УДК 004.032.24

ВИКОРИСТАННЯ ЗАКОНІВ АМДАЛА ТА ГУСТАФСОНА ПРИ ОЦІНЮВАННІ ПОКАЗНИКА ПРИСКОРЕННЯ В БАГАТОПРОЦЕСОРНИХ СИСТЕМАХ

© Леонід Квурт, Любомир Цигилик, 2009

Національний університет "Львівська політехніка", кафедра електронних обчислювальних машин,
вул. С. Бандери, 12, 79013, Львів, Україна

Проаналізовано залежності продуктивності обчислювальних систем від кількості процесорів, здійснено порівняння значень показника прискорення відповідно до залежностей. Амдала та Густафсона, зроблено висновки щодо застосувань законів для різних типів обчислювальних систем.

Проанализированы зависимости производительности вычислительных систем от количества процессоров, осуществлено сравнение значений показателя ускорения в соответствии с зависимостями Амдала и Густафсона, сделаны выводы относительно применения законов для разных типов вычислительных систем.

The analysis of efficiency dependences of the computer systems is conducted from the amount of processors are presented. Comparison of index acceleration accordance with Amdahl's and Gustafson's lows are carried out. For applications of laws are made conclusions, for the different types of the computing systems.

Вступ. Одним із показників ефективності роботи обчислювальної системи вважається показник прискорення обчислень за рахунок паралелізму. Цей показник використовується для вибору кількості процесорів у багатопроцесорній системі [1], його значення впливає на процес вибору структури обчислювальної системи з можливостями реконфігурації, його можна використовувати при порівнянні ефективності обчислювальних систем різної складності.

Мета роботи. Мета роботи – дослідити значення показника прискорення виконання задачі за рахунок паралелізму обчислень. Дослідження виконати на основі закону Амдала та Густафсона. Проаналізувати отримані результати та зробити відповідні висновки.

Вирішення задачі. Відомо, що прискорення S – це відношення часу T_s, який необхідний для виконання обчислень на однопроцесорній обчислювальній системі (ОС) у варіанті найкращого послідовного алгоритму, до часу T_p розв'язання тієї самої задачі на паралельній системі [2] при використанні найкращого паралельного алгоритму, а саме:

$$S = \frac{T_s}{T_p} \qquad (1)$$

При визначенні значення відношення (1) можна користуватися законом Амдала або законом Густафсона [2].

Відповідно до закону Амдала:

$$S = \frac{n}{1+(n-1)\cdot f}, \qquad (2)$$

де n – кількість процесорів ОС, f – частка операцій, які повинні виконуватись послідовно одним із процесорів довільної ОС. При $f = 0$ – система повністю паралельна, а при $f = 1$ – система послідовна (рис. 1.).

Відповідно до закону Густафсона (рис. 2):

$$S = n + (1 - n) \cdot f, \qquad (3)$$

Як видно з [2], відмінності в залежностях (2) і (3) пояснюються тим, що постановка задачі Амдалом ґрунтувалась на тому, що об'єм задачі при зміні кількості процесорів, задіяних при розв'язанні задачі, залишається незмінним. Густафсон допускає, що користувач, зберігаючи незмінним час роботи ОС, намагається пропорційно до її потужності збільшити обсяг виконуваної роботи. І цей обсяг обчислень збільшується, переважно, за рахунок частини програми, що розпаралелюється.

Перед порівнянням результатів обчислень за залежностями (2) і (3) проаналізуємо фактори, які додатково з'являються при введенні паралелізму в ОС та обмежують показники прискорення.

Дослідження паралелізму завжди пов'язується із збільшенням накладних витрат на диспетчеризацію [3]. Відповідно збільшується додатковий обсяг обчислень, що зменшує показники прискорення.

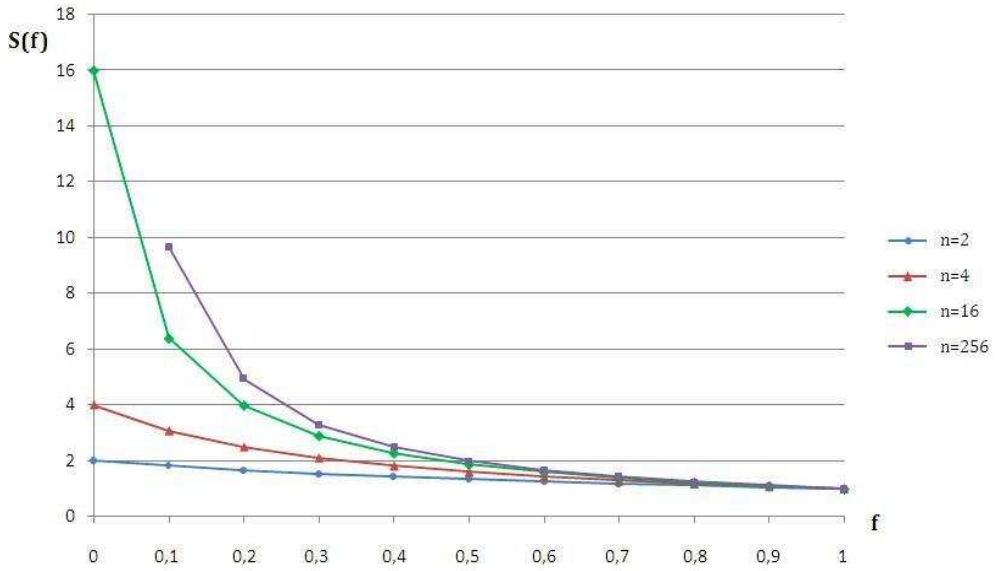

Рис. 1. Залежність між послідовною частиною алгоритму та рівнем прискорення за законом Амдала

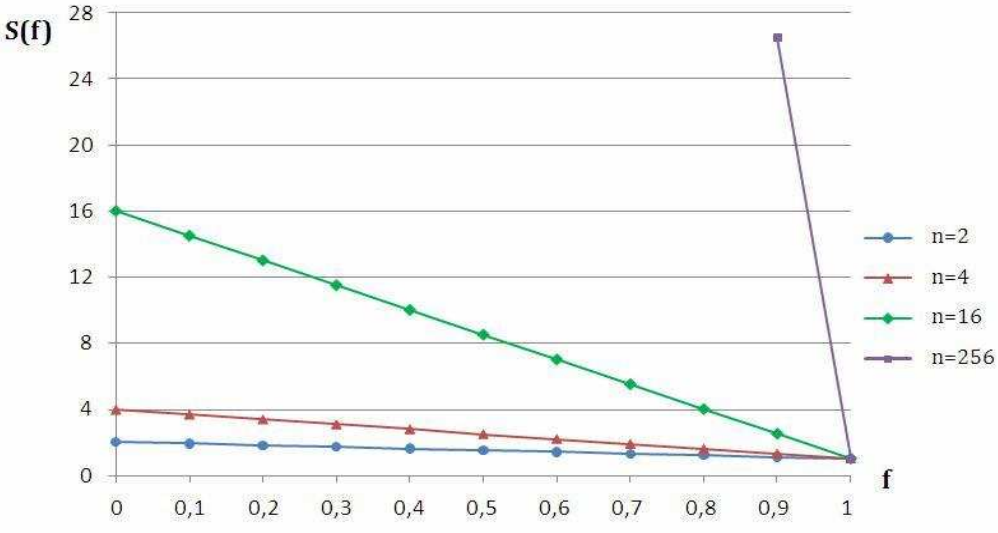

Рис. 2. Залежність між послідовною частиною алгоритму та рівнем прискорення за законом Густафсона

До факторів, які обмежують показники прискорення, належать [2,3]:

1) Програмні витрати:

a) при паралельних обчисленнях існують додаткові індексні обчислення, що пов'язуються із декомпозицією даних та розподілом їх за процесорами;

b) при паралельних обчисленнях існують різні види облікових операцій, яких немає у послідовних алгоритмах;

2) витрати через дисбаланс завантаженості процесорів. Між точками синхронізації кожен із процесорів повинен завантажуватись однаковим обсягом роботи (при однаковій їхній продуктивності), бо у іншому випадку частина процесорів буде в режимі очікування, поки інші завершать свої операції. Отже, прискорення ОС обмежується найповільнішим із процесорів;

3) комунікаційні витрати. Будь-які комутації між процесорами вимагають додаткового часу, а це прямо впливає на залежність (1) і зменшує прискорення. Для зменшення комутаційних витрат можна було б формувати достатньо великі обчислювальні блоки, та

межею такої стратегії буде повернення до послідовної системи.

Аналіз наведених факторів показує, що програмні витрати, витрати на диспетчеризацію можна віднести до послідовної частини алгоритму розв'язання задачі, витрати через дисбаланс завантаженості процесорів, комунікаційні витрати – до паралельної частини алгоритму.

Висновки

1. Розрахунки показують, що значення прискорень обчислень при використанні паралелізму за залежностями (1) та (2) мають істотні відмінності (рис. 1, 2), які збільшуються із збільшенням в системі кількості процесорів, що працюють паралельно.

2. Коментарі до закону Амдала про те, що обсяг задачі при зміні кількості процесорів залишається незмінним [2] – некоректні. Фактично залежність Амдала (2) враховує однакову пропорцію збільшення як послідовної, так і паралельної складової, що зумовлює (і саме про це засвідчує) стале значення параметра *f*.

3. Для визначення прискорення обчислень за рахунок застосування паралелізму перевагу потрібно надавати залежності (2) Амдала, тому що порівняно із законом Густафсона (3) ця залежність критичніша до процесів в ОС, пов'язаних із нарощуванням кількості процесорів.

1. Мельник А. О. Архітектура комп'ютера. Наукове видання. – Луцьк: Волинська обласна друкарня, 2008. – 470 с. 2. Цилькар Б. Я., Орлов С. А. Организация ЭВМ и систем: Учебник для вузов. – СПб.: Питер, 2004. – 668 с. 3. Тербер К. Дж. Архитектура высокопроизводительных вычислительных систем. – М.: Наука, 1985. – 272 с.

УДК 681.3

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ ТА РЕАЛІЗАЦІЇ МЕХАНІЗМУ ПРИХОВАНОГО ЗБИРАННЯ ІНФОРМАЦІЇ З СИСТЕМ ВІРТУАЛЬНИХ ПРИМАНОК

© Назар Тимошик, Роман Тимошик, 2009

Національний університет "Львівська політехніка", кафедра захисту інформації,
вул. С. Бандери, 12, 79013, Львів, Україна

Наведено результати досліджень та реалізації механізму інтроспекційного аналізу системи-приманки, який забезпечує стійкий до виявлення та блокування моніторинг діяльності зловмисника в операційній системі Linux. Детально розглянуто труднощі реалізації та особливості архітектурної реалізації. На основі реалізованого програмного забезпечення уможливлюється повний та ефективний контроль подій у віртуалізованій ОС.

Приведено результаты исследований и реализации механизма интроспекционного анализа системы-приманки, который обеспечивает стойкий к выявленню и блокированию мониторинг деятельности злоумышленника в операционной системе Linux. Детально рассмотрены трудности реализации и особенности архитектурной реализации. На основании реализованного программного обеспечения стает возможным полный и эффективный контроль событий в виртуализированной ОС.

The results of researches and realization of mechanism of introspection analysis of the honeypot systems, which provides proof to the exposure and blocking monitoring of activity of malefactor in the operating system of Linux, are resulted in the article. Difficulties of realization, and features of architectural realization, are considered in detail. On the basis of the realized software complete and effective control of events becomes possible in vitrualized OS.

Особливості інтроспекційного аналізу. У роботах [1, 2] описувались важливість та актуальність новітніх стійких до блокування зловмисником механізмів збору інформації з систем-приманок, недоліки відомих нині рішень та був виконаний огляд можливих способів вирішення цієї проблеми. У цій статті пропонуються результати досліджень та реалізації механізму Virtual Machine Introspection (VMI) [3], завданням якого є