УДК 681.3, 621.3

Навантажувальне тестування мережного інтерфейсу клієнтськими запитами

Пастернак І. І., аспірант 2-го року навчання
Морозов Ю. В., к.т.н., доц. каф. ЕОМ

Національний університет «Львівська політехніка»
(вул. С. Бандери, 12, м. Львів, 79013, Україна)

Зрозуміло, що в загальному випадку, щоб прикладна програма, що виконується на робочій станції, могла запросити послугу у деякого сервера, як мінімум потрібно деякий інтерфейсний програмний шар, який підтримує такого роду взаємодія (було б щонайменше неприродно вимагати, щоб прикладна програма безпосередньо користувалася примітивами транспортного рівня локальної мережі). З цього, власне, і випливають основні принципи системної архітектури "клієнт-сервер"[2].Система розбивається на дві частини, які можуть виконуватися в різних вузлах мережі, - клієнтську та серверну частини. Прикладна програма або кінцевий користувач взаємодіють з клієнтською частиною системи, яка в простому випадку забезпечує просто надсетевой інтерфейс. Клієнтська частина системи при потребі звертається по мережі до серверної частини. Зауважимо, що в розвинених системах мережеве звернення до серверної частини може і не знадобитися, якщо система може передбачати потреби користувача, і в клієнтської частини містяться дані, здатні задовольнити його наступний запит [1]. Інтерфейс серверної частини визначений і фіксований. Тому можливе створення нових клієнтських частин існуючої системи (приклад інтероперабельності на системному рівні). Основною проблемою систем, заснованих на архітектурі "клієнт-сервер", є те, що відповідно до концепції відкритих систем від них вимагається мобільність в якомога більш широкому класі апаратно-програмних рішень відкритих систем. Спроби створення систем, що підтримують всі можливі протоколи, призводить до їх перевантаження мережевими деталями на шкоду функціональності.

Ще більш складний аспект цієї проблеми пов'язаний з можливістю використання різних представлень даних в різних вузлах неоднорідною локальної мережі. У різних комп'ютерах може існувати різна адресація, представлення чисел, кодування символів і т.д. Це особливо важливо для серверів високого рівня: телекомунікаційних, обчислювальних, баз даних.

Спільним рішенням проблеми мобільності систем, заснованих на архітектурі "клієнт-сервер" є опора на програмні пакети, що реалізують протоколи віддаленого виклику процедур (RPC - Remote Procedure Call). При використанні таких коштів звернення до сервісу у віддаленому вузлі виглядає як звичайний виклик процедури. Засоби RPC, в яких, природно, міститься вся інформація про специфіку апаратури локальної мережі та мережевих протоколів, переводить виклик в послідовність мережевих взаємодій. Тим самим, специфіка мережного середовища і протоколів прихована від прикладного програміста [3].

Навантажувальне тестування мережного інтерфейсу з клієнт-серверною архітектурою призначене для визначення максимальної кількості запитів від клієнта (пропускної здатності) та оптимальної конфігурації сервера при наступних станах мережного інтерфейсу:

- нормальному (base);
- навантаженому (load);
- критичному (stress).

У всіх випадках при тестуванні відбувається визначення часу стабільного перебування у фіксованому стані і межа переходу у наступний (рис. 1).

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

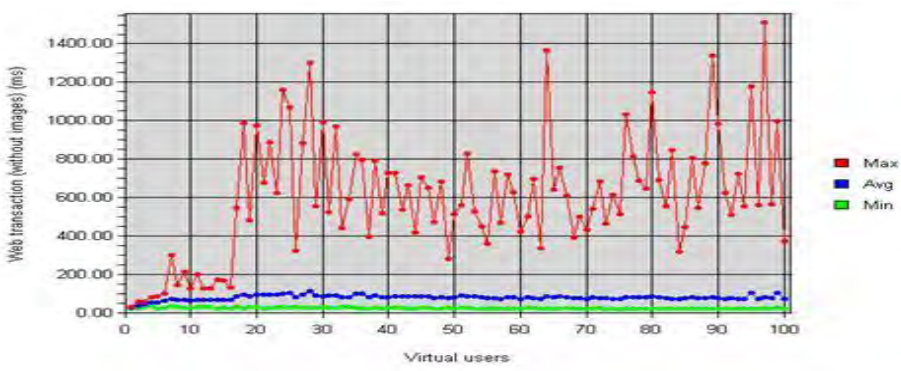

Рис. 1 Залежність часу відгуку сервера від кількості запитів від клієнта

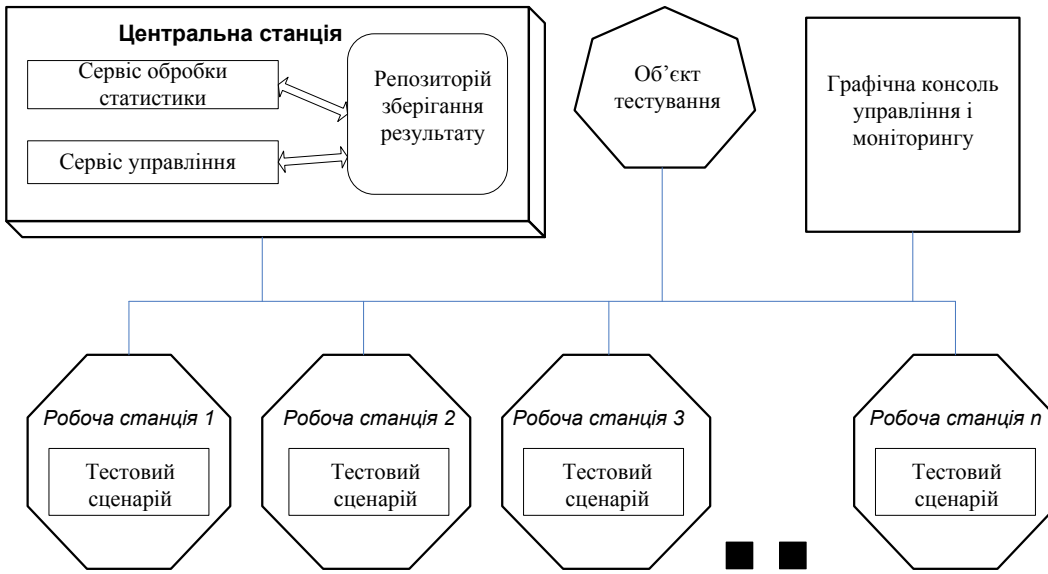

Рис. 2 Структура випробувального стенду

Зазначимо, що після перевищення межі критичного навантаження відбувається збій мережного інтерфейсу різного ступеня складності.

Етапність, методика і інструментарій. У проектах по тестуванню навантаження можна виділити наступні етапи:

- аналіз і проектування моделі навантаження;
- налаштування випробувального стенду (рис. 1);
- розробка типових сценаріїв;
- реалізація моделі навантаження;
- проведення навантажувального тестування;
- аналіз результатів випробувань і побудова моделей прогнозу.

Необхідною основою для проведення навантажувального тестування є інструментальні засоби, що дозволяють в лабораторних умовах емулювати складне оточення реального світу телекомунікаційних, клієнт-серверних і Інтернет-взаємодій і виконати усестороннє тестування мережного інтерфейсу. Завдання таких інструментальних засобів – організація лабораторного випробувального стенду, який емулюватиме від десятків до тисяч

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

користувачів, що надсилають і одержують інформацію, відтворюючи тим самим складну взаємодію між клієнтським і серверним ПЗ, базами даних, Інтернет-серверами і іншими системами [4].

Більшість розвинених інструментальних засобів тестування навантаження. наприклад WAPT, Apache, JMeter, Mercury LoadRunner, дозволяють організувати випробувальний стенд з розподіленим виконанням тестів і централізованим управлінням.

Як правило, структура такого стенду включає:

1. Об'єкт випробувань;
2. Центральну станцію управління, де розташовуються сервіси, репозиторій результатів тестування і графічна консоль користувача;
3. Декілька робочих станцій моделювання навантаження, на яких розташовані компоненти для імітації діяльності користувачів (до декількох сотень кожна);
4. Додаткові станції моніторингу і генерації тестових даних для забезпечення безперебійного постачання тестовими даними робочих станцій. Також на окремих станціях запускаються вимірювальні монітори, що здійснюють спостереження за процесами обробки і передають виміряну статистику на центральну станцію.

Моделювання навантаження здійснюється шляхом виконання тестових транзакцій, кожна з яких включає певний набір тестів, що виконуються в послідовному або довільному порядку (рис. 3).

Важливим є також визначення максимального часу відгуку серверу, при якому відбувається робота системи у заданому стані (рис. 4).

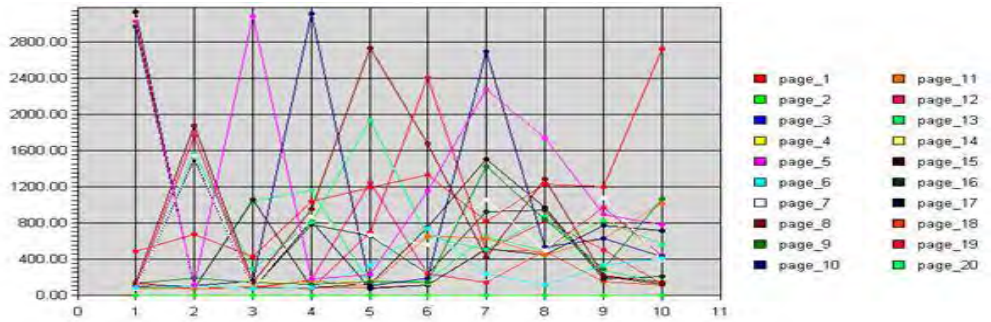

Рис. 3 Залежність середнього часу (в мс.) завантаження всіх сторінок (Pages) зі сценарію для кожного віртуального користувача (Virtual Users)

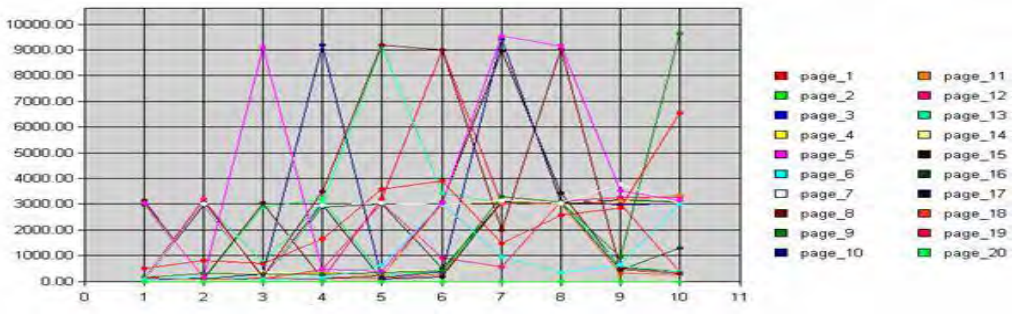

Рис. 4 Максимальний час (в мс.) завантаження всіх сторінок зі сценарію для кожного віртуального користувача

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Розглянемо проблему формування ймовірностей для опису реакції мережного інтерфейсу на запит. Можливі два підходи до її розв'язання:

1) визначення ймовірності конкретної реакції мережного інтерфейсу на запит користувача зі врахуванням значення конкретного параметра;
2) визначення загальної ймовірності конкретної реакції мережного інтерфейсу на запит користувача без врахування значення конкретного параметра;

Перший підхід забезпечує максимально можливу визначеність реакції мережного інтерфейсу на запит користувача. Проте недоліком такого підходу є велика складність визначення імовірності переходу. Крім того, при використанні операції усунення параметра потрібно буде перераховувати імовірності переходів. Для проведення фізичної оптимізації також доцільно використовувати загальні значення ймовірностей реакції мережного інтерфейсу. Тому доцільніше використовувати другий підхід. У такому разі ймовірність мережного інтерфейсу визначається для кожної пари (St,Q) без врахування значень параметрів запиту до мережного інтерфейсу та її бази даних. Визначеність поведінки мережного інтерфейсу буде меншою ніж у першому випадку, проте для усунення параметрів з розгляду та фізичної оптимізації мережного інтерфейсу непотрібно буде додаткових перетворень. Якщо в мережного інтерфейсу існують запити з такими параметрами, що суттєво впливають на реакцію системи та є важливими для її оптимізації, тоді пару (параметр, запит) можна перетворити в окремий запит.

1. Липаев В. В. Обеспечение качества программных средств. Методы и стандарты. - М.: Синтег, 2001.- 246 с.
2. Макгрегор Дж., Сайкс Д. Тестирование объектно-ориентированного программного обеспечения.- К: Диасофт, 2002.- 432с.
3. Тамре Л. Введение в тестирование программного обеспечения.- М.: Издательский дом "Вильямс", 2003.- 368с.
4. Кеннет Г. "Основы сетей Windows". - К.:Диалектика.
5. Бормотов С.В. Системное администрирование на 100%. – СПб.; Питер, 2006. – 256.:ил.
6. Єд Уілсон ,, Моніторинг і аналіз мереж" Москва видавництво ,,Лори" 2002р.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua