УДК 004.75

МЕТОДИ УПРАВЛІННЯ РЕСУРСАМИ ВЕЛИКИХ ДАНИХ В РОЗПОДІЛЕНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ

Борис Т.В., Алєксєєв М.О.
Інститут телекомунікаційних систем НТУУ «КПІ», Україна
E-mail: boris.tatyana.ua@gmail.com; alexeyev@its.kpi.ua

Methods of big data resource management in distributed computing systems
This paper analyzed the basic methods of of Big Data resource management. To perform the experiment was deployed and configured test environment in Amazon Compute Cloud.

В умовах постійного зростання інформації оптимізація її обробки є першочерговою задачею. З'явилися технічні можливості для вирішення масштабних завдань в галузі науки, техніки і комерції на територіально розподілених ресурсах, що належать різним власникам та є гетерогенними. З іншого боку, майже не існує комплексних досліджень методів управління такими ресурсами.

Проблема управління ресурсами Великих Даних в розподілених обчислювальних системах за допомогою технологій MapReduce та альтернативних підходів являється актуальною. Управління ресурсами в традиційних гомогенних розподілених системах - добре вивчене і опрацьоване питання. Існує велика кількість менеджерів ресурсів для подібних систем [1]. Менеджери даного типу реалізують механізми і політики для ефективного використання даних, мають повний контроль, але тільки над такими ізольованими ресурсами. Однак, алгоритми управління для ізольованих гомогенних розподілених систем не будуть ефективними в гетерогенних обчислювальних середовищах.

Розглянемо управління великими масивами даних в хмарі на прикладі використання Amazon Web Services (AWS). Дану платформу відносить до класу IaaS-рішень так як вона надає широкий спектр хмарних сервісів. Рішення, побудовані на її основі можна розглядати як розподілені гетерогенні ресурси.

Amazon Elastic MapReduce забезпечує розміщення в рамках Hadoop, що працює на глобальних веб-інфраструктурах Amazon Elastic Compute Cloud (Amazon EC2) і дозволяє створювати користувацькі JobFlows.

Проблемно-орієнтована специфіка потоків робіт в тому, що в переважній більшості випадків, ще до виконання завдання, для кожного завдання можуть бути отримані оцінки таких якісних характеристик, як час виконання завдання, межі масштабованості і обсяг даних, що генеруються. Обчислювальні завдання по обробці ресурсів Великих Даних в багатьох випадках мають потокову структуру і можуть бути описані за допомогою моделі потоку робіт (workflow) [2], відповідно до якої завдання представляється у вигляді орієнтованого ациклічного графа. Вузлами графа є завдання, які є складовими частинами завдання, а дуги відповідають потокам даних, переданих між окремими завданнями (Рис.1). При цьому набір завдань, з яких будуються завдання, є

кінцевим і визначеним. Використання подібних знань про специфіку завдань в конкретній проблемно-орієнтованої області може істотно поліпшити ефективність методів управління обчислювальними ресурсами.

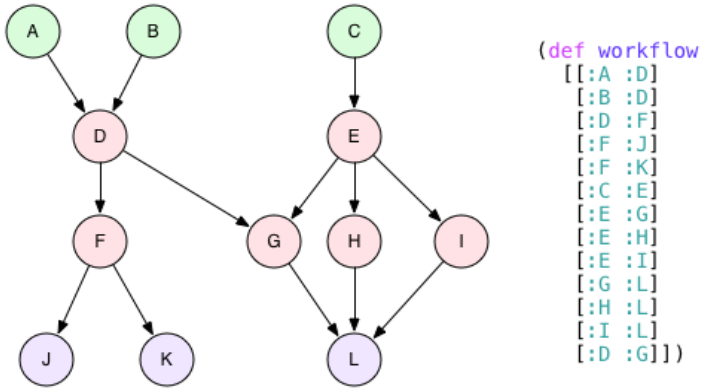

Рис. 1 Приклад орієнтованого ациклічного графа. Входи A, B і C. Виходи J, K і L. Всі інші вузли є функціями

В даний час відомо кілька програмних систем, орієнтованих на управління складними додатками з потокової структурою в розподілених обчислювальних середовищах. Потокова структура породжує гетерогенність ресурсів. Управління даними подібних ресурсів призвело до створення алгоритмів по аналізу асоціативних правил відносно специфіки завдань в конкретній проблемно-орієнтованій області [3]. Ранні роботи в галузі управління даними в розподілених середовищах фокусуються на пошуку асоціативних правил в гомогенних ресурсах. Однак для виконання експериментального дослідження застосуємо даний клас алгоритмів на гетерогенних ресурсах. До даного класу можна віднести як алгоритми Apriori, Eclat, FP-growth.

Для виконання експерименту було сконфігуровано Ubuntu Server в якості тестового середовища в Amazon EC2. В ході експерименту була поставлена тестування методів управління оцінюючи пошук асоціативних правил в за допомогою алгоритму Apriory. Перевага цього алгоритму в тому, що після того, як кожна транзакція з вихідного набору даних «пройде» через дерево, можна перевірити чи задовольняють значення підтримки кандидатів мінімального порогу.

Алгоритм Apriori працює в два етапи: на першому кроці необхідно знайти як часто зустрічаються набори елементів, а потім, на другому, отримати з них правила. Кількість елементів у наборі будемо називати розміром набору, а набір, що складається з k елементів, k-елементних набором. На першому кроці алгоритму підраховуються 1-елементні набори,що часто зустрічаються. Для цього необхідно пройтися по всьому набору даних і підрахувати для них підтримку, тобто скільки разів зустрічається в базі. Наступні кроки будуть складатися з двох частин: генерації потенційно часто зустрічаються наборів елементів (їх називають кандидатами) і підрахунку підтримки для кандидатів. Хеш-дерево будується щоразу, коли формуються кандидати. Спочатку дерево

складається тільки з кореня, який є листом, і не містить ніяких кандидатів-наборів.

Проведений експеримент показав, що швидкість обробки асоціативних правил знижується з зростанням об'єму даних (Рис.2). Таким чином зі зростанням об'єму даних знизиться відсоток корисного використання дискової пам'яті і обчислювальних потужностей і сповільниться пошук та видобування інформації для кінцевого користувача, що є сильним недоліком.

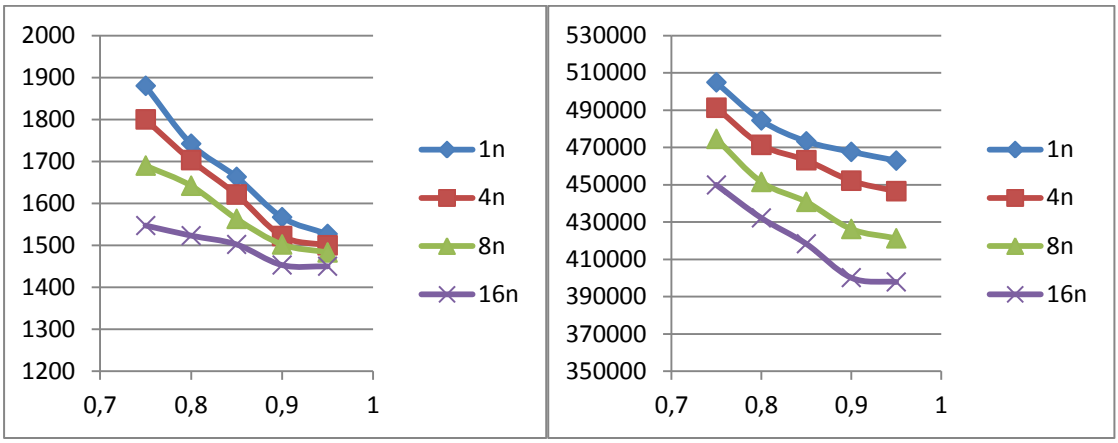

Рис. 2 Пошук асоціативних правил в хмарі за допомогою алгоритму Apriory(1 Gb, 300 Gb)

На основі проведеного експериментального тестування методів управління ресурсами було підтверджено, що алгоритми управління для однорідних розподілених обчислювальних систем погано адаптуються для розподілених гетерогенних систем [4], які генерують Великі Дані. Управління ресурсами в неоднорідних розподілених обчислювальних середовищах вимагає принципово нових моделей обчислень і управління ресурсами.

Висновки

В даній роботі окреслені перспективні напрямки досліджень роботи з великими даними. Аналіз існуючих методик на основі наведених вище графічних результатів експериментів підтверджує необхідність адаптації алгоритмів управління ресурсами Великих Даних в розподілених обчислювальних системах.

Література

1. Distributed resource management for high throughput computing [Електронний ресурс]/ Режим доступа: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=709966, вільний доступ.
2. Research on Component and DAG Based Dynamic Workflow System Chuan-Sheng Zhou ; Li-Hua Niu ; Jie Liu (PCSPA), 2010.
3. Association_rule_learning [Електронний ресурс] /Режим доступа: http://en.wikipedia.org/wiki/Association_rule_learning#Algorithms, вільний доступ.
4. Adaptive distributed algorithms for distributed computing systems Yichuan Hu ; Ribeiro, A. Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference.