УДК 519.8

Прогнозування ефективності багатокомпонентних обчислювальних систем

Василь Горбачук[1], Максим Дунаєвський[2] , Сеїт-Бекір Сулейманов[3]

[1] д. ф .-м. н., с. н. с., Інститут кібернетики імені В.М. Глушкова НАН України, просп. Глушкова 40, 03187, Київ, e-mail: GorbachukVasyl@netscape.net
[2] магістр, Інститут кібернетики імені В.М. Глушкова НАН України, просп. Глушкова 40, 03187, Київ, e-mail: MaxDunaievskyi@gmail.com
[3] магістр, Інститут кібернетики імені В.М. Глушкова НАН України, просп. Глушкова 40, 03187, Київ, e-mail: SBSuleimanov@gmail.com

Показано переваги та недоліки відомих у галузі інформаційно-комунікаційних технологій законів Мура, Гілдера, Амдала, Густафсона–Барсіса, пропонуючи при цьому необхідний математичний апарат до побудови подібних законів прогнозування ефективності сучасних багатокомпонентних обчислювальних систем. Цей апарат включає як параметри компонентів обчислювальних систем, так і можливі взаємозалежності між цими параметрами. В цілому прогнозування ефективності обчислювальних систем потребує докладного документування роботи обчислювальних систем даного класу (даної серії) на певних типах завдань з подальшою обробкою отриманих даних. Збір та обробка цих даних мають відбуватися і реєструватися в динаміці за допомогою відповідних інтелектуальних датчиків класу Інтернету речей.

Ключові слова: рівні паралелізму, хмарні обчислення, коефіцієнт послідовного звуження, прискорення, робоче навантаження, великі дані

Вступ. Технології, що є рушійною силою застосувань обчислювальних ресурсів, мають передбачувані тренди. Наприклад, за законом Мура, швидкість процесора подвоювалася кожні півтора роки в минулому столітті; Гордон Мур (Gordon Moore) народився у 1929 р., у 1954 р. отримав науковий ступінь доктора філософії з хімії у Каліфорнійському інституті технології (California Institute of Technology, Caltech), у 1968 р. заснував корпорацію Intel (INTC у лістингу біржі NASDAQ), яку очолював до 1997 р., коли став її почесним президентом. Зазначені тренди важливі для передбачення технологічних спроможностей майбутніх обчислювальних систем, які проектуються та програмуються.

1. Закон Гілдера та рівні паралелізму

За законом Гілдера, пропускна здатність мережі подвоювалася щороку в минулому столітті; Джордж Гілдер (George Gilder) народився у 1939 р., у 1962 р. отримав науковий ступінь бакалавра з державного управління у Гарвардському університеті, працював спічрайтером відомих політиків США і став інвестором у

високотехнологічні галузі. Водночас на зростання відношення ціни до продуктивності товарного апаратного забезпечення впливає ринок настільних комп'ютерів (desktops), ноутбуків і планшетів, що також впливає на прийняття та використання товарних технологій у великомасштабних обчисленнях. Закони Мура і Гілдера потребують модифікацій для систем, де використовуються розподіл ресурсів і паралелізм (concurrency) чи високий ступінь паралелізму. Хмарні обчислення допускають різні типи паралелізму.

Більшість перших товарних комп'ютерів проектувалася для послідовної обробки бітів інформації, коли апаратне забезпечення було громіздким і дорогим. Паралелізм на рівні бітів (bit-level parallelism, BLP) поступово перетворював послідовну обробку бітів на обробку бітів на рівні слів. З часом користувачі переходили від 4-бітових (4-розрядних) мікропроцесорів до 8-, 16-, 32- і 64-розрядних CPUs. Наступним кроком розвитку паралелізму став паралелізм на рівні інструкцій (instruction-level parallelism, ILP) [1]. Переходячи від використання процесорів для виконання окремих інструкцій до процесорів для виконання декількох інструкцій одночасно, ILP втілювався через конвеєризацію (pipelining), суперскаляризацію, багатопоточність (multithreading), паралелізм на рівні дуже довгого слова-інструкції (very-long instruction word, VLIW). ILP потребує передбачення розгалужень, динамічного планування, формування гіпотез (speculation) і підтримки компілятора [2].

Паралелізм на рівні даних (data-level parallelism, DLP) поширився завдяки окремій інструкції з множинними даними (single instruction and multiple-data, SIMD), а також векторним машинам з використанням інструкцій типу вектора чи масиву (array). Належна дія DLP вимагає підтримки не лише компілятора, але й апаратного забезпечення. Впровадження багатоядерних процесорів і мікросхемних мультипроцесорів (chip multiprocessors, CMPs) сприяло паралелізму на рівні завдань (task-level parallelism, TLP). На сучасних процесорах застосовуються вищезазначені типи паралелізму − BLP, ILP, DLP, TLP, паралелізм на рівні VLIW тощо.

Якщо BLP, ILP, DLP дістають належну підтримку сучасного апаратного забезпечення і відомих компіляторів, то TLP не дістає такої підтримки для ефективної дії на багатоядерних процесорах і мікросхемних мультипроцесорах через складність програмування та компіляції кодів. При переході від паралельної до розподіленої обробки даних зростає обчислювальна гранулярність (computing granularity) до паралелізму на рівні робіт (job-level parallelism, JLP); гранулярність − це характеристика паралельних обчислень, яка визначається відношенням часу обчислень у конкретному завданні (task) до часу комунікації з паралельними йому завданнями (процесами чи потоками). Грубозернистий (coarse-grain) паралелізм, де час комунікації є порівняно малим, будується на тонкозернистому (fine-grain) паралелізмі, де час комунікації є порівняно великим.

2. Закони Амдала та Густафсона-Барсіса

Масштабованість (scalability) і доступність (availability) хмарних кластерів є фундаментальними характеристиками роботи хмар. У комп'ютерній архітектурі потенційне прискорення алгоритму при збільшенні числа процесорів оцінив Джин Амдал (Gene Amdahl, 1922−2015), який у 1952 р. успішно захистив дисертацію в Університеті Вісконсін − Медісон (Wisconsin − Madison) на здобуття наукового ступеня доктора філософії, створивши Вісконсінський інтегрально синхронізованй комп'ютер.

Для оцінки масштабованості можна скористатися згаданими законами Амдала і Густавфсона-Барсіса, а для оцінки доступності - зв'язок між середнім часом до відмови і середнім часом для ремонту.

Загальний час (time) виконання (типової) коду (програми) обчислюється як $t = \alpha T + n^{-1}(1-\alpha)T$, де αT – час послідовного виконання коду на одному сервері, $n^{-1}(1-\alpha)T$ – час паралельного виконання коду на n серверах ($t = T$ при $n = 1$), α – певний коефіцієнт (послідовного звуження (sequential bottleneck)) на відрізку $[0,1]$; для простоти тут нехтуються всі системні та комунікаційні накладні витрати (overheads) серед n серверів. При аналізі роботи CPU (сервера) також нехтуватимемо часом вводу/виводу (input/output, I/O) чи часом обробки виняткових програм. За законом Амдала, коефіцієнт прискорення (speedup factor) при використанні n-серверного кластера відносно використання окремого сервера дорівнює

$$S = \frac{T}{t} = \frac{1}{\alpha + n^{-1}(1-\alpha)}.$$

Оскільки $\frac{\partial S}{\partial \alpha} = \frac{-1-n^{-1}}{(\alpha + n^{-1}(1-\alpha))^2} < 0$, то значення S максимізується при $\alpha = 0$, тобто при $t = n^{-1}T$ і виключно паралельному виконанні коду. З іншого боку, $S \to \alpha^{-1}$ при $n \to \infty$, тобто при $\alpha \neq 0$ значення S обмежується зверху величиною α^{-1}, яка не залежить від розміру n кластера. Оскільки в законі Амдала неявно припускається, що робоче навантаження (workload) чи розмір W завдання є фіксованим $\forall\ n$, то S називають прискоренням фіксованого робочого навантаження (fixed workload speedup). Коефіцієнт α послідовного звуження − це частина коду, яка не допускає паралелізму. Наприклад, при $\alpha = 0.2$ значення S обмежується зверху величиною $\alpha^{-1} = 5\ \forall\ n$. За законом Амдала, коефіцієнт прискорення зростає при зменшенні α, а не при збільшенні n: програмна структура є послідовною по своїй суті. Збільшення α означає збільшення неактивних (idle) серверів даного кластера.

Через 20 років після публікації закону Амдала [3], Національні лабораторії Sandia (Sandia National Laboratories, SNL) США перевіряли його на масовій

паралельній обробці; SNL засновані у м. Альбукерке (Albuquerque) штату Нью-Мексико (столиця штату − м. Санта Фе, де розташований Інститут Санта Фе (Santa Fe Institute, SFI) (заснований у 1984 р. деякими дослідниками Лос-Аламоської національної лабораторії (Los Alamos National Laboratory, LANL), відомої за Манхеттенським проектом (Manhattan Project)), з яким успішно співпрацював Інститут кібернетики імені В.М. Глушкова НАН України [4]) за Манхеттенським проектом [5]. Вимірювання часу обробки на 1024-процесорній системі показало, що згадане припущення закону Амдала не відповідає сучасному підходу до масового ансамблевого (massive ensemble) паралелізму.

Якщо $n=1024$, $S=100$, то за законом Амдала

$$100=\frac{1}{\alpha+1024^{-1}(1-\alpha)},\quad 100\alpha+10.24^{-1}-10.24^{-1}\alpha=1,\quad \alpha=\frac{1-10.24^{-1}}{100-10.24^{-1}}=0.9\%.$$

Водночас експерименти SNL при $0.4\%\le\alpha\le 0.8\%$ на 1024-процесорному гіперкубі показали: 1) прискорення 1021 методу спряжених (conjugate) градієнтів для аналізу напруг балки; 2) прискорення 1020 методу явних скінченних різниць для моделювання відбитої поверхневої хвилі; 3) прискорення 1016 методу корекції потоків переносу для моделювання нестійкого потоку рідини.

Звідси $\alpha=\alpha(n)$: для розв'язання завдання фіксованого розміру різні величини n використовуються при теоретичних дослідженнях, а при практичних дослідженнях величина n масштабується залежно від зазначеного розміру. На практиці користувачі застосовують не обов'язково однакові процесори, вибирають роздільну здатність сітки (grid resolution), кількість проміжків часу (time steps), складність різницевого оператора та інші параметри, які зазвичай підбираються так, щоб програма могла виконуватися за деякий бажаний період часу. Тому фіксованим є радше такий період часу, ніж розмір завдання.

Виконання фіксованого робочого навантаження на кластері з n серверів при паралельній обробці програми може давати ефективність (efficiency)

$$E=\frac{S}{n}=\frac{1}{n(\alpha+n^{-1}(1-\alpha))}=\frac{1}{n\alpha+1-\alpha}.$$

Експерименти SNL виявили, що до розміру завдання масштабується, насамперед, паралельна чи векторна частина програми. Від розміру W завдання не залежать тривалості введення (start-up) вектора, завантаження програми, серійних (послідовних) звужень, I/O, які визначають величину α. При подвоєнні числа процесорів обсяг роботи, яку можна виконати паралельно, лінійно залежатиме від цього числа: у вищезгаданих прикладах 1), 2), 3) паралельне виконання програми масштабувалося на коефіцієнти 1023.9969, 1023.9965, 1023.9965 відповідно, тобто майже на 1024. Тоді αW − частина послідовного виконання завдання, $n(1-\alpha)W$ − частина паралельного виконання завдання, W − виконуваний розмір завдання при $n=1$, $\alpha W+n(1-\alpha)W$ − виконуваний

розмір завдання на n процесорах, а прискорення Густавфсона-Барсіса (Едвін Барсіс очолював проект SNL, в якому брав участь Джон Густафсон) становитиме

$$S_{GB} = \frac{\alpha W + n(1-\alpha)W}{W} = \alpha + n(1-\alpha).$$

Висновки. З розвитком інформаційно-комунікаційних технологій закони ефективності обчислювальних систем потребують перевірки і перегляду, враховуючи новітні можливості збору й обробки великих даних.

Література

[1] *Горбачук В.М.* Параллельные методы решения задач двухуровневого программирования и их применения. Компьютерная математика. 2008. № 1. С. 159–165.
[2] *Сергієнко І.В., Хіміч О.М.* Математичне моделювання: від МЕЛМ до екзафлопсів. Вісник Національної академії наук України. 2019. № 8. С. 37–50.
[3] Amdahl G. Validity of the single-processor approach to achieving large-scale computer capabilities. AFIPS Conference Proceedings. 1967. 30. P. 483–485.
[4] *Arthur W.B., Ermoliev Yu.M., Kaniovski Yu.M.* Path-dependent processes and the emergence of macro-structure. European Journal of Operations Research. 1987. 30. P. 294–303.
[5] *Gustafson J.L.* Reevaluting Amdahl's law. Communications of the ACM. 1988. 31 (5). P. 532–533.

Forecasting the efficiency of multicomponent computer systems

Vasyl Gorbachuk, Maksym Dunaievskyi, Seit-Bekir Suleimanov

The advantages and disadvantages of Moore's, Hilder's, Amdal's, Gustafson-Barsis laws known in the field of information and communication technologies are shown, offering the necessary mathematical apparatus for constructing similar laws for predicting the efficiency of modern multicomponent computing systems. This apparatus includes both the parameters of the components of computer systems, and possible interdependencies between those parameters. In general, forecasting the efficiency of computer systems requires detailed documentation of the work of computer systems of the class given (the series given) on certain types of tasks with subsequent processing of the data obtained. The collection and processing of this data must take place and be recorded in the dynamics with assistance of appropriate smart sensors of the Internet of Things class.

Отримано 15.03.21