УДК 004.94

МОДЕЛІ РОБОЧОГО НАВАНТАЖЕНЯ ДЛЯ ВИЗНАЧЕННЯ ПРОДУКТИВНОСІ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

В.М. Томашевський
Національний технічний університет України «КПІ»

Для визначення продуктивності обчислювальних систем застосовуються методи вимірювання та моделювання. Методи вимірювання поділяються на апаратні, програмні та комбіновані. Ці методи використовуються за наявністю обчислювальної системи, але виникає питання, з яким робочим навантаженням проводити виміри. Робоче навантаження – це сукупність усієї вхідної інформації (програм, даних, команд), що надходить у обчислювальну систему ззовні.

У загальному випадку моделі робочого навантаження поділяють на природні та штучні. Якщо модель – це просто вибірковий потік завдань, узятий із довільного робочого навантаження, то це природна модель робочого навантаження. Така модель може використовуватися тільки у вимірювальних системах. У всіх інших випадках маємо штучну модель робочого навантаження. Якщо штучна модель складається з однієї або більш програм, то вона називається штучною виконуваною моделлю й може використовуватися для вимірювання та моделювання систем.

Однією з перших штучних моделей були виконуючі суміші команд для вимірювання загального часу центрального процесора. Стандартні суміші команд, серед яких найбільш відомі суміші Гибсона та Флінна, відповідно застосовуються для комерційних і наукових задач.

Суміш команд для реального робочого навантаження є частотним розподілом типів команд, виконуваних під час обробки робочого навантаження. Набір цих частот є відносними запитами на процесорні ресурси.

Суміші можуть використовуватися й у мовах більш високого рівня, ніж машинний. Для алгоритмічних мов можна використати суміші речень, для діалогових мов – суміші команд, для мов керування завданнями – суміші керівних речень. Основні вади сумішей зв’язані із тим, що з допомогою сумішей не враховується взаємодія між виконуючою послідовністю команд і ЦП та іншими ресурсами системи. Отже, з допомогою сумішей не можна визначити дійсну пропускну спроможність системи.

Моделі робочого навантаження, придатні для діалогової системи, можуть розглядатися як категорії ядер, що складаються із сценаріїв.

Сценарій – це послідовність діалогових команд, розділених інтервалами для обміркування. Передбачається, що ця послідовність є типовим сеансом роботи користувача за терміналом.

Жодна з цих моделей не може відтворити реальне робоче навантаження та призвести до адекватної оцінки продуктивності мультипрограмних ЕОМ, оскільки в цих моделях:

– не враховується характер робочого навантаження користувача;

– не розглядається ефективність системного програмного забезпечення, зокрема робота модулів ОС;

– слабо представлені можливості апаратних засобів.

Зокрема, ігнорується інформаційна місткість системи команд, паралелізм у роботі пристроїв, своєрідність системи введення-виведення та багато інших параметрів.

Теоретичний максимум швидкодії комп'ютера за самих ідеальних умов визначає пікову продуктивність, яка є єдиною об'єктивною оцінкою. Важливо також розуміти різницю між продуктивністю процесора та продуктивністю обчислювальної системи. Остання мусить враховувати використання різних ресурсів під час виконання програм і буде різнитися для різних типів спрямованості завдань. Зазвичай, розрізняють науково-технічні, комерційні завдання, роботу з базами даних, оброблення зображень і транзакцій. Крім того, є завдання для пакетного режим (Batch Mode) та оперативного режиму обробки транзакцій (OLTP), що, також потребує різних тестових наборів завдань, які відтворюють робоче навантаження.

Отже, тест – це стандартна процедура для виміру або оцінки продуктивності обчислювальної системи. Комп'ютерний тест – це комп'ютерна програма, що строго виконує певний набір команд (робоче навантаження) та видає як результат деякий показник, що характеризує ефективність роботи комп'ютера. Показники комп'ютерних тестів, зазвичай, визначають швидкість роботи комп'ютера, тобто, наскільки швидко він справляється з робочим навантаженням. Тестові показники також можуть визначати пропускну спроможність, тобто з яким обсягом роботи справляється комп'ютер за заданий інтервал часу. З допомогою запуску одного й того ж тесту на різних комп'ютерах можна порівнювати ефективність їхньої роботи. Під benchmark (тестом) розумітимемо алгоритм або метод, програму або програмний комплекс, що відповідає низки вимог: повнота, легкість у використанні, масштабованість для різного апаратного забезпечення, переносимість (наявність мови та компілятора

під різні платформи), доступність його початкового коду, відтворюваність (отриманням аналогічних результатів у разі повторного використання).

На сьогодні є велика кількість програм, що підпадають під означення benchmark's. Для зручності тести краще поділити на декілька категорій.

«Іграшкові» (toy benchmarks) тести — маленькі, завдовжки в декілька сотень рядків початкового коду. Зазвичай вони розв'язують одну дуже відому задачу, наприклад, Решето Ератосфену, пірамідальне сортування, перемішування та багато інших.

Мікротести (micro benchmarks) — спеціалізовані, спрямовані на визначення однієї з основних кількісних характеристик апаратного забезпечення:

– продуктивність центрального процесора;

– продуктивність і пропускна здатність локальної оперативної пам'яті;

– швидкість базових операції введення-виведення;

– продуктивність і пропускна спроможність локальної обчислювальної мережі.

У групу мікротестів входять тести оцінки продуктивності операцій, що вимагають синхронізації та тести операційної системи (перемикання контекстів, системні виклики та створення процесів). Часто мікротести об'єднуються в пакети тестів. Прикладом може слугувати тест AnTuTu Benchmark — найвідоміший тест для смартфону або планшета. Він тестує відразу за декількома критеріями: швидкість картки пам'яті; обсяг вбудованої пам'яті (RAM); 2d і 3d графіку (GPU); процесор (CPU); пристрої введення-виведення (I/O) і виводить загальну оцінку тестованого пристрою в балах. Після цього можна порівняти отримані бали й результати роботи інших мобільних пристроїв.

Тести як ядра (kernels) — це фрагменти коду, узяті з реальних застосувань. Саме ці фрагменти виконуються застосуванням більшу частину часу. Ядра дозволяють визначити швидкість виконання реальної програми на різних платформах.

Синтетичні тести (synthetic benchmarks) оцінюють продуктивність на основі набору великої кількості показників і не прив'язані до якого-небудь окремого застосування. Вони використовують параметричне налаштування тесту під конкретне застосування.

Типові застосування (application benchmarks) – найчастіше використовувані програми для реалізації тих або інших завдань. До них можна віднести і псевдо-застосування — це програми, створені на

основі реальних застосувань, але адаптовані з різних причин спеціально для завдань тестування.

Пакети тестів (benchmarks suites) – колекції різних типів тестів із переважанням застосувань.

З усієї множені тестів є і такі, які спеціально спрямовані на тестування масивно-паралельних систем MPP. Найширше вони представлені у класі ядер для визначення ефективності паралельних обчислень. Це тести High-Performance Linpack (HPL) [1], тести для застосувань NAS Parallel Benchmarks (NPB) [2] і тестові пакети SPEC MPI2007 [3]. З мікротестів велику цікавість можуть представляти програми, спрямовані на дослідження сітьового оточення (наприклад, Netperf).

Специфікою ядер і застосувань є те, що вони «прив'язані» до одного з основних алгоритмів чисельних методів або базових операції, тобто до тієї частини програми, виконання якої займає найбільший час. У зв'язку з цим можна заздалегідь вибрати потрібний тест, знаючи, які завдання завантажуватимуться в обчислювальну систему. Якщо розрахунки зв'язані із застосуванням прямих методів розв'язування систем лінійних рівнянь алгебри (СЛРА), то можна використати LINPACK, у разі застосування ітераційних методів розв'язування СЛРА, потрібно використати, наприклад, ядра SP, BP з NPB і Iterative Solver Benchmark.

Є також популярний набір незалежних тестів SPC (Storage Performance Council) для аналізу продуктивності підсистем зберігання даних. Ці підсистеми включають такі компоненти, як: електронні диски, магнітні диски, магнітні стрічки, оптичні диски, медіароботів, засобів масової інформації роботи програмних систем, програмного забезпечення засобів масової інформації бібліотечних систем, систем резервного копіювання або архівних програм, ієрархічних систем керування зберіганням даних, а також адаптери, контролери та мережі, які підключають пристрої зберігання даних у комп'ютерну систему.

Основні метрики тестового набору SPC-1 – кількість операцій введення-виведення за секунду (IOPS), а також вартість операцій введення-виведення за секунду ($/IOPS) й усереднений час відгуку (Response Time). Версія тесту SPC-2 розширена засобами аналізу ефективності обробки великих файлів і громіздких запитів до баз даних. Основна метрика – пропускна спроможність (throughput).

З додаванням у 2011 р. тестових наборів SPC-2/E і SPC-2C/E до раніше випущених розширень SPC-1/E і SPC-1C/E SPC ці тести охоплюють повний спектр стандартних тестів зберігання для отримання

показників: порівняльна продуктивність, ціна - продуктивність і енергетичні витрати використання даних. Цей всеосяжний пакет тестів стосується оперативної обробки транзакцій (OLTP), а також послідовного застосування як для складних конфігурацій зберігання так і зберігання компонентів, використання операцій введення-виведення робочого навантаження, які відтворюють функціонування реальних застосувань.

Розглянуті моделі оцінки продуктивності мають обмежену сферу застосування. Вони призначені для вирішення проблеми оцінки продуктивності ЕОМ з урахуванням специфіки майбутнього робочого навантаження. Можливості ж моделювання як інструменту дослідження наявних і проектованих обчислювальних систем набагато ширше. З допомогою засобів моделювання розв'язуються задачі: вибору первинної конфігурації проектованої та реконфігурації вже наявної обчислювальної системи; аналізується вплив передбачуваних змін у робочому навантаженні або в ресурсах ЕОМ на продуктивність усієї системи; прогнозуються оптимальні режими функціонування; перевіряються альтернативні стратегії проектування та досліджуються багато інших проблем.

Умовно виділяється два класи моделей – аналітичні та імітаційні. В аналітичних моделях дуже важко врахувати функціонування програмного забезпечення, особливо роботу керівних програм ОС. Аналітичні моделі відрізняються за типами математичного апарату, що використовується, але частіше для опису обчислювального процесу використовуються марківські моделі.

Розглядають два підходи до побудови імітаційних моделей. Перший із них базується на дослідженні емпіричних даних і співвідношень, що характеризують функціонування обчислювальної системи. Цей підхід дозволяє відтворити роботу обчислювальної системи тільки в минулому часі, коли збирались емпіричні дані. Другий підхід передбачає моделювання виконуваних системою операцій у прив'язці до календаря поточних подій. Функціонування деяких ресурсів системи задається з допомогою задавання ймовірносних законів розподілу часу використання ресурсів, які вважаються незмінними в часі.

У імітаційних моделях другого типу з різним ступенем деталізації, залежно від поставленої мети, відтворюється весь механізм мультипрограмування, мультипроцесування або розділення часу. Зазвичай, використовують мови дискретно-подійного моделювання, таки як GPSS [4] і інші, яки дозволяють детально моделювати процеси, що протікають у обчислювальній системі. Ступень детальності

опису залежить від мети моделювання та дозволяє відтворювати роботу прикладних і системних програм.

На сьогодні є спеціалізовані засоби моделювання обчислювальних систем і мереж із готовими до використання бібліотеками для моделювання різних конфігурацій таких систем. Одним із перших таких засобів була пакет CSS (Computer System Simulation), побудований за блочним принципом і схожий на мову GPSS.

Для моделювання модель робочого навантаження можна представити як сукупність моделей завдань. Модель кожного завдання складається з *v* рядків параметрів завдання, більшість із яких є запитами ресурсів для цього завдання. Найпоширенішими типами параметрів завдання є скалярний параметр і послідовність. Остання є часовим рядом скалярних змінних (детермінованих або випадкових). Наприклад, де *l* – число рядків друку (скаляр), а решта – послідовності, у яких *l* позначає довжини записів на магнітних носіях.

Аналітичні моделі здебільшого базуються на теорії масового обслуговування з використанням мереж масового обслуговування (замкнених або розімкнених), в яких робоче навантаження задається як розподіл часу надходження вимог у обчислювальну систему з ймовірностями зайняття певних ресурсів. Ці моделі найбільш узагальнені, але дозволяють у деяких випадках визначити час відгуку на запит користувача з точністю до 10 %.

Література

1. A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary. HPL — A Portable Implementation of the HighPerformance Linpack Benchmark for Distributed-Memory Computers // Netlib Repository, [Electronic resource]. – Access mode: http://www.netlib.org/benchmark/hpl/, 2008.
2. The NAS Parallel Benchmarks // NASA Advanced Supercomputing Division, http://www.nas.nasa.gov/Resources/
Software/npb.html, 2008.
3. SPEC MPI2007 // Standard Performance Evaluation Corporation, [Electronic resource]. – Access mode: http://www.spec.org/mpi/, 2008.
4. Томашевский, В.Н., Жданова, Е.Г. Имитационное моделирование в среде GPSS [Текст]. – М.: Бестселлер, 2003, – 416 с.