

10. LanguageTool Forum. URL: <https://forum.languagetool.org> (дата звернення 11.11.2021)
11. Grammarly Kyiv @GrammarlyKyiv IT-компания. URL: <https://www.facebook.com/GrammarlyKyiv/> (дата звернення 11.11.2021)
12. Український правопис (2019). URL: <https://mon.gov.ua/ua/osvita/zagalna-serednya-osvita/navchalni-programi/ukrayinskij-pravopis-2019> (дата звернення 11.11.2021)
13. Apache OpenOffice - Official Site - The Free and Open Productivity Suite. URL: <https://www.openoffice.org/> (дата звернення 10.10.2021)
14. LibreOffice - Free Office Suite - Based on OpenOffice - Compatible with Microsoft. URL: <https://www.libreoffice.org/> (дата звернення 10.10.2021)
15. Language Accessory Pack для Office. URL: <https://support.microsoft.com/ru-ru/office/language-accessory-pack-%D0%B4%D0%BB%D1%8F-office-82ee1236-0f9a-45ee-9c72-05b026ee809f> (дата звернення 10.10.2021)
16. Ukrainian Dictionary. URL: <https://extensions.openoffice.org/en/project/ukrainian-dictionary> (дата звернення 10.10.2021)
17. Установка словарей в LibreOffice. URL: <https://openlearn.ru/forum/openlearn-1/question/ustanovka-slovarej-v-libreoffice-51> (дата звернення 10.10.2021)

References

- The fortunes of the two Grammarly founders exceeded \$ 4 billion each. URL: <https://ain.ua/2021/11/23/sostoyanie-osnovatelej-grammarly/> (access date 11.11.2021)
- Ukrainians - the founders of Grammarly - have become dollar billionaires richer than Akhmetov. URL: https://radiotrek.rv.ua/news/ukrayinci---zasnovniki-grammarly---stali-dolarovimi-milyarderami-bagatshimi-za-ahmetova-video_279801.html (accessed 11.11.2021)
- History of Grammarly: how three people of Kiev created a spell check service. URL: <https://mc.today/istoriya-grammarly-kak-troe-kievlyan-sozdali-servis-proverki-pravopisaniya-stoimostyu-1-mld> (accessed 11.11.2021)
- LanguageTool vs Grammarly - Which Is Better? . URL: <https://rigorousthemes.com/blog/languagetool-vs-grammarly/> (accessed 11/11/2021)
- LanguageTool Vs. Grammarly - Which Grammar Checker Is Worth Your Time And Money? URL: <https://becomeawritertoday.com/languagetool-vs-grammarly/> (accessed 11/11/2021)
- Grammarly: Free Online Writing Assistant. URL: <https://www.grammarly.com> (accessed 11.11.2021)
- LanguageTool - Online Grammar, Style & Spell Checker. URL: <https://languagetool.org> (accessed 10.11.2021)
- Grammarly. From Wikipedia, the free encyclopedia. URL: <https://en.wikipedia.org/wiki/Grammarly> (accessed 11/11/2021)
- LanguageTool Wiki. URL: <http://wiki.languagetool.org/> (accessed 11/11/2021)
- LanguageTool Forum. URL: <https://forum.languagetool.org> (accessed 11/11/2021)
- Grammarly Kyiv @GrammarlyKyiv IT company. URL: <https://www.facebook.com/GrammarlyKyiv/> (access date 11/11/2021)
- Ukrainian spelling (2019). URL: <https://mon.gov.ua/en/osvita/zagalna-serednya-osvita/navchalni-programi/ukrayinskij-pravopis-2019> (access date 11.11.2021)
- Apache OpenOffice - Official Site - The Free and Open Productivity Suite. URL: <https://www.openoffice.org/> (accessed 10.10.2021)
- LibreOffice - Free Office Suite - Based on OpenOffice - Compatible with Microsoft. URL: <https://www.libreoffice.org/> (accessed 10.10.2021)
- Language Accessory Pack for Office. URL: <https://support.microsoft.com/en-us/office/language-accessory-pack-%D0%B4%D0%BB%D1%8F-office-82ee1236-0f9a-45ee-9c72-05b026ee809f> (date appeal 10.10.2021)
- Ukrainian Dictionary. URL: <https://extensions.openoffice.org/en/project/ukrainian-dictionary> (accessed 10.10.2021)
- Installing dictionaries in LibreOffice. URL: <https://openlearn.ru/forum/openlearn-1/question/ustanovka-slovarej-v-libreoffice-51> (access date 10.10.2021)

УДК 004.052
DOI : 10.53920/ITS-2021-1-4

Юрій ПОНОЧОВНИЙ

Полтавський державний аграрний університет
ORCID ID: 0000-0002-6856-2013
e-mail: yuriy.ponch@gmail.com

Олег ПРЯДА

ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая»
e-mail: oleg_pryada@ukr.net

Юрій СОРОКА

ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая»
e-mail: yuriy_soroka@ukr.net

Юрій ДИКУН

ЗВО «Міжнародний науково-технічний університет імені академіка Юрія Бугая»
e-mail: yuriy_dikun@ukr.net

МОДЕЛЬ ПУЛУ СЕРВЕРІВ ДЛЯ ОЦІНЮВАННЯ ЕНЕРГОСПОЖИВАННЯ ПРИ ОБРОБЦІ ВЕЛИКИХ ДАНИХ

Розглянуто платформи для організації систем обробки великих даних. Деталізовано питання розгортання, використання, складу архітектури та можливостей Apache Spark в хмарі Azure. Розглянуто компоненти кластера Apache Spark в Azure HDInsight. Виділено види диспетчерів кластерів Apache Mesos, Apache Hadoop YARN і Spark. Наведено загальну модель обслуговування завдання у Spark кластері, що дозволяє оцінити ймовірність відмови завдання, серверну складову часу затримки до відгуку SparkContext, показники енергоспоживання компонент архітектури. В цій моделі розглядається три типи груп ресурсів: гарячого (hot), теплого (warm) і холодного (cold) пулів фізичних серверів. Побудована стохастична модель фізичного сервера гарячого пулу у вигляді марковського графа. Наведені формули для розрахунку загального середнього енергоспоживання фізичного сервера.

Ключові слова: великі дані, кластер, пул серверів, марковська модель, енергоспоживання.

Yuriy PONOCHOVNIY

Poltava State Agriculture University

Oleg PRYADA,

Yuriy SOROKA,

Yuriy DIKUN

Higher Educational Institution "Academician Yuriy Bugay
International Scientific and technical university"

MODEL OF SERVER POOL FOR ESTIMATION OF ENERGY CONSUMPTION IN BIG DATA PROCESSING

Platforms for the organization of big data processing systems are considered. Details of the deployment, use, architecture, and capabilities of Apache Spark in the Azure cloud are detailed. The components of the Apache Spark cluster in Azure HDInsight are considered. The types of cluster managers Apache Mesos, Apache Hadoop YARN and Spark are distinguished. The general model of task maintenance in the Spark cluster is given, which allows to estimate the probability of task failure, the server component of the delay time to the response of

SparkContext, the energy consumption of the architecture components. This model considers three types of resource groups: hot, warm, and cold pools of physical servers. A stochastic model of a physical hot pool server in the form of a Markov graph is constructed. The formulas for calculating the total average power consumption of a physical server are given.

Key words: *big data, cluster, server pool, Markov model, power consumption.*

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

На ринку сьогодні існує багато платформ для організації систем обробки великих даних. Існують як пропріетарні (комерційні), так і відкриті (вільні). На основі відкритих платформ, таких як Apache Hadoop, Apache Spark та ін. багато компаній створюють свої інфраструктури та пропонують засоби для їх управління, зокрема, надають комплекси для перетворення наявних ресурсів в хмарні OpenStack, CloudFoundry, Azure.

Для того, щоб вибрати найбільш підходящу платформу і хмарного провайдера необхідно чітко сформулювати вимоги, що пред'являються до хмари, а також провести пробне тестування всіх можливих платформ. Найчастіше це найкращий спосіб зрозуміти, чи підходить рішення або необхідно пробувати створювати своє на основі відкритих платформ.

Аналіз останніх досліджень і публікацій

Питання моделювання кластерних ресурсів були розглянуті в [1,2], аналітичне моделювання ресурсів обробки великих даних надано в [3]. В дослідженнях [4,5] розглянуті імітаційні моделі кластерних обчислень.

Виділення не вирішених раніше частин загальної проблеми

Метою даної статті є розробка марковської моделі компоненти Spark кластера – гарячого пулу фізичних серверів для оцінки їх енергоспоживання.

Виклад основного матеріалу

Apache Spark – це платформа паралельної обробки, яка підтримує роботу в пам'яті, щоб підвищити продуктивність додатків для аналізу великих даних. Однією із кількох пропозицій Spark в Azure (це хмарна реалізація Apache Spark від Майкрософт) є Apache Spark в Azure HDInsight [6].

Кластери Spark в HDInsight сумісні з BLOB-об'єктами Azure, Azure Data Lake Storage першого покоління або Azure Data Lake Storage 2-го покоління, що дозволяє застосовувати роботу Spark до існуючих сховищ даних.

Spark пропонує примітиви для кластерних обчислень у пам'яті. Завдання Spark може завантажувати дані, розмішувати їх у кеш у пам'яті та багаторазово їх запитувати [7]. Обчислення в пам'яті виконується на багато швидше, ніж у додатках, що користуються дисками (наприклад, додаток Hadoop, який керується доступом через розподілену файлову систему Hadoop (HDFS)). Spark також інтегрується в мову програмування Scala, що дає можливість керувати розподіленими наборами даних, такими як локальні колекції. Немає необхідності структурувати обмін даними через операції порівняння та редукації.

Кластери Apache Spark в HDInsight включають наступні компоненти:

- Ядро Spark. Включає ядро Spark, Spark SQL, потокові API-інтерфейси Spark, GraphX і MLlib.
- Anaconda
- Apache Livy

- Записна книжка Jupyter
 - Записна книжка Apache Zeppelin
- Кластери Spark в HDInsight включають драйвер ODBC для підключень засобів бізнес-аналітики, таких як Microsoft Power BI [6].

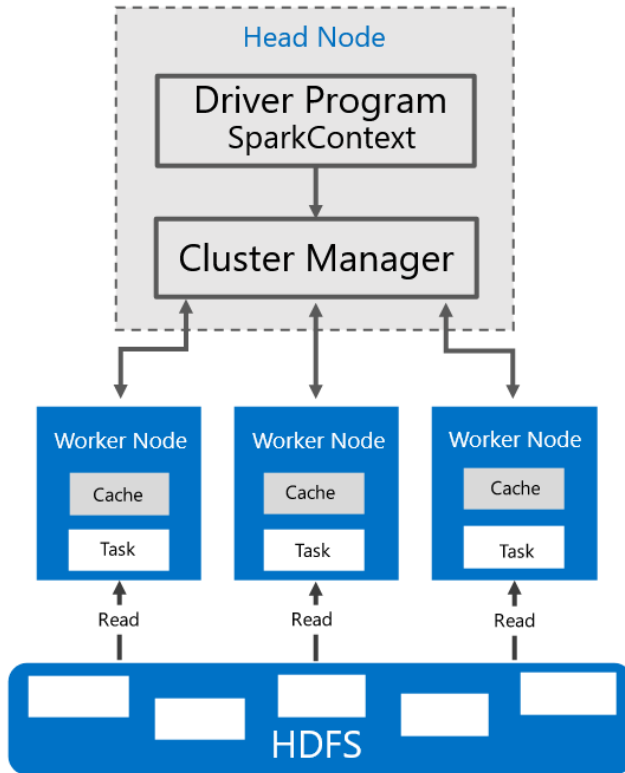


Рис. 1. Архітектура кластера Spark [6]

Завдання Spark виконуються як незалежні набори процесів у кластері. Вони координуються об'єктом SparkContext в основній програмі, що називається програмою драйвера).

SparkContext може підключатися до різних типів диспетчерів кластерів, які розділяють ресурси між додатками. До цих диспетчерів кластерів відносяться Apache Mesos, Apache Hadoop YARN і Spark. У HDInsight Spark виконується з використанням диспетчера кластерів YARN. Після підключення Spark знаходить виконавців на робочих вузлах кластера. Виконавці – це процеси, які вибирають і зберігають дані для додатків. Потім Spark відправляє виконавцям код додатків (визначений у JAR- або Python-файлах, переданих у SparkContext). В кінці SparkContext відправляє виконавцям підзадачі для виконання.

SparkContext виконує основну функцію користувача та здійснює різні паралельні операції на робочих вузлах. Потім SparkContext збирає результати операцій. Робочі засоби обчислюють дані з розподіленої файлової системи Hadoop і записують їх назад. Крім того, робочі вузли розміщують перетворені дані в кеш-пам'яті як стійкі розподілені набори даних (RDD).

SparkContext підключається до головного вузла Spark і відповідає за перетворення додатків в орієнтований граф (направлений ациклічний граф, DAG) для окремих задач. Такі завдання виконуються в рамках процесу виконання на робочих вузлах. Кожен додаток отримує окремі процеси виконавця, які залишаються активними під час виконання додатків і обробляють завдання в кількох потоках.

У роботі розглянута загальна модель обслуговування завдання у Spark кластері, що дозволяє оцінити ймовірність відмови завдання, серверну складову часу затримки до відгуку SparkContext, показники енергоспоживання компонент архітектури. Модель, згідно [2], представлена на рис.2.

Після надходження заявки на SparkContext, вона потрапляє у буфер (чергу) спеціальної системи-вирішувача (RPDE - Resource provisioning decision engine), здійснюючої пошук вільного ресурсу, здатного обслужити цю заявку. Як проілюстровано на рис.2, в системі можливі два види відмов в обслуговуванні заявки - при переповненні черги вхідних заявок системи-вирішувача і при недостатності фізичних, віртуальних і буферних ресурсів безпосередньої ланки обслуговування заявки. Також в цій моделі розглядається три типи груп ресурсів: гарячого (hot), теплого (warm) і холодного (cold) пулів фізичних серверів.

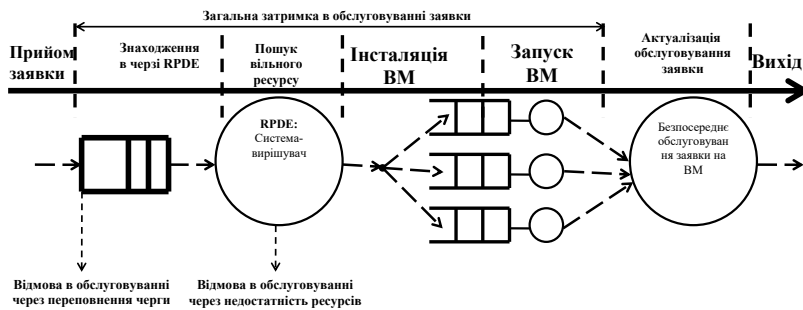


Рис.2. Загальна модель функціонування Spark кластера

У гарячому пулі фізичні сервера постійно включені і готові до розгортання на них необхідної кількості віртуальних машин (VM). На фізичних серверах теплого пулу живлення включене, але вони знаходяться в режимі очікування, і не готові до розгортання VM. Фізичні сервера холодного пулу знаходяться у вимкненому стані.

Таким чином, загальна модель функціонування Spark кластера повинна включати наступні елементи:

- модель системи-вирішувача;
- модель фізичних серверів гарячого пулу;

- модель фізичних серверів теплового пулу;
- модель фізичних серверів холодного пулу;
- модель безпосереднього обслуговування завдання.

Перелічені моделі в [2,5] побудовані з використанням математичного апарату неперервних та дискретних ланцюгів Маркова.

Стохастична модель фізичного сервера гарячого пулу у вигляді марковського графа представлена на рис.3. Граф складається з набору станів $S_{i,j,k}$: де i - кількість заявок в черзі, j - кількість розгорнутих ВМ (0 або 1), k - кількість ВМ, обслуговуючих заявки. Якщо ВМ не розгорнуті, то система простає; у такому режимі вона споживає h_l Ватт електроенергії (табл.1).

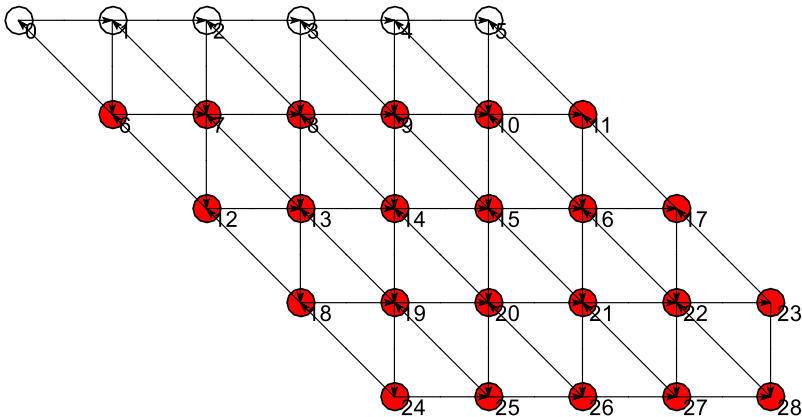


Рис.3. Модель споживання електроенергії при функціонуванні одного фізичного сервера в гарячому пулі, білим позначено стани зі споживанням h_l Ватт, червоним – зі споживанням h_l+k*v_a Ватт

Передбачається, що при запусненій ВМ в середньому споживається v_a Ватт електроенергії. Виходячи з цього, кожен стан $S_{i,j,k}$ характеризується споживанням $r(i,j,k)=h_l+k*v_a$ Ватт електроенергії. Загальне середнє енергоспоживання фізичного сервера W_h визначається як сума сталих значень енергоспоживання усіх його станів:

$$W_h = \sum_{\forall(i,j,k)} P_{i,j,k} \cdot (h_l + k \cdot v_a)$$

Сумарне енергоспоживання гарячого пулу, що містить n_h серверів складе $T_h=n_h*W_h$ Ватт електроенергії.

Таблиця 1

Енергоспоживання станів моделі гарячого пулу

№ з/п	Індекси станів	Потужність, що споживається (Ватт)
1	(0,0,0) (i,1,0), 0≤i≤Lh	h_l
2	(i,j,k), 1≤k≤mh	h_l+k*v_a

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

У статті було розглянуто марковські моделі енергоспоживання компоненти Spark інфраструктури: гарячого пулу фізичних серверів. Розроблені моделі подібні до класичних СМО-моделей з чергами, але на відміну від них мають декілька ярусів. Подальші дослідження слід спрямувати на