

## Метод підвищення продуктивності для неоднорідних кластерних систем

Т. В. Дрегалю, В. П. Симоненку, Л. В. Дрегалю, О. Р. Педоренку, М. С. Соловійову

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна  
Corresponding author. E-mail: TarasDregalo@gmail.com

Paper received 22.06.17; Accepted for publication 28.06.17.

**Анотація.** Дана стаття присвячена кластерним системам. Автори статті розглянули загальну архітектуру кластерних систем. Вони пояснили базові поняття та парадигми, що пов'язані з комп'ютерними кластерними системами. Автори запропонували метод оцінки вузлів у неоднорідних кластерних системах. Даний метод базується на двох задачах: proof of work та proof of trust. Перша задача складається з двох етапів. Першим етапом задачі є отримання конфігурацій програмного та апаратного забезпечення кожного вузла в системі. Усі вузли відправляють свої конфігурації центральному управляючому обчислювальному вузлу (ЦУОВ), котрий робив запит. Далі на основі отриманих результатів ЦУОВ приймає рішення, буде вузол займатись обчисленням задачі чи ні. ЦУОВ відправляє типову задачу вузлам, котрі мають необхідну конфігурацію. З отриманих результатів формується рейтингова таблиця вузлів.

**Ключові слова:** комп'ютерний кластер, вузол, proof of work, proof of trust, закон Амдала

**Вступ.** Для вирішення задач підвищеної складності зазвичай використовують обчислювальні системи. Обчислювальна система, або кластер – це система, котра складається з  $n$ -ої кількості обчислювальних машин, які функціонують як одна система для вирішення певних задач [2].

Зазвичай виділяють декілька типів кластерних систем:

- обчислювальні кластерні системи;
- розподілені обчислювальні системи;
- системи високої доступності;
- системи з розподіленим навантаженням.

### Короткий огляд публікацій по темі

Основним елементом кожної обчислювальної системи є обчислювальний вузол (ОВ). У якості вузлів можна використовувати персональні комп'ютери та робочі станції [2]. У кластерних системах виділяють два типи вузлів: ЦУОВ та ОВ. Центральні управляючі обчислювальні вузли – це вузли, котрі назначають задачі на обчислювальні вузли та управляють ними, а обчислювальні вузли виконують команди, які приходять від ЦУОВ. Обчислювальні вузли діляться на декілька типів в залежності від конфігурацій як апаратного так і програмного забезпечення. За архітектурою вузлів кластерні системи діляться на два основні типи:

- однорідні – де всі вузли мають однакове апаратне та програмне забезпечення, завдяки цьому всі вузли мають однакову продуктивність;
- неоднорідні – де всі вузли різної мають різне

апаратне та програмне забезпечення, а отже мають і різну продуктивність.

Зазвичай всі кластерні системи на етапі проектування мають однорідну архітектуру, але коли постає питання про збільшення обчислювальної потужності додаються вузли іншої архітектури. Таким чином, кластерна система перетворюється на систему з неоднорідними вузлами. Варто зазначити, що іноді кластерні системи складаються з декількох обчислювальних систем, котрі локально розташовані в різних частинах світу.

**Ціль та постановка задачі.** За різних конфігурацій різні вузли мають різну продуктивність, тому для вирішення однієї задачі різні вузли будуть витратити різний час. За законом Амдала [3], якщо задача розділяється на декілька частин, сумарний час її виконання в паралельній системі не може бути менше часу виконання найдовшого фрагмента. За даним законом, продуктивність кожного вузла впливає на продуктивність цілої системи. Кластерні системи з неоднорідними вузлами, а саме система розподілення задач по вузлах - потребують подальшого удосконалення.

**Задача "proof of work".** У ЦУОВ є дві бази даних, перша з яких – це база типових (tasks\_db) задач та відповідей. Типові задачі задаються операторами системи. ЦУОВ обчислює типу задачу декілька разів і після цього записує в базу даних цю задачу та її результат. Іншою базою є база даних вузлів (nodes\_db), де зберігається інформація про кожний вузол системи. Архітектура системи показана на Рисунку 1.

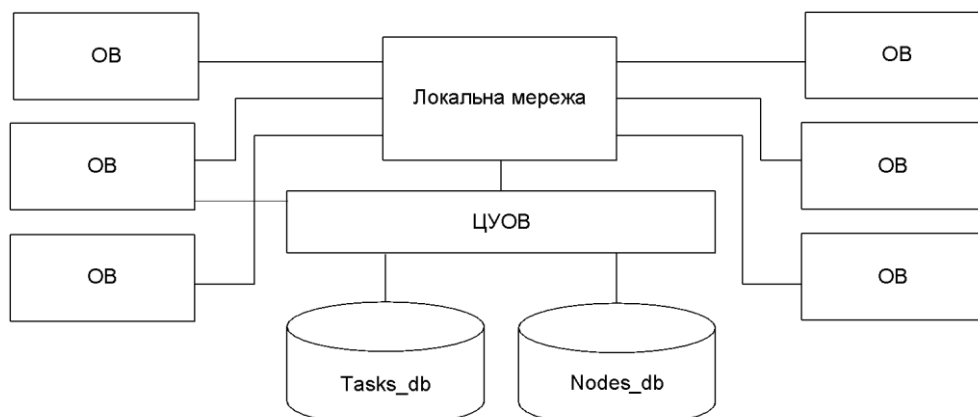


Рисунок 1. Архітектура системи.

Перш за все, кожний обчислювальний вузол повинен підтверджувати свою надійність та продуктивність. Для цього кожному вузлу системи необхідно засвідчити свою працездатність. Для цього ЦУОВ потрібно сформулювати критерії для вузлів, в якості яких зазвичай використовують конфігурацію програмного та апаратного забезпечення. Але цього все ж недостатньо, необхідний критерій продуктивності - найпростішим способом є відправка типової задачі вузлу та отримання результатів задачі. Дану задачу можна назвати *proof of work*, котра складається з двох етапів:

- етап ініціалізації (initiation step);
- етап обчислення (calculation step).

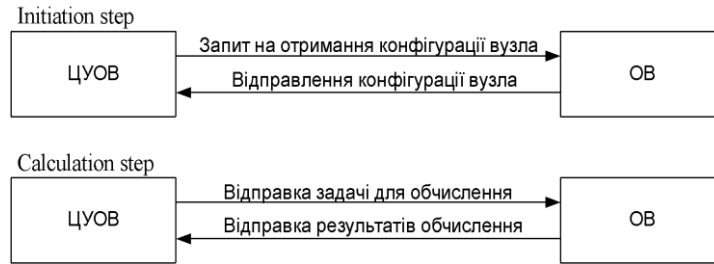


Рисунок 2. Алгоритм роботи задачі *proof of work*.

Кожний вузол в кластерній системі повинен виконати задачу *proof of work*. Якщо це не зробив, на нього не може бути назначена обчислювальна задача. Крім того, задача *proof of work* повинна повторюватись з певною періодичністю або ж у випадку, коли необхідно змінити критерії вузлів, котрі необхідні для системи. Дана задача виконується також у випадку, коли додається нова типова задача.

**Задача *proof of trust*.** Після отримання інформації про вузол можна обчислити надійність кожного вузла ( $P_n$ ) в залежності від апаратного забезпечення. У кожного вузла є свій запис в базі даних *nodes\_db*, який має наступні поля:

- назва вузла;
- конфігурація вузла;
- надійність вузла, що вираховується на основі задачі *proof of work*;
- версія типової задачі;
- час обрахунку типової задачі;
- результат типової задачі;
- загальний час обрахунку;
- актуальний час обрахунку;
- загальна кількість задач;
- актуальна кількість задач.

Поле «версія типової задачі» повинно співпадати з версією в базі даних *tasks\_db*. Якщо вона не співпадає, це означає, що вузол не виконував задачу *proof of work* і він не може використовуватись в задачі *proof of trust*. Після перевірки версії відбувається перевірка результату роботи вузла, який повинен співпадати з результатом, котрий зберігається в базі даних *tasks\_db*.

Кожного разу коли приходиться обчислювальна задача, ЦУОВ розраховує, скільки часу ( $T_{cal}$ ) необхідно для вузла щоб обчислити його частину задачі. Кожного разу коли вузлу надають обчислювальну задачу його загальний час ( $T_{total}$ ) збільшується на  $T_{cal}$  та вираховується, скільки часу ОВ витратив на підрахунки

На етапі ініціалізації ЦУОВ відправляє запит на отримання конфігурації від кожного вузла, а ОВ відправляє відповідь з своєю конфігурацією. Далі ЦУОВ перевіряє критерії апаратного та програмного забезпечення вузла та записує їх в базу даних вузлів. І приймає рішення передавати вузол на наступний етап чи ні.

На етапі обчислення ЦУОВ відправляє типову задачу обчислювальному вузлу, ЦУОВ фіксує час відправлення задачі та отримання результату. ОВ обчислює типову задачу та відправляє ЦУОВ. А ЦУОВ перевіряє результат та записує його в базу вузлів. Алгоритм роботи задачі *proof of work* зображений на рисунку 2.

( $T_{run}$ ), а  $N_{total}$  збільшують на одиницю. Якщо вузол виконав поставлену задачу до його актуального часу ( $T_{ex}$ ), до загального часу ( $T_{total}$ ) додають  $T_{run}$ , і  $N_{ex}$  збільшується на одиницю, після чого необхідно від загального часу відняти ( $T_{cal}$ ). Якщо вузол новий, всі наступні поля ( $N_{total}$ ,  $N_{ex}$ ,  $T_{total}$ ,  $T_{ex}$ ) встановлені за замовчуванням як одиниці.

Загальна кількість задач ( $N_{total}$ ) – це кількість задач, котру вузол повинен був виконати. Актуальна кількість задач ( $N_{ex}$ ) - це кількість задач, котру вузол виконав. На основі даної інформації можливо порахувати надійність вузла кластерної системи або “*proof of trust*”, за формулою  $P_{pot} = P_n \frac{T_{ex}}{T_{total}} \frac{N_{ex}}{N_{total}}$  (1), де  $P_n$  - апаратна надійність вузла (від 0 до 1).

На основі *proof of trust* можна сформувати таку рейтингову таблицю вузлів, що чим ближче рейтинг вузла до одиниці тим більш надійним він є.

**Вибірка вузлів для вирішення поставленої задачі.**

З отриманих результатів із попередньої задачі ми отримаємо таблицю вузлів. Якщо вузлів не вистачає для обчислення поставленої задачі, тоді необхідно виконати задачу *proof of work*, але змінити критерії конфігурації. Якщо ж вузлів більше ніж потрібно, тоді необхідно виконати середнє арифметичне відхилення за  $P_{pot}$ , для чого необхідно скористатись формулою  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$  (2), де  $x_i$  - це  $P_{pot}$  для  $i$ -го вузла,  $n$  - загальна кількість вузлів в таблиці, а  $\bar{x}$  - середнє арифметичне.

Далі вирахувати стандартне відхилення за формулою  $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2}$  (3) [1]. Використовуючи правило 3х сигм, котре стверджує, що практично всі значення нормально розподіленої випадкової величини лежать в діапазоні  $\{\bar{x} - 3\sigma; \bar{x} + 3\sigma\}$  [1]. Усі наші вузли також знаходяться в даному діапазоні. Отже, необхідно визначити коефіцієнт  $k$ , котрий знаходиться в діа-

пазоні  $\{0.1; 3\}$  і підставити його в діапазон  $\{\bar{x} - k\sigma; \bar{x} + k\sigma\}$ . Цей коефіцієнт необхідно збільшувати з кроком в 0.1 до тих пір, поки кількість вузлів, що знаходиться в заданому діапазоні, не буде більше ніж кількість вузлів, які необхідні для обчислення задачі.

**Висновки.** Рейтингова таблиця формується з результатів двох задач proof of work та proof of trust. Основа мета цих задач - відсіяти ненадійні вузли або вузли з малою продуктивністю, що в свою чергу збільшить продуктивність неоднорідної кластерної сис-

теми. Використання даного методу зменшить час на обчислення, збільшить швидкість та надійність вирішення задачі. На основі даного методу всі вузли з однаковою продуктивністю будуть зайняті обчисленням задачі. Це значно зменшить час простою систем, що, в свою чергу, є однією з основних проблем в неоднорідних кластерних системах. Проте варто зазначити, що системі потрібен час та ресурси для виконання задач proof of work та proof of trust.

#### ЛИТЕРАТУРА

1. Боровиков В. STATISTICA. Искусство анализа данных на компьютере: Для профессионалов / В. Боровиков. — СПб.: Питер, 2003. — 688 с.
2. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стеен.. – Санкт-Петербург: Питер, 2003. – 877 с.
3. Amdahl, G. (April 1967) «The validity of the single processor approach to achieving large-scale computing capabilities». In Proceedings of AFIPS Spring Joint Computer Conference, Atlantic City, N.J., AFIPS Press, pp. 483-85.
4. Laurie B. "Proof of Work" Proves Not to Work [Электронный ресурс] / B. Laurie, R. Clayton // ALD Ltd, The Stores, 2 Bath Road, London W41LT, United Kingdom 2 University of Cambridge, Computer Laboratory , William Gates Building, 15 JJ Thompson Avenue, Cambridge CB30FD, United Kingdom – Режим доступа до ресурсу: <https://www.cl.cam.ac.uk/~rnc1/proofwork.pdf>.

#### REFERENCES

1. Borovikov V. STATISTICA. The art of data analysis on a computer: For professionals / V. Borovikov. - St. Petersburg: Peter, 2003. - 688 p.
2. Steen, Maarten van; Tanenbaum, Andrew S. (2007). Distributed systems: principles and paradigms. Upper Saddle River, NJ: Pearson Prentice Hall. ISBN 0-13-239227-5
3. Amdahl, G. (April 1967) «The validity of the single processor approach to achieving large-scale computing capabilities». In Proceedings of AFIPS Spring Joint Computer Conference, Atlantic City, N.J., AFIPS Press, pp. 483-85.
4. Laurie B. "Proof of Work" Proves Not to Work / B. Laurie, R. Clayton // ALD Ltd, The Stores, 2 Bath Road, London W41LT, United Kingdom 2 University of Cambridge, Computer Laboratory , William Gates Building, 15 JJ Thompson Avenue, Cambridge CB30FD, United Kingdom

#### A method for increasing productivity in heterogeneous cluster systems

T. V. Dregalo, V. P. Simonenko, L. V. Dregalo, O. R. Pedorenko, M. S. Soloviova

**Abstract.** This paper deals with cluster systems. Authors have paid much attention to architecture of these systems. They have also explained all laws and paradigms that are related to computer cluster systems. The authors propose a new method rating of nodes in inhomogeneous cluster systems. This method based on two tasks: proof of work and proof of trust. Task proof of work consists of two steps, on the first server sends requests for all nodes and takes information about software and hardware. After this server makes the decision if server can use this node for calculation or not. If all is ok, the server starts step two and sends simple task for calculation. Using a result of the task server makes rating table with nodes, which can be used for calculation.

**Keywords:** Computer cluster, node, proof of work, proof of trust, Amdahl's law.

#### Метод повышения продуктивности для неоднородных кластерных систем

Т. В. Дрегало, В. П. Симоненко, Л. В. Дрегало, О. Р. Педоренко, М. С. Соловьева

**Аннотация.** В статье рассматриваются кластерные системы. Авторы уделяли большое внимание архитектуре этих систем. Они также объяснили все законы и парадигмы, связанные с компьютерными кластерными системами. Авторы предложили новый метод оценки узлов в неоднородных кластерных системах. Этот метод основан на двух задачах: proof of work и proof of trust. Задача proof of work состоит из двух этапов: на первом этапе сервер отправляет запросы на все узлы и получает информацию о программном и аппаратном обеспечении. После этого сервер принимает решение, можно ли использовать этот узел для расчета или нет. Если все в порядке, сервер начинает второй этап и отправляет простую задачу для расчета. На основе полученных результатов сервер формирует таблицу с рейтингом всех узлов в системе, которые могут использовать для расчета.

**Ключевые слова:** Компьютерный кластер, вузол, proof of work, proof of trust, закон Амдала.