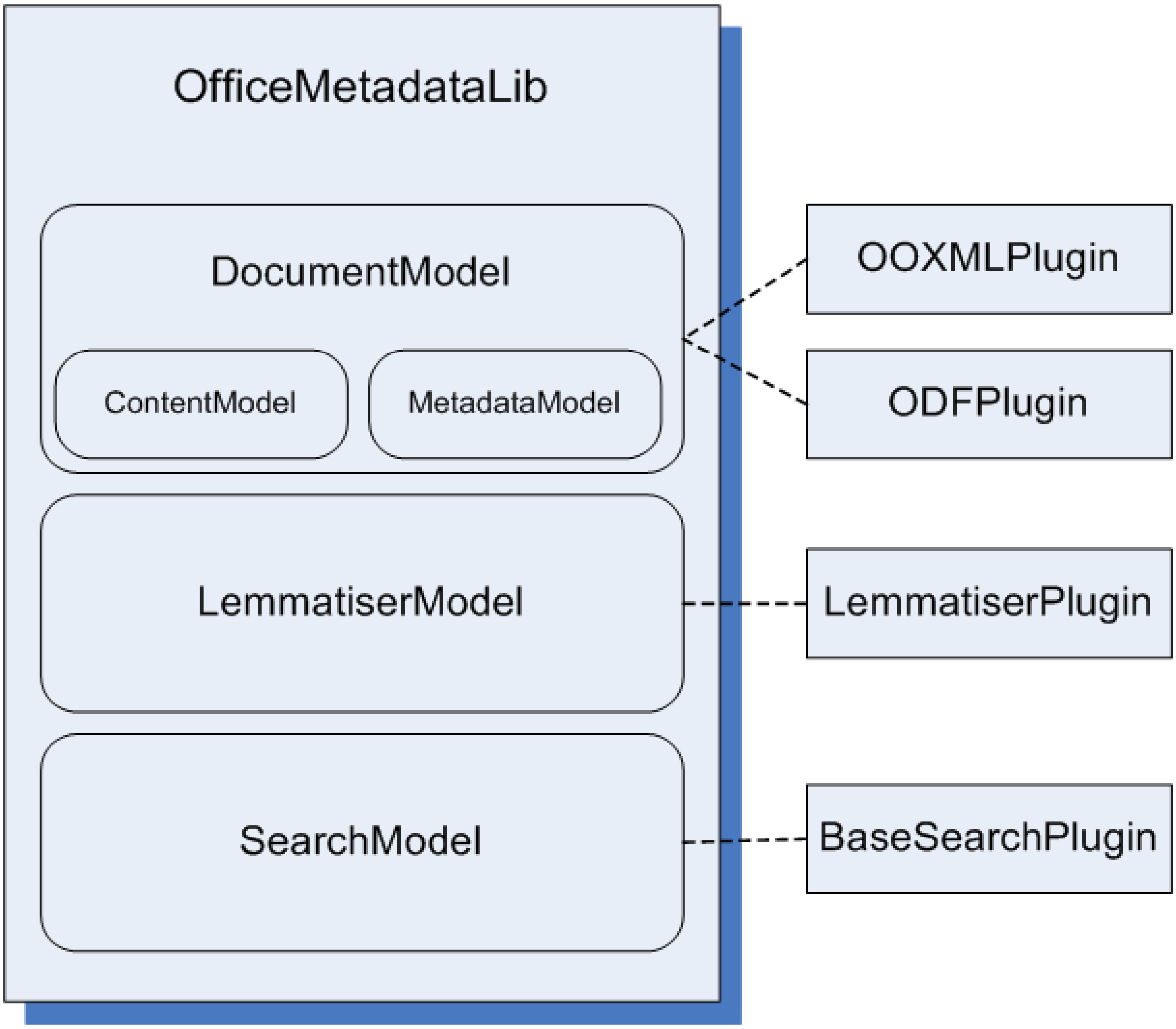


Fig. 1. The architecture of OfficeMetadataLib

Conclusion

In this work a software library providing unified access (control) to metadata of office document of Office Open XML format and OpenDocument format was developed. The main component of the library is *OfficeMetadataLib.DocumentModel*. It is a model of an electronic document and metadata based on the models of ISO / IEC 29500 (Office Open XML) and OASIS ODF 1.2. This model adequately reflects the characteristics of both formats and allows working with documents that use these formats in unified way. It is also worth noting that despite the fact that *OfficeMetadataLib.DocumentModel* was originally designed to work with documents in the Office Open XML Formats and the OpenDocument (thanks to its flexible structure) there is a theoretical possibility of the use of the library for work with other document formats.

OfficeMetadataLib.DocumentModel describes document model and its metadata. Software implementation of model's transformation functions into document of specific format is contained in a special plugins that use special API for this (for example, Open XML SDK, OpenOffice.org SDK). Using of an approach based on the plugins allows eliminating the need for self-realization of all the features of the work with the above formats and allows using existing software solutions. Also worth noting that the use of plugins provides a high degree of flexibility and extensibility. In the case of obsolescence of any library or the appearance of a new more user-friendly library, it is possible to simply replace or add a plugin without changing existing code model.

The use of a unified approach to the development of a model to work with electronic documents is led to the fact that there is no possibility to use some features of formats.

In the future versions of the library it is planned to implement an interface for executing SPARQL queries to metadata document.

Bibliography

[Berners-Lee, 2001] Berners-Lee T., Hendler J., Lassila O. The Semantic Web. In: Scientific American (May 2001). Pp. 28-37.

[Bakalov,] Bakalov F., Sateli B., Witte R., Meurs M.-J., Komg-Ries B. Natural Language Processing for Semantic Assistance in Web Portals. In: IEEE Sixth International Conference on Semantic Computing (ICSC 2012), 2012. Pp. 67–74.

[Witte, 2008] Witte R., Gitzinger T. Semantic Assistants – User-Centric Natural Language Processing Services for Desktop Clients. In: 3rd Asian Semantic Web Conference (ASWC 2008), ser. LNCS, vol. 5367. Bangkok, Thailand: Springer, 2008, p. 360–374. [Online]. Available: http://rene-witte.net/semantic-assistants-aswc08.

[Fink, 2010] Fink JL, Fernicola P, Chandran R, et al. Word add-in for ontology recognition: semantic enrichment of scientific literature. BMC Bioinformatics. 2010;11:103. doi:10.1186/1471-2105-11-103.

[Lucas, 2010] Lucas de Oliveira Arantes, Ricardo de Almeida Falbo. An Infrastructure for Managing Documents. In: 14th IEEE International Enterprise Distributed Object Computing Conference Workshops. 2010.

[ISO/IEC 29500-1] ISO/IEC 29500-1 Third edition, 2012-09-01. Information technology – Document description and processing languages – Office Open XML File Formats. Part 1: Fundamentals and Markup Language Reference.

[ISO/IEC 29500-2] ISO/IEC 29500-2 Third edition, 2012-09-01. Information technology – Document description and processing languages – Office Open XML File Formats. Part 2: Open Packaging Conventions.

[Open, 2011] Open Document Format for Office Applications (OpenDocument) Version 1.2 Part 1: OpenDocument Schema 29 September 2011.

[Lanin, 2014] *Lanin V., Sokolov G.* Using multidimensional ontology of electronic document for solving semantic indexing problem. In: Proceedings of the 8th Spring/Summer Young Researchers' Colloquium on Software Engineering (SYRCoSE 2014). M. : ISP RAS, 2014. Pp. 166–169.

Authors' Information

Viacheslav Lanin *– National Research University Higher School of Economics, Department of Business Informatics; senior teacher; Perm, 614070, Studencheskaya st., 38; e-mail: lanin@perm.ru, vlanin@hse.ru.*

Major Fields of Scientific Research: Intelligent agents, Ontologies, Document processing.

TABLE OF CONTENT IJ ITK VOL.9 NO.:1

TABLE OF CONTENT IJ ITK VOL.9 NO.:2

TABLE OF CONTENT IJ ITK VOL.9 NO.:3

TABLE OF CONTENT IJ ITK VOL.9 NO.:4