

класу  $k$  одним сервером кластером  $s$  є незалежною випадковою величиною  $T_c^k$  із математичним очікуванням

$$m T_c^k = M[T_c^k].$$

Всі сервери, що належать кластеру  $s$ , працюють разом і обробляють вхідні запити як єдиний сервер. Тоді час обробки запиту класу  $k$  кластером  $s$  буде дорівнювати  $T_c^k / S_c$ . Нехай  $q_c$  є номінальним завантаженням кластера  $s$ :

$$q_c = \sum_{k=1}^K \lambda_c^k m_c^k$$

Тоді при кількості серверів  $S_c$  завантаження одного сервера кластера з буде  $P_c = q_c / S_c$ . Для того щоб система була стаціонарною, мінімальні значення вектора  $\bar{S}$  повинні задовольняти умові  $S_c > q_c$ .

Нехай  $N_c^k$  означає число запитів класу  $k$ , оброблених кластером  $s$  у стаціонарному режимі.

Покладемо, що  $N_c = \sum_{k=1}^K N_c^k$  - кількість класів запитів, оброблених кластером  $s$  у стаціонарному режимі. Аналогічно  $N^k$  - загальне число, що встановилося, запитів класу  $k$  в системі.

Запит належить класу  $k$  з імовірністю  $P_c^k / P_c$ , де

$$P_c^k = \frac{\lambda_c^k m_c^k}{S_c} \text{ й } P_c = \sum_{k=1}^K P_c^k. \text{ Тоді:}$$

$$M[N_c] = \frac{P_c}{1 - P_c};$$

$$M[N_c^k] = \frac{P_c^k}{1 - P_c}.$$

Застосовуючи теорему Литтла, одержимо середній час перебування заявки класу  $k$  в кластері  $s$ :

$$M[T_c^k] = \frac{M[N_c^k]}{\lambda_c^k} = \frac{m_c^k / S_c}{1 - P_c} = \frac{m_c}{S_c - q_c}.$$

Час  $T^k$  відповіді на запит класу  $k$  — це час, що пройшов з моменту вступу запиту в систему до моменту його виходу із системи, що є сумарним часом проходження запиту через усі ланки застосування:

$$M[T^k] = \sum_{c=1}^C \lambda_c^k \frac{m_c^k}{S_c - q_c},$$

де  $S_c$  - число кластерів ЦОД;

$S_c$  - число серверів у кластері  $s$ ;

$q_c$  - номінальне завантаження кластера  $s$ ;

$m_c^k$  - середній час відповіді на запит класу  $k$  сервером кластера  $s$ ;

$\lambda_c^k$  - середнє число відвідувань кластера  $s$  запитом класу  $k$  за час його знаходження в системі.

## Висновки

Побудовані моделі ЦОД як системи масового обслуговування без поділу запитів на класи і з поділом запитів на класи можуть використовуватись для подальшого аналізу і постановки завдань оптимізації розподілом ресурсів центру обробки даних телекомунікаційної мережі.

## Література

1. Павлов О.А. Інформаційні технології та алгоритми-заячі в управлінні / О.А. Павлов, С.Ф. Теленик – К., 2002.
2. Клейнрок Л. Теория массового обслуживания / Л. Клейнрок / Пер. с англ.; Пер. И.И.Грушко; ред. В.И.Нейман. – М.: Машиностроение, 1979. – 432 с.
3. Клейнрок Л. Вычислительные системы с очередями / Л. Клейнрок / Пер. с англ.; Под ред. Б.С.Цыбакова. – М.: Мир, 1979. – 600 с.

## Резюме

Розглянуті питання функціонування центрів обробки даних. Описані особливості їх архітектури, побудовані моделі цих центрів як системи масового обслуговування без та з поділом запитів на класи

Рассмотрены вопросы функционирования центров обработки данных. Описаны особенности их архитектуры, построены модели этих центров как системы массового обслуживания без и с разделением запросов на классы

The problems of data processing centers functioning are considered. The peculiarities of their architecture are described and models of these centers as systems of queuing are constructed without and with partition of inquiries into classes

**Ключові слова:** центр обробки даних, інфраструктура системи, метод масового обслуговування

Поступила 20.09.2011 г.

УДК 621.39

ЯРЕМКО І. М., аспірант,  
ТУРУПАЛОВ В. В., к.т.н., доцент (ДонНТУ)

## Моделі масового обслуговування в ЦОД

*Представив д.т.н., професор Приходько С. І.*

### Вступ

Концепція центра обробки даних (ЦОД) втілена багатьма корпораціями переважно для забезпечення доступу великої кількості користувачів до певних ресурсів. Виявилось, що ефективна організація ЦОД пов'язана з необхідністю розв'язання низки проблем [1]. Проблема створення і забезпечення ефективного функціонування ЦОД постає перед корпораціями з розвиненою розподіленою ІТ-інфраструктурою, наперед хостинговими компаніями. Вкладаючи копії компанії полівають на прибуток. У будь-якому випадку вони очікують зменшення витрат на експлуатацію ЦОД, зниження вартості обслуговування користувачів, що дозволить, зрештою, закласти основу для ефектivelyної діяльності як самої компанії, так і клієнтів.

Клієнти своє бачення роботи ІТ-інфраструктури поєднують із хостинговою компанією на рівні вимог, до яких належать: вартість послуг, доступність, керуваність, цілісність даних, безпека, надійність, масштабованість і т.ін.

Досягнення вимог користувачів найменшими коштами і становить суттєві проблеми створення забезпечення функціонування ЦОД. Завдяки що компетенції проблему розбивають на ряд менших проблем, але від того не набагато простіших. Однією з них є проблема управління ресурсами і плавтаженням ЦОД.

### Постановка задачі

Сучасні технології побудови програмних систем є складними програмними комплексами, що мають багатуланкову архітектуру, розташованими на сукупності серверів, об'єднаних у кластери. Клієнтами програмних застосувань можуть бути браузерні користувачі, пристрої доступу, наприклад, мобільні телефони, і різні web-сервіси, такі як поштові роботи.

Запити користувачів надходять на розподільник навантаження, який направляє запит на підходящий web-сервер у кластері серверів, який вибирається від-

повідно до алгоритму роботи розподільника навантаження.

Якщо запитується статична сторінка, що зберігається на web-сервері, то web-сервер відразу посилає користувачеві HTTP-відповідь. Якщо запитується динамічна сторінка, то дані запит обробляються сервером засосувань. Сервери засосувань відповідальні за логіку засосування, вони аналізують користувачські запити, визначають дежій об'єкти, а також керують дані, що зберігаються в базі даних. Якщо потрібне одержання або відновлення інформації з бази даних, то сервер засосувань звертається до серверів БД. Відповідь сервер БД посилає на сервер засосувань, а сервер засосувань посилає відповідь на web-сервер, який, у свою чергу, генерує HTML-відповідь і видає його користувачеві.

Таким чином, постає питання побудови моделі ЦОД, яка б відображала особливості обслуговування і доступу до ресурсів.

### Модель масового обслуговування в центрах обробки даних без поділу запитів на класи

Розглянемо ЦОД, що має багатуланкову архітектуру, де під кожному ланку відведений кластер серверів, позначений  $s, s = 1, \dots, S$ . Розглянемо ЦОД без врахування класів запитів як відкритої мережу масового обслуговування (SeMO), що має  $S$  послідовних ланок (рис. 1).

Обслуговуючим приладом в моделі вибрано кластер серверів, який, у свою чергу, може бути представлений як система M/G/1 з дисципліною обслуговування «поділ: процесора» [2]. У якості зазвок будуть виступати запити користувачів до ЦОД, які відповідають Пуассонівському розподілу. Запити користувачів надходять на обслуговування на першу ланку web-серверів з інтенсивністю  $\lambda$ . Після обробки запиту web-сервером з імовірністю  $P_{12}$  заявка переходить у чергу другої ланки до кластера серверів засосувань

© І. М. Яремко, В. В. Турупапов, 2011

або з імовірністю  $(1 - p_{12})$  заявка виходить із системи, тобто користувачеві надсилається відповідь.

Заявка може пройти через усе ланки системи, причому деякі може відвідати кілька раз.

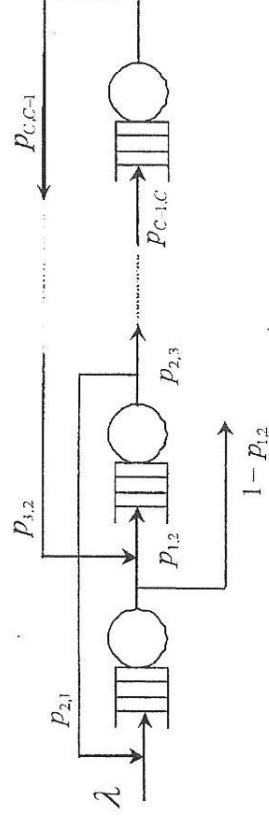


Рисунок 1 – Функціональна модель ЦОД без врахування класів запитів

Знаючи ймовірності переходів заявки між вузлами, можна визначити інтенсивності надходжень запитів на кожну ланку  $\lambda_c$ ,  $c = 1, \dots, C$ :

$$\begin{cases} \lambda_1 = \lambda + \lambda_2 P_{21} \\ \lambda_2 = \lambda_1 P_{12} + \lambda_3 P_{32} \\ \lambda_3 = \lambda_2 P_{23} + \lambda_4 P_{43} \\ \vdots \\ \lambda_C = \lambda_{C-1} P_{C-1C} \end{cases}$$

Визначення середнього часу відповіді. Час обробки запити одним сервером кластера  $c$  є випадковою величиною  $T_c$  з математичним очікуванням

$$m_{T_c} = M[T_c].$$

Розглянемо вектор розподілу ресурсів  $\bar{S} = (S_1, S_2, \dots, S_C)$  такий, що кожному кластеру  $c$  належить  $S_c$  ресурсів (серверів). Припустимо, що кожний сервер у кластері має однакову продуктивність, а час обробки запити кластером  $c \in T_c / S_c$ . Номінальне завантаження  $q_c$  кластера з можна визначити як

$$q_c = \lambda_c m_{T_c}.$$

Середній час обробки запити кластером  $c$  обслуговування, що має дисципліну «поділ процесора» відповідно до [3], можна записати

$$\frac{m_i / S_i}{1 - \frac{\lambda_i m}{S_i} - q_i}$$

Час відповіді на запит - час, що пройшов з моменту вступу запити в систему до моменту його виходу із системи, тобто сумарний час проходження заявки че-

рез усі ланки системи:

$$T_{cp} = \sum_{c=1}^C \frac{\lambda_c}{\lambda} \left( \frac{m_{T_c}}{S_c - q_c} \right),$$

де  $\frac{\lambda_c}{\lambda}$  — середнє число відвідувань запитом кластера  $c$ ;  $c=1, \dots, C$ .

Модель масового обслуговування в центрах обробки даних з урахуванням сесій і класів запитів

Одержаний вираз не враховує класів запитів і те, що в SLA можуть задаватися обмеження на якість обслуговування для запитів різних видів, тому необхідно розробити модель ЦОД, яка б враховувала таку структуру вхідного потоку заявок. Для кожного класу запитів в SLA можуть задаватися обмеження на середній час відповіді або максимальний час відповіді для заданої частки запитів.

Поведінка користувачів на сайтах характеризується їх сесіями. Сесія користувача складається з послідовності запитів до сайту, а також часу на обмірковування, тобто часу, що пройшов з моменту відправлення web-сервером відповіді й до надходження нового запити в систему в рамках однієї сесії. Типова сесія може тривати кілька хвилин. Тому в моделі необхідно враховувати тривалість сесії життє сесії.

Для різних сайтів користувацькі сесії різні. Будемо припускати, що кожному сайту відповідає одна сесія. Припустимо, що  $c=1, \dots, C$  кластерів, що представляють собою різні ланки застосувань. Для моделювання процесу обмірковування введемо в систему віртуальний сервер ( $c=0$ ) з нескінченим числом паралельних незалежних каналів обслуговування. Нескінченне число паралельних незалежних каналів обслуговування в моделі віртуального сервера характеризує час на обмірковування користувачів. Для прикладу розглянемо випадок, коли сесія складається із двох і більш запитів.

Після виконання першого запиту сесії він надходить у чергу до віртуального сервера ( $c=0$ ), де, провівши якийсь час обмірковування, знову надходить у систему у вигляді наступного запиту. Так триває логі, поки не зробить останній запит сесії, після чого сесія завершується, тобто залишає систему. Сервер з нескінченним числом паралельних незалежних каналів обслуговування дозволяє відбити в моделі незалежність часу на обмірковування користувачів від часу обробки запиту застосування.

Нехай також  $\epsilon k = 1, \dots, K$  класів запитів,  $j$  різних типів сесій,  $i, j$  різних сайтів  $j = 1, \dots, J$ . Ко-жний тип сесії відповідає одному сайту. Так само як і в моделі ЦОД без обліку сесій і класів запитів, припустимо, що зовнішнє джерело надходжень запитів формує Пуассонівський процес. Користувачькі сесії типу  $j$  надходять у систему з інтенсивністю  $l_j$  й почи-наються із запиту класу  $k$ , де  $j = 1, \dots, J$ . Після

виконання запиту класу  $k$  користувачі з типом сесії  $j$  витрачають на обмірковування випадковий час  $d_j$ .

Після цього вони або вертаються в систему із запитом класу  $k'$  з імовірністю  $p_{kk'}$ , або виходять із системи,

звернувшись до сесії, з імовірністю  $1 - \sum_{k'=1}^K p_{kk'}$ . При-

пустимо матриця розмірністю  $K \times K$   $P^j = [p_{kk'}^j]$  є матрицею ймовірностей переходів користувачів по веб-сторінкам сайту  $j$ . Ця матриця визначає послдов-ність надходжень запитів у СеМО в межах користувачької сесії  $j$  і відбиває зв'язок між надхо-дженнями запитів класу  $k$  і  $k'$  від того самого користувача (рис. 2).

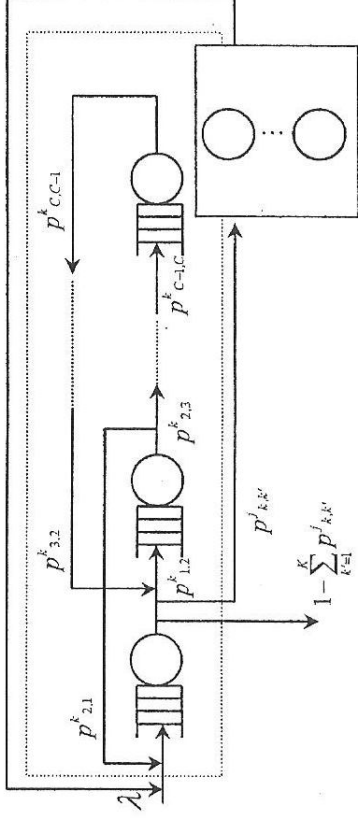


Рисунок 2 – Модель ЦОД урахування класів запитів

Нехай  $\Lambda_k^j$  означає інтенсивність надходжень запи-тів класу  $k$  із сесії  $j$ , яка буде дорівнювати

$$\Lambda_k^j = \sum_{k'=1}^K \Lambda_k^j p_{kk'}^j + l_j, \quad j = \overline{1, J}.$$

Сумарна інтенсивність надходжень запитів класу  $k$  із усіх сесій буде рівна:

$$\lambda^k = \sum_{j=1}^J \Lambda_k^j.$$

Запити класу  $k$  в багатоканковій системі за одне відвідування можуть мати різний маршрут і кілька раз відвідувати різні кластери. Значення ймовірності пере-ходів запиту класу  $k$  між кластерами, можна визначити інтенсивності надходжень запитів класу  $k$  в кожний

кластер  $\lambda_{k,c}^k$ ,  $c = \overline{1, C}$

$$\left\{ \begin{aligned} \lambda_1^k &= \lambda^k + \lambda_{2,c}^k p_{c,c-1}^k \\ \lambda_2^k &= \lambda_{1,c}^k p_{12}^k + \lambda_{3,c}^k p_{32}^k \\ \lambda_3^k &= \lambda_{2,c}^k p_{23}^k + \lambda_{4,c}^k p_{43}^k \\ &\vdots \\ \lambda_c^k &= \lambda_{c-1,c}^k p_{c-1,c}^k \end{aligned} \right.$$

Визначення середнього часу відповіді. Обслугову-ючим приладом в моделі обрано кластер серверів, який, у свою чергу, може бути представлений як сис-тема M/G/1 з дисципліною обслуговування «поділ процесора». Припустимо, що всі сервери, що належать одному кластеру, є однаковими по продуктивності. Вектор розподілу ресурсів  $\bar{S} = (S_1, S_2, \dots, S_c)$  - кож-ному кластеру  $c$  належить  $S_c$  ресурсів (серверів).

Припустимо, що час на обслуговування запиту