УДК 004.021

МЕТОДИКА РОЗРАХУНКУ ПОТУЖНОСТІ СЕРВЕРНОГО ПРОЦЕСОРУ

Маслов Є.Ю., Аноприенко О.Я.
Донецький національний технічний університет

Розробка нових алгоритмів і методик розрахунку серверних параметрів, а також "апгрейд" вже застарілих на даний момент. Вони відкривають величезні можливості в області оптимізації Інтернет-орієнтованих проектів, що дозволить не тільки скоротити фінансові витрати на розвиток наявних потужностей, але й підвищити ефективність адміністрування інфраструктури в цілому. Особливий інтерес такі дослідження мають для ефективної організації обчислень на базі Cloud Computing.

Вимоги

Один із основних параметрів Інтернет системи - час відповіді WEB сервера на запит користувача. Час відповіді складається з часу виконання запиту на сервер, часу очікування виконання запиту і часу відповісти по мережі. Так як в нашому випадку все тестування проводиться в рамках Інтранет мережі та обсяг видаваної WEB сервером інформації невеликий, то час передачі відповіді по мережі можна опустити. Час відповіді повинно бути менше або дорівнює 5 секунд. Крім вимог по предметній області існують і технічні вимоги які повинні дотримуватися в процесі тестування. Основним технічним вимогою є використання процесора не більш ніж на 90%

Планування розміру ресурсів

Завдання з планування конфігурації для програмних систем можливо вирішувати двома основними шляхами. Перший - традиційний підхід (Conventional Approach), має на увазі використання декількох конфігурацій для тестування з наступним порівнянням отриманих на кожній результатів тестів і вибору оптимальної конфігурації.

Інший метод планування конфігурації - метод аналізу вартості транзакції в системі (Transaction Cost Analysis), його завдання визначити вартість в ресурсах для кожної операції виробляється в системі користувачем.

Традиційний підхід

Складається з двох видів тестів:

Stress test - завдання тесту визначити максимальне навантаження на систему при даній конфігурації і виміряти використовувані при цьому ресурси системи: пропускну здатність, завантаженість процесора і т.п. Тест проводиться в кілька ітерацій з поступовим підвищенням навантаження наприклад 1000 користувачів, 2000 користувачів, 3000 користувачів. Після кожної ітерації перевіряються показники роботи системи, якщо показники перевищують зазначені допустимі межі (наприклад CPU Utilization> 90%) то навантаження що викликала це вважається критичною для даної конфігурації і подальші ітерації припиняються.

\Scalability test - завдання тесту підібрати необхідну конфігурацію для системи. Тест складається з декількох ітерацій Stress tests на різних конфігураціях. Якщо при критичному навантаженні на даній конфігурації не виконуються вимоги до деяких параметрів роботи системи, то конфігурація вважається неприйнятним і підбирається нова, в якій найбільш дефіцитний ресурс нарощується.

Аналіз вартості транзакцій

Для вирішення завдання визначення та прогнозування конфігурації найбільш оптимальним методом, на мій погляд, є аналіз вартості транзакції в системі (Transaction Cost Analysis).

Завдання методу визначити скільки ресурсів буде коштувати виконання кожної транзакції в системі. І на основі отриманої інформації спрогнозувати залежність необхідних ресурсів від навантаження. Скористаємося цим методом для вирішення поставленого завдання.

План виконання

1. Визначити час роботи в системі одного користувача (час сесії).
2. Визначити транзакції, які виконує користувач у системі.
3. Визначити завантаження системи для кожної транзакції (вартість транзакції).
4. Визначити профілі використання системи.
5. Визначити вартість ресурсів для одного користувача в системі.
6. Визначити кількість користувачів на добу.
7. Визначити максимальну кількість одночасних користувачів.
8. Прогнозування необхідної кількості ресурсів для вказаної кількості користувачів.

Визначення транзакцій, які виконує користувач у системі

Список цих транзакцій визначається виходячи з функціональності системи. Операціями можуть бути: пошук, запит певної сторінки на сайті і пр.

Визначення часу роботи в системі одного користувача (час сесії). Цей параметр визначається при аналізі лог-файлів системи. Так само параметр можна визначити, аналізуючи бізнес логіку системи. Визначення завантаження системи для кожної транзакції (вартість транзакції). На цьому етапі для кожної транзакції необхідно визначити вартість в ресурсах. Для цього кожна транзакція виконується в системі окремо деякий час, не менше 3 хвилин. Під час її виконання фіксуються показники системних датчиків. Після чого аналізуються середні показники датчиків. Якщо вони в межах встановлених у вимогах обмежень, то навантаження збільшується і проводиться ще один тест. Якщо обмеження перевищені то середні показники датчиків записуються табл. 1

Таблиця 1

Статистичні данні по системі

	Request Execution Time	Request Waite Time	ASP Requests/Sec	Processor utilization%	Response time
Операція 1	4655	355	3	61	5000
Операція 2	1062	0	14	93	1062
Операція 3	43	5	20	94	48
Login	3099	0	3	89	3099

Після чого розраховуэться вартість транзакції за секунду за формулою:
*(% Resource utilization * resource capacity) / (transaction per second) = transaction resource cost*

Приклад. Починаємо виконувати транзакцію Операція 1 з кількості одночасних користувачів в системі N = 10. Поступово збільшуємо кількість користувачів до тих пір, поки обмеження в системі не будуть перевищені: Response time
*(61% * 365MHz) / 3 = 74 MHz вартість виконання однієї транзакції з ім'ям "Операція 1"*

Аналогічно розраховується вартість будь-якого ресурсу. Тепер необхідно розрахувати

вартість кожної транзакції з таблиці 1 за допомогою зазначеної вище формули. Результатом цього етапу стане табл. 2.

Таблиця 2

Вартість кожної операції

Transaction	CPU Cost MHz
Операція 1	74.62333
Операція 2	24.37929
Операція 3	17.249
Login	108.8767

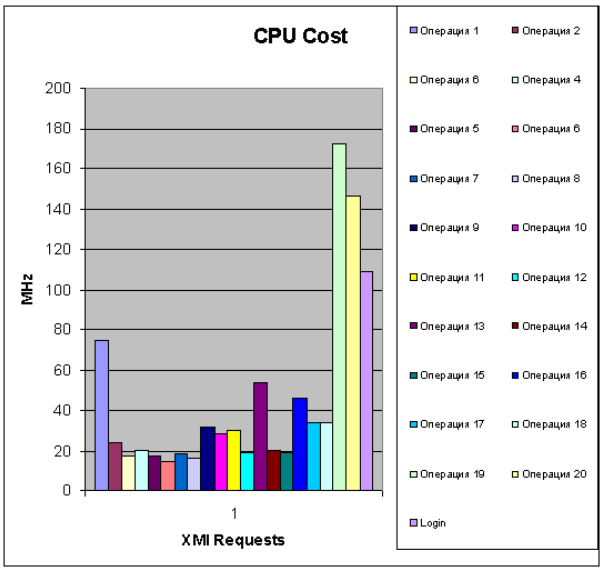

Рисунок 4 – Витрати ресурсів процесора на кожну з транзакцій

Визначення профілю використання системи

Під профілем використання мається на увазі сукупність операцій виконуваних користувачем за сесію в системі і їх кількість. Наприклад, система може використовуватися в режимі активного пошуку інформації, або в режимі перегляду певних документів. Створимо профіль при якому йде активне використання операції 1. Для цього припустимо що користувач виконує 90% транзакцій за викликом операції 1 і 10% витрачає на всі інші.

Таблиця 3

Опис профілю

Transaction	$n_{session}$	n_{sec}^{*}
Операція 1	60	0.05
Операція 2	60	0.05
Операція 3	60	0.05
Login	1	0.000833

* nsec = nsession / t

t - час сесії

nsesion - транзакцій за сесію
nsec - транзакцій за секунду

Визначення вартості ресурсів для одного користувача в системі

На основі профілю використання та вартості кожної транзакції в системі визначимо вартість одного користувача в системі.

Таблиця 4

Вартість на одного користувача в секунду

Transaction	Cost per User Transaction per sec. CPU MHz*
Операція 1	3.731166667
Операція 2	1.218964286
Операція 3	0.86245
Login	0.090730556
Total	Xxxxxxx

*Cost per User Transaction per sec. CPU MHz = CPU Cost MHz * nsec
Вартість користувача в системі в секунду буде дорівнювати сумі вартостей його операцій за секунду.

Визначення кількості користувачів на добу

Кількість користувачів за добу в нашому випадку буде задано як одна з вимог до системи, також цей параметр можна визначити аналізуючи лог-файли системи.

Визначення максимальної кількості одночасних користувачів

Під одночасними користувачами маються на увазі користувачі, які в один і той же проміжок часу виконують транзакції в системі. Кількість таких користувачів визначається з лог-файлів системи, на основі вимог до системи. Так само можна приблизно спрогнозувати цей параметр виходячи з правила 80/20, що означає - 80 відсотків користувачів можуть виконувати транзакції в 20 відсотків часу. Визначимо середню кількість одночасних користувачів системи. В результаті виконання цього пункту розширимо табл. 5.

Таблиця 5

Середня кількість одночасних користувачів

Users per day (Users)	Max Users per second (Nmax)*	Avg Users per second (N)**	CPU MHz max	CPU MHz avg
1000	0.046296296	0.011574	0.269101	0.067275
10000	0.462962963	0.115741	2.69101	0.067275
50000	2.314814815	0.578704	13.45505	3.363762
100000	4.62962963	1.157407	26.9101	6.727524
1000000	46.2962963	11.57407	269.101	67.27524
10000000	462.962963	115.7407	2691.01	672.7524

** Nmax = (0.8 * users) / (24 * 60 * 60 * 0.2)*
*** N = users / (24 * 60 * 60)*

Прогнозування необхідної кількості ресурсів для вказаної кількості користувачів

Total_Cost_CPU_MHz_per_User_Transaction per sec - вартість ресурсів процесора на одного користувача в секунду.

Необхідний CPU MHz для системи =

Total_Cost_CPU_MHz_per_User_Transaction_per_sec * Users.

Тобто, якщо пропускна здатність системи повинен дорівнювати 100 користувачам в секунду, то необхідний для цього CPU MHz буде 5.8 * 100 = 580 Mhz при заданому профілі

Аналогічно розраховуються і вимоги по інших ресурсів. В результаті отримаємо табл. 6.

Таблиця 6

Підсумкова таблиця параметрів

Users per day (Users)	Max Users per second (Nmax)	Avg Users per second (N)	CPU MHz max	CPU MHz avg
1000	0.046296296	0.011574	0.269101	0.067275
10000	0.462962963	0.115741	2.69101	0.067275
50000	2.314814815	0.578704	13.45505	3.363762
100000	4.62962963	1.157407	26.9101	6.727524
1000000	46.2962963	11.57407	269.101	67.27524
10000000	462.962963	115.7407	2691.01	672.7524

З табл. 6 видно, що для обслуговування 10000000 користувачів у день можлива необхідна пікова пропускна здатність FWS в 463 користувача в секунду і для її забезпечення потрібно мультипроцесорна система яка буде давати сумарну продуктивність в 2700 MHz.

Висновки

Виконані розробки і дослідження дозволили отримати алгоритми і інструментарій для розрахунку основних параметрів серверної інфраструктури. Подальші дослідження дозволять адаптувати отримані результати під потреби організації ефективних обчислень на базі Cloud Computing.

Література

[1] Шолл Ф. Азбука планирования нагрузки. Журнал LAN№8, 1996г

[2] "Capacity Planning and Performance Testing", Microsoft Developer Network, January 2002 http://msdn.microsoft.com/library/en-us/dncold/html/storcpctyplan.asp?frame=true

[3] "Performance Optimization and Capacity Planning", Microsoft Corporation, January 2002 http://msdn.microsoft.com/library/en-us/dnmscms01/html/cmsperfoc.asp?frame=true

[4] "Performance Testing with the Web Application Stress Tool", Microsoft Developer Network, January 2001

[5] http://msdn.microsoft.com/library/en-us/dnduwon/html/d5wast_2.asp?frame=true