

УДК 004.2

Скулиш М. А., канд. техн. наук (Тел.: +380 05 607 42 29. E-mail: mb\_s@ukr.net)  
(Національний технічний університет України «КПІ», Інститут телекомунікаційних систем, м. Київ)

## МЕТОД СКЛАДАННЯ РОЗКЛАДУ ЗАЛУЧЕННЯ РЕСУРСІВ ДЛЯ ВИСОКОНАВАНТАЖЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

**Скулиш М. А. Метод складання розкладу залучення ресурсів для високонавантажених інформаційних систем.** Запропоновано метод регулювання кількості обслуговуючих вузлів, який дозволяє на основі довгострокової статистики побудувати розклад залучення технічних ресурсів для забезпечення процесу обслуговування високонавантажених інформаційних систем. Даний метод може бути застосований при плануванні роботи хостінг-провайдерів, при плануванні кількості ресурсів, які мають бути додатково орендовані для забезпечення систем реального часу, наприклад системи онлайн тарифікації.

**Ключові слова:** телекомунікаційна система, розподілена система, тарифікація послуг, динаміка навантаження, високонавантажені системи, якість послуг

**Скулиш М. А. Метод составления расписания привлечения ресурсов для высоконагруженных информационных систем.** Предложен метод регулирования количества обслуживающих узлов, который позволяет на основании долгосрочной статистики построить расписание привлечения технических ресурсов для обеспечения процесса обслуживания высоконагруженных информационных систем. Данный метод может быть применен при планировании работы хостинг-провайдеров, или при планировании количества ресурсов, которое должно быть дополнительно арендовано для обеспечения высоконагруженных систем реального времени, например, систем онлайн тарификации.

**Ключевые слова:** телекоммуникационная система, распределенная система, тарификация услуг, динамика нагрузки, высоконагруженная система, качество услуг

**Skulysh M. A. Method of drawing up the schedule of attraction of resources for the high-loaded information systems.** The method of regulation of quantity of the serving nodes which allows to construct on the basis of long-term statistics the schedule of attraction of technical resources for ensuring process of service of the high-loaded information systems is offered. This method can be applied when scheduling a hosting-provider, or when planning quantity of resources which has to be in addition leased for providing the high-loaded systems of real time, for example, of online charging systems.

**Key words:** telecommunication system, distributed system, tariffing of services, loudspeaker of loading, high-loaded system, quality of services

**Вступ.** На сьогоднішній день спостерігається перехід до використання глобальних інформаційних ресурсів. Це передбачає необхідність організації процесу одночасного обслуговування великої кількості користувачів. Список сервісів, які обслуговують мільйони користувачів, все збільшується – починаючи від соціальних мереж та пошукових машин і закінчуючи системами онлайн тарифікації крупних операторів зв'язку. Основними вимогами до технічних ресурсів, що обслуговують подібні системи, є забезпечення доступності сервісу та безперебійної роботи.

Організація процесу надання послуг великій кількості користувачів, яка оцінюється в мільйони, має спільні риси. Так, все більше компаній користуються послугами хмарних сервісів, які забезпечені технічними ресурсами та можуть надавати інфраструктури як сервіс.

На відміну від моделі зберігання даних на власних виділених серверах, у випадку використання віртуальних серверів, їх структура і кількість в загальному випадку не видна користувачеві. Вся інформація зберігається і обробляється у хмарі, яка являє собою, з точки зору клієнта, один великий віртуальний сервер. На сьогоднішній день велика кількість компаній надає таку послугу як оренда сервера у хмарі.

Основні переваги віртуальних серверів:

- *Гнучкість ресурсів.* Оренда сервера у хмарі дозволяє забезпечити його високу масштабованість. Такий сервер легко налаштовується під збільшення навантаження, наприклад, можна легко додати оперативної пам'яті або дискового простору. Так само легко

можна зменшити дані параметри віртуальної системи. Сервер стає “резиновим” у відношенні своїх ресурсів.

- *Швидкодія.* Віртуальний сервер працює значно швидше ніж звичайний, а також знижуються часові затрати на впровадження і оперативний перерозподіл ресурсів. Забезпечується висока швидкість розгортання системи.

- *Безпека.* На віртуальному сервері клієнт має можливість власноручно контролювати користувачів, процеси, а також використовувати власну політику безпеки. Підвищення рівня безпеки Cloud Server відбувається також за рахунок зведення до мінімуму “людського фактору”.

- *Мобільність.* Доступ до серверу можна отримати з будь-якої точки земної кулі.

Для забезпечення роботи сервісів використовується модель функціонування, яка зображена на Рис. 1.

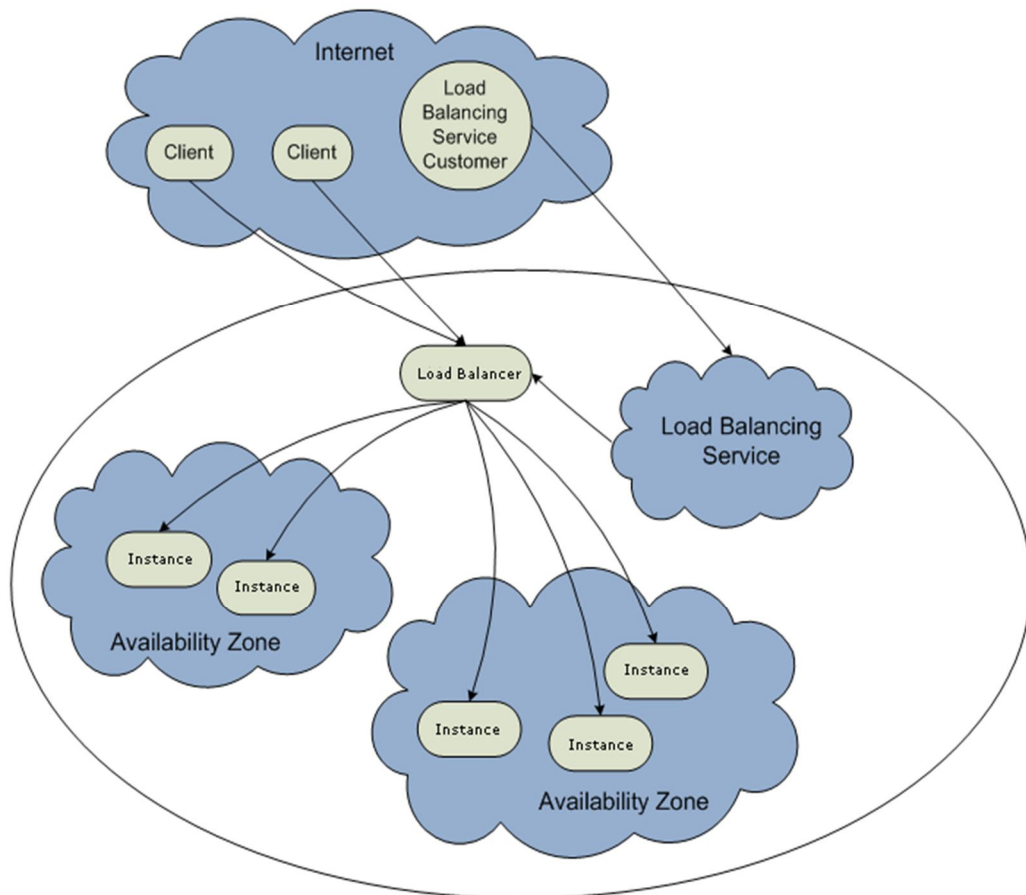


Рис. 1. Організація процесу розподілу задач високонавантажених системи в Cloud

Розглянемо процес обслуговування великої кількості абонентів на прикладі системи онлайн тарифікації оператора зв'язку. Використання віртуальних серверів для обслуговування і тарифікації послуг може значно спростити процес розподілу ресурсів на сервері тарифікації, забезпечить його масштабованість, підвищить безпеку і відмовостійкість.

Віртуальний сервер може бути використаний трьома способами. *По-перше*, він може бути додатком до основного фізичного сервера і використовуватись коли виникає потреба у збільшенні кількості ресурсів при збільшенні навантаження. В такій конфігурації вся логіка серверу реалізована у хмарі. *В другому варіанті* все обладнання для обслуговування і тарифікації викликів може розміщуватись у хмарі. В такому випадку фізичний сервер взагалі не потрібен і процес обслуговування запитів буде повністю проходити на віртуальному сервері. *В третьому випадку*, який розробляється, у хмарі можуть знаходитися окремі

ресурсозатратні підсистеми або частини підсистем, організація взаємодії яких є окремою мережевою задачею.

Наведені вище способи значно розширюють можливості технічного забезпечення системи онлайн тарифікації. При цьому для ефективного планування необхідної кількості технічних ресурсів необхідно розробити методи, що дозволяють планувати кількість необхідних ресурсів. В (2) біло запропоновано динамічний метод оцінки достатності підключених ресурсів у поточний момент часу.

В даній статті пропонується довгостроковий метод, який забезпечує безперебійну та економічну роботу системи тарифікації. Економічність полягає в тому, що відповідно до статистичних даних про вхідне навантаження, відбувається включення потрібної кількості серверів, які забезпечують роботу системи тарифікації або його підсистем. Фактично такий розклад визначає, між скількома обслуговуючими одиницями (Instance, див. Рис.1) буде здійснюватися балансування навантаження.

**Метод складання розкладу включення серверів.** Для забезпечення безперебійної роботи серверу тарифікації необхідно розробити комплекс засобів, які враховують не тільки поточну ситуацію, але й статистику набрану за тривалий період часу. Це дозволить планувати ресурси, розробляти методи балансування навантаження, тощо.

Для рішення даної задачі необхідно провести аналіз статистики протягом тривалого періоду. Наприклад, аналіз статистики за понеділок протягом 15-20 тижнів. Результатом таких статистичних даних стане матриця де, для кожного малого інтервалу часу, наприклад, для часу 8:00:00 – 8:00:01, такий інтервал в Табл.1 позначатимемо значенням закінчення інтервалу: 8:00:01. В матриці будуть зібрані дані про кількість заявок (навантаженні), яке потребувало обслуговування на всіх доступних серверах для яких складається даний розклад.

Розглядатимемо групу серверів як єдину систему з одним входом, задача розподілу вхідного навантаження між доступними серверами не розглядається. Необхідно знайти скільки серверів з відомими технічними характеристиками повинно бути залучено до обробки вхідного навантаження для забезпечення мінімуму втрат через перевантаження серверу.

**Першим етапом** методу є розбиття загального періоду часу на відрізки, які характеризуються однаковою динамікою зміни навантаження. Оцінка динаміки зміни навантаження буде проводитися на основі аналізу прямої, яка апроксимує статистичні дані.

**Алгоритм розбиття на відрізки.** Вхідні дані: Статистика навантаження  $x(n,t)$ , яка описана в Табл. 1, де  $t$  – час вимірювання;  $n=1,..N$  – номер тижня для якого збиралася статистика, проводилося вимірювання навантаження;  $x(n,t)$  – кількість заявок на тарифікацію або навантаження на сервер, що було заміряно на  $n$ -му тижні в момент часу  $t$ .

**Табл. 1**

$n \backslash t$	00:00:00	00:00:01	...	23:59:58	23:59:59
1	$x(n,t)$				
2					
$N$					

**Крок 1.** Для кожного значення  $t$  знаходимо середнє значення за формулою (1) дані зводяться в таблицю, як показано в Табл. 2

$$\bar{x}(t) = \sum_n x(n,t) / N \quad (1)$$

**Табл. 2**

$T$	00:00:00	00:00:01	00:00:02	00:00:03	...
$\bar{x}$	$\bar{x}(t)$	$\bar{x}(t)$	$\bar{x}(t)$		

Крок 2. Задати час завантаження серверу  $T$ , задати мале число  $\varepsilon_1$

Крок 3. Необхідно розділити множину допустимих значень  $t$  на підмножини  $t_i$ , таким чином, щоб  $t_{i+1}-t_i=T$ . Окремо розглядається окремі частини матриці з Табл. 2, з них формується множина матриць  $I = \{I_1, \dots, I_n\}$ , в процесі виконання алгоритму ці матриці будуть об'єднуватися та перенумеровуватися, на початку роботи алгоритму кількість матриць  $n=24 \cdot 3600/T$ ,  $(i+1)$ -я матриця буде мати вигляд такий як показано в Табл. 3:

Табл. 3

$t$	$t_i + 00:00:01$	$t_i + 00:00:02$	...	$t_{i+1} - 00:00:01$	$t_{i+1}$
	$(t)$	$(t)$	$(t)$		

Крок 4. Методом найменших квадратів знаходимо для кожної  $i$ -ї матриці, для пар значень  $(t, n)$  за методом найменших квадратів знаходимо оцінку коефіцієнту апроксимуючої прямої  $n = a \cdot t + b$ .

Крок 5. Для всіх  $i=1, \dots, n$  проводиться аналіз  $n_i$ . Якщо  $|n_i - n_{i-1}| < \varepsilon_1$ , тоді множини  $i$  і  $(i+1)$  об'єднуються. Отримані множини отримують нову нумерацію, кількість нових множин позначається, як  $n_{new}$ . Перехід на Крок 4. Інакше, якщо для всіх  $i=1, \dots, n_{new}$ :  $|n_i - n_{i-1}| > \varepsilon_1$ , тоді позначаємо  $n_{new}$  як  $n_{last}$ , розбиття знайдене.

Результатом роботи даного алгоритму є розбиття на ділянки з однаковою динамікою зростання навантаження. Для отриманих значень  $t_i$  ( $i=1, \dots, n_{last}$ ,  $n_{last}$  – кількість інтервалів, якщо більше немає можливості поєднувати підмножини) відбувається порівняння середньостатистичного навантаження  $(t_i)$  з допустимим навантаженням на доступний сервер.

Другим етапом побудови розкладу включення серверів є визначення кількості серверів, яка необхідна для обслуговування заявок. Результатом роботи даного етапу є інформація про час, коли необхідно змінити кількість обслуговуючих пристроїв та відповідна кількість серверів, яка має бути включена в той чи інший момент часу. Ілюстрація динаміки зміни кількості заявок протягом доби наведена на Рис. 2

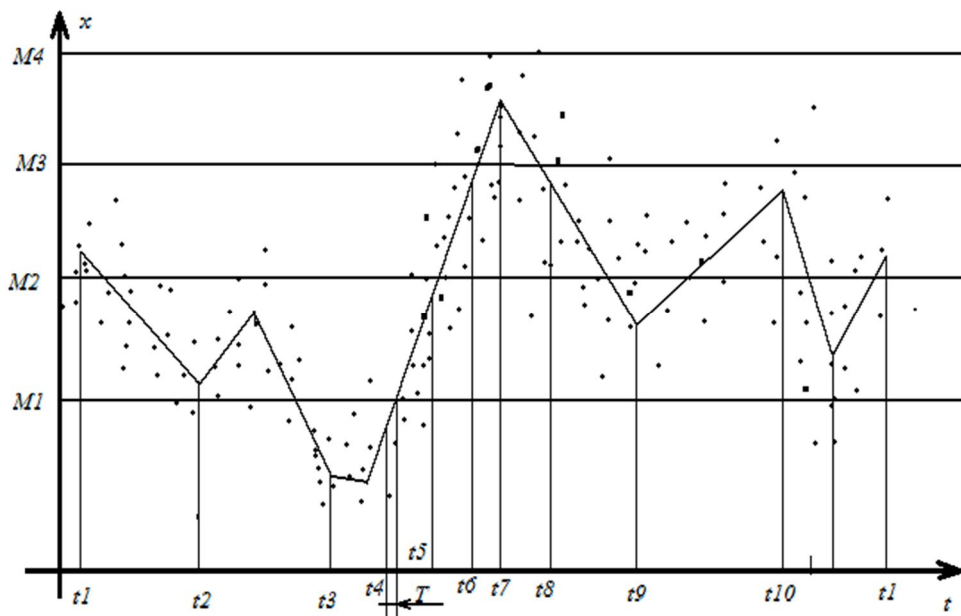


Рис. 2. Динаміка зміни кількості заявок протягом доби

Алгоритм пошуку моментів переключення: Вхідні дані:

1.  $(t_i)$  – середньостатичне навантаження в момент часу  $t_i$  (масив  $\{ (t_i) \}$  кількість елементів:  $i=1, \dots, n_{last}$ , де  $n_{last}$  – кількість моментів, в які змінюється динаміка вхідного навантаження (переломні моменти).
2.  $D(t_i)$  – середньоквадратичне відхилення.

3. Множина  $M=\{M_1, M_2...M_K\}$  – значення допустимого навантаження, яку може обслужити 1, 2, ...,  $K$  серверів відповідно.

4. Масив значень  $\widehat{\alpha}_i$ , які відповідають куту нахилу прямої апроксимації статистичних даних до моменту можливого переключення  $t_i$ .

5.  $T$  – час завантаження серверу.

*Крок 1.* Для кожного  $i=1,.. n_{last}$  знайти  $x'(t_i) = \bar{x}(t_i) + D(t_i)$ .

*Крок 2.* Створити матрицю  $R=\{r_{qp}\}$ ,  $q=1,..3$ ,  $p=1,..n_{last}$ . Змінній  $p$  присвоїти значення "1" ( $p=1$ ).

*Крок 3.* Для кожного  $i=1,.. n_{last}$  знаходимо пару  $(M_k, M_{k+1})$  так, щоб виконувалось  $x'(t_i) \in [M_k, M_{k+1}]$ .

*Крок 4.* Для всіх  $i=1,.. n_{last}$  перевірити умову: якщо  $x'(t_i) \in [M_k, M_{k+1}]$  та  $x'(t_{i+1}) \in [M_k, M_{k+1}]$ , тоді  $i=i+1$ ; повторити *Крок 3*, інакше в момент часу  $t_i$  потрібно буде переключення (перехід на *Крок 4*).

*Крок 5.* Якщо  $x'(t_i) \in [M_k, M_{k+1}]$  і  $x'(t_{i+1}) > M_{k+1}$ , і  $x'(t_{i+1}) < M_{k+s}$ , тоді в момент часу  $t_i$  необхідно включення серверу. Якщо  $s > 2$ , перехід на *Крок 6*; інакше додати в матрицю  $R$  елементи  $r_{1p} = t_i$ ,  $r_{2p} = k+2$ ,  $r_{3p} = 1$ ,  $p=p+1$ . Якщо  $i = n_{last}$ , перейти на *Крок 8*, інакше  $i = i+1$  перейти на *Крок 5*.

Якщо  $x'(t_i) \in [M_k, M_{k+1}]$  і  $x'(t_{i+1}) < M_k$ , і  $x'(t_{i+1}) > M_{k-s}$ , тоді в момент часу  $t_i$  необхідно включення додаткового серверу. Якщо  $s > 0$ , тоді перехід на *Крок 7*, інакше додати в матрицю  $R$  елементи  $r_{1p} = t_i$ ,  $r_{2p} = k+1$ ,  $r_{3p} = 0$ ,  $p=p+1$ . Якщо  $i = n_{last}$  перейти на *Крок 8*, інакше  $i = i+1$  перейти на *Крок 5*.

*Крок 6.* Для кожного  $M_{k+1}... M_{k+s-1}$  момент включення відповідного серверу розраховується за формулою (2).

$$t_g = (M_{k+g} - x'(t_i)) / \widehat{\alpha}_{i-1} \quad (2)$$

Доки  $g=1$ ,  $g < s$ , додати в матрицю  $R$  елементи  $r_{1p} = t_i + t_g - T$ ,  $r_{2p} = k+g$ ,  $r_{3p} = 1$ ,  $p=p+1$ ,  $g=g+1$ . Якщо  $i = n_{last}$  перейти на *Крок 8*, інакше  $i = i+1$  перейти на *Крок 5*.

*Крок 7.* Для кожного  $M_{k-1}... M_{k-s'+1}$  знайти моменти відключення відповідних серверів за формулою (3). Поки  $g=1$ ,  $g < s'+1$

$$t_g = (x'(t_i) - M_{k-g}) / \widehat{\alpha}_{i-1} \quad (3)$$

Поки  $g=1$ ,  $g < s'+1$ , додати в матрицю  $R$  елементи  $r_{1p} = t_i + t_g - T$ ,  $r_{2p} = k-g$ ,  $r_{3p} = 0$ ,  $p=p+1$ ,  $g=g+1$ . Якщо  $i = n_{last}$  перейти на *Крок 8*, інакше  $i = i+1$  перейти на *Крок 5*.

*Крок 8.* Завершення роботи алгоритму.

Результатом роботи даного алгоритму є матриця, *перший* рядок якої відображає моменти часу в які необхідно проводити операцію включення або виключення серверу, *другий* рядок означає кількість активних серверів які повинні обслуговувати вхідне навантаження у наступний період часу, *третій* рядок означає процес який має бути ініційований у відповідний момент часу: 1 – включення одного з серверів, 0 – виключення одного з серверів.

Слід зауважити, що робота даного алгоритму є лише складовою частиною процесу технічного забезпечення системи онлайн тарифікації. Оскільки крім статичного розкладу включення серверів потребує розгляду питання балансування навантаження між серверами.

**Оцінка ефективності.** Як наводилося на початку статті, спектр проблем, які можуть вирішуватися за допомогою запропонованого методу та алгоритмів досить широкий. Основною ознакою систем, для яких може бути застосований запропонований метод, є виконання великої кількості процедур, ініціатором яких є люди або інші програми. Виконання процедур здійснюється за допомогою програмного забезпечення серверу, при цьому використовуються технічні ресурси системи.

На сьогоднішній день проблема перевантаження вирішується за рахунок не допуску в систему надлишкової кількості запитів на тарифікацію, тобто якщо сервер є перевантаженим, то поступає сигнал на керуючий пристрій і тимчасово заявки на

тарифікацію не приймаються. При цьому одночасно задіяні (знаходяться у режимі очікування) всі ресурси серверу, завантаження обладнання на 20-30% є нормальним в процесі обслуговування.

Був проведений експеримент роботи імітаційної моделі в двох режимах:

1. На обслуговування виділявся обмежений технічний ресурс, у разі перевантаження заявки відкидалися (Режим без застосування запропонованих методів).

2. Обслуговування заявок може проводитися від одного до трьох аналогічних серверів з обмеженими технічними ресурсами, на основі статистичних вибірки від оператора зв'язку був сформований розклад включення серверів відповідно до алгоритму пошуку моментів переключення, далі було згенеровано вхідний потік максимально наближений до реального. Включення серверів відбувалося за розкладом, а за методом включення додаткового технічного обладнання кожен три хвилини відбувалась перевірка достатності ресурсів для обслуговування вхідного потоку.

За результатами роботи імітаційної моделі було проведено дослідження використання електричної енергії, яке показало скорочення споживання енергії на 60% (Рис. 3).

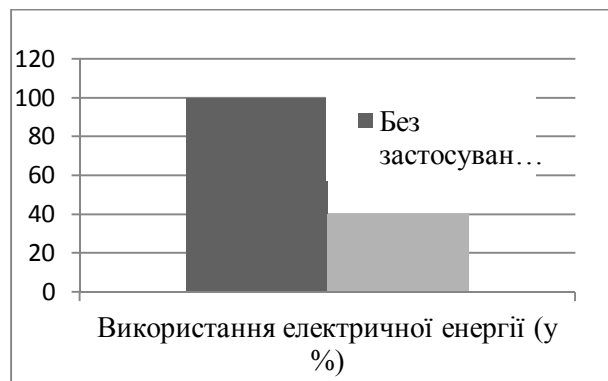


Рис. 3. Порівняння використання електричної енергії

**Висновки.** В статті розглянуті проблеми організації роботи високонавантажених систем. На прикладі роботи системи онлайн тарифікації наведено принцип організації процесу обслуговування. Запропоновано метод складання розкладу включення серверів забезпечує довгостроковий розклад включення серверів, побудований на даних довгострокової статистики, дозволить спланувати роботи технічних засобів на тривалий час. Проведене моделювання показало, що скорочується споживання електричної енергії на 60%

### Література

1. Chu R. A clustering model for memory resource sharing in large distributed system / R. Chu, N. Xiao, X. Lu // IEEE. – 2007. – P. 1-8.
2. Скулиш М. А. Організація роботи групи серверів для забезпечення потреб розподіленої системи тарифікації послуг // Наукові записки Українського науково-дослідного інституту зв'язку. – 2014. – №5(33). – С. 58-64.
3. Cohen B., New opportunities for Cloud Application Development / B. Cohen // IEEE. – 2013. – P. 97-100.
4. Sawas I. K. On Resource Clustering Techniques for Grid Resource Discovery / I. K. Sawas, IEEE. – 2007. – P. 302-307.
5. Ouellette S. A potential evolution of the policy and charging control/QoS architecture for the 3GPP IETF-based evolved packet core / S. Ouellette, L. Marchand, Pierre Samuel // Communications Magazine, IEEE. – 2011. – P. 231-239.