УДК 004.4'2

РОЗШИРЕННЯ СИСТЕМИ СИНТЕЗУ ПРОГРАМ З МЕТОЮ АНАЛІЗУ ВЕЛИКИХ НАБОРІВ ДАНИХ

О.М. Овдій

Аналіз великих обсягів даних є актуальною проблемою у сучасному світі. У даній роботі представлено розширення онлайнового діалогового конструктору синтаксично правильних програм ОДСП для проектування та синтезу програм з метою аналізу великих наборів даних на базі програмного забезпеченням Apache Hadoop. Перевагою використовуваного у інструментарії підходу є застосування методу, який забезпечує синтаксичну правильність алгоритмів та програм, що проектуються. Проведено експерименти та проілюстровано роботу системи на прикладі проектування програм для аналізу великого набору метеорологічних даних. Даний підхід є перспективним для виконання наукових досліджень, зокрема в сфері метеорології.
Ключові слова: проектування і синтез програм, розподілені обчислення, Великі Дані, Map Reduce, Hadoop.

Анализ больших объемов данных является актуальной проблемой в современном мире. В данной работе представлено расширение онлайнового диалогового конструктора синтаксически правильных программ ОДСП для проектирования и синтеза программ с целью анализа больших наборов данных на базе программного обеспечения Apache Hadoop. Преимуществом используемого в инструментарии подхода является применение метода, который обеспечивает синтаксическую правильность алгоритмов и программ, которые проектируются. Проведены эксперименты и проиллюстрировано работу системы на примере проектирования программ для анализа большого набора метеорологических данных. Данный подход является перспективным для проведения научных исследований, в частности в области метеорологии.
Ключевые слова: проектирование и синтез программ, распределенные вычисления, Большие Данные, Map Reduce, Hadoop.

Analysis of large data sets is a challenging problem in the modern world. This paper presents an extension of the online dialog designer of syntactically correct programs ODSP for programs designing and synthesis for analyzing large data sets based on Apache Hadoop software. The advantage of the proposed approach is the use of a method that ensures the syntactic correctness of algorithms and programs that were designed. Experiments have been carried out and the operation of the system is illustrated by the example of program designing for analyzing a large meteorological data set. This approach is promising for the scientific research conducting, in particular in the field of meteorology.
Key words: design and synthesis of programs, distributed computing, Big Data, Map Reduce, Hadoop.

Вступ

В останні десятиліття світ переживає інформаційний вибух. Сьогодні доступна величезна кількість цифрових даних, які позначають терміном Великі Дані (Big Data) [1, 2] і обсяг цих даних стрімко зростає з кожним днем. Для зберігання, обробки та аналізу таких даних використовуються високопродуктивні обчислювальні систем, на основі Грід та хмарних технологій. Розвиток таких систем дозволяє впроваджувати більш складні програми обробки великих обсягів даних, та виконувати дослідження на розподілених обчислювальних платформах як в бізнес так і в науковому секторах. Поряд з цим, із зростом можливостей зростає і складність систем, а відповідно складність розробки програм для них. Великий обсяг обчислень над великими масивами даних потребує впровадження паралельних реалізацій та забезпечення їх оптимізації. В зв'язку з цим виникає необхідність створення спеціальних високорівневих засобів генерації високопродуктивних програм для таких платформ. Ще більше ця задача є актуальною для надання предметним експертам та науковцям, що зазвичай не є професійними програмістами, інструментів, якими вони зможуть з легкістю користуватися.

В даній статті розглянуто розширення можливостей раніше розроблених засобів для автоматизації проектування програм [3–10] для роботи з програмним забезпеченням для зберігання і обробки великих наборів даних Apache Hadoop [11, 12] та Apache Pig [13–15], яке широко використовується для реалізації розподілених обчислень. Проведено експерименти та проілюстровано роботу системи на прикладі проектування програм для аналізу великого набору метеорологічних даних.

1. Задача обробки великих обсягів даних

В науковому світі, також як і в бізнес світі на даний час доступна величезна кількість цифрових даних, Big Data. Вони надходять з чисельних наукових експериментів та спостережень і потребують зберігання, обробки та аналізу. Крім того, доступність таких даних та засобів їх обробки породжує нові методи досліджень, орієнтованих на дані, які базуються на інтелектуальному аналізі великих сховищ даних для виявлення правил, прихованих у даних. Базовими характеристиками Big Data є великий обсяг, різноманіття, наявність як структурованих так і не структурованих даних, велика швидкість збільшення та необхідність швидкісної обробки та аналізу.

Одним з базових принципів обробки Big Data є розподілені обчислення, тобто горизонтальна масштабованість, яка забезпечує зберігання та обробку даних, розподілених на тисячі обчислювальних вузлів, без зниження продуктивності [2]. Найуживанішою моделлю розподілених обчислень великих обсягів даних є модель MapReduce [16]. MapReduce було розроблено компанією Google, з метою використання для паралельних обчислень над дуже великими наборами даних в комп'ютерних кластерах, що складаються в більшості зі звичайних, не суперпотужних комп'ютерів. MapReduce – це модель програмування та відповідна реалізація для обробки великих наборів даних. Базис моделі складають дві функцій Map() та Reduce(). Функція Map() обробляє пару ключ/значення, щоб створити набір проміжних пар ключ/значення, а функція Reduce() об'єднує всі проміжні значення, пов'язані з одним і тим же проміжним ключем. Програми, написані за цією моделлю, автоматично розпаралелюються і виконуються на кластері. Система керування дбає про подробиці розподілу вхідних даних, планування виконання програми на кластері, обробку збоїв та комунікацію між обчислювальними вузлами. Це дозволяє більшому колу розробників, без досвіду роботи з паралельними та розподіленими обчисленнями, використовувати ресурси великих розподілених систем. Реалізація MapReduce є високопродуктивною та добре масштабованою, що дозволяє обробляти терабайти даних на тисячах вузлів.

Одною з найпоширеніших реалізацій парадигми MapReduce є Apache Hadoop [11, 12]. Це проект фонду Apache Software Foundation, вільно розповсюджуване програмне забезпечення для розподіленого зберігання й обробки великих обсягів даних на великих комп'ютерних кластерах. Використовується для реалізації пошукових та контекстних механізмів багатьох потужних інформаційних систем, наприклад, у Facebook та Yahoo!.

Перед зберіганням Hadoop виконує розбивку вхідних файлів на блоки та розподіляє їх серед вузлів у кластері. Потім у вузли для обробки даних передається код, що буде виконуватися паралельно. Таким чином вузли маніпулюють даними, які на них зберігаються. Це дозволяє обробляти набір даних швидше і ефективніше.

Проект Apache Hadoop включає такі модулі:

– Hadoop Common: загальні утиліти, які підтримують інші модулі Hadoop;

– Hadoop HDFS: розподілена файлова система, яка забезпечує високопродуктивний доступ до даних програми;

– Hadoop YARN: основа для планування роботи та керування ресурсами кластерів;

– Hadoop MapReduce: система для паралельної обробки великих наборів даних.

В екосистему Hadoop в проекті Apache входить багато інших програмних продуктів, таких як HBase (масштабована, розподілена база даних, яка підтримує структуроване зберігання даних для великих таблиць), ZooKeeper (високопродуктивна координаційна служба для розподілених програм), Hive (інфраструктура сховища даних, яка забезпечує узагальнення даних і спеціальні запити) та багато інших.

Незважаючи на те, що MapReduce надає певний рівень абстракції для роботи з розподіленими обчисленнями та великими обсягами даних все ж це інструмент для професійних програмістів, а не прикладних експертів та науковців. Окрім того слід зауважити, що наукові дослідження, а зокрема наукові робочі процеси (Scientific Workflow) [17, 18] зосереджені на самих даних, та на їх перетвореннях, а не на процесах. Тому, на даний момент, є окремі програмні системи для реалізації Scientific Workflow, що відрізняються від бізнес аналогів.

З огляду на вищесказане в якості посередника для роботи з Hadoop було обрано Apache Pig [13–15]. Це одним з проектів екосистеми Apache Hadoop, що підвищує рівень абстракції для обробки великих наборів даних. Пристосування необхідного алгоритму обробки даних до моделі MapReduce є задачею нетривіальною і часто вимагає декількох складних етапів. За допомогою Apache Pig можна використовувати набагато багатіші структури даних, з багатьма значеннями та вкладеннями, і перетворення, які можна застосувати до даних, набагато потужніші. Найважливішою властивістю програм Pig є те, що їх структура піддається суттєвому розпаралелюванню, що, в свою чергу, дозволяє їм обробляти дуже великі набори даних.

Apache Pig – це платформа для аналізу великих наборів даних, яка на даний час складається з:

– мови високого рівня Pig Latin, що використовується для вираження потоків даних.

– середовище виконання, для запуску програм Pig Latin.

Мова Pig Latin є мовою потоків даних та має наступні ключові властивості:

– Простота програмування. Дуже легко досягнути паралельного виконання простих завдань аналізу даних. Комплексні завдання, що складаються з багатьох взаємопов'язаних перетворень даних, явно кодуються як послідовності потоку даних, що робить їх простими для написання, розуміння та підтримки.

– Можливості оптимізації. Спосіб, за допомогою якого кодуються завдання, дозволяє системі оптимізувати їх виконання автоматично.

– Розширюваність. Користувачі можуть створювати свої власні функції для виконання спеціальної обробки.

Програма Pig Latin складається з серії операцій або перетворень, які застосовуються до вхідних даних для отримання результату. В цілому операції описують потік даних, який середовище виконання Pig перетворює на виконуване представлення, а потім запускає на виконання. Pig сам перетворює програму на серію задач MapReduce, це дозволяє зосередитись на даних, а не на характері виконання. Одним з недоліків MapReduce називають довгий цикл розробки навіть для простих операцій, а за допомогою декількох рядків Pig Latin можна обробляти терабайти даних. Недарма вона була створена в Yahoo! щоб дозволити дослідникам та інженерам легше обробляти та аналізувати величезні набори даних. Мова має SQL-подібний синтаксис та є своєрідним поєднанням декларативного стилю SQL та низькорівневого процедурного стилю MapReduce. Pig легко розширюється за допомогою визначених користувачем функцій, налаштовуються практично всі частини обробки: завантаження, зберігання, фільтрація, групування та об'єднання.

Завдяки властивостям платформи Apache Pig вона добре підходить для виконання досліджень над великими наборами даних та використання у наукових робочих процесах. Тому її було обрано для подальшого розвитку розроблюваної системи синтезу програм ОДСП [3–5].

2. Система синтезу програм ОДСП

Одним з перспективних напрямів у розробці та дослідженні систем паралельних і розподілених обчислень нині є побудова програмних абстракцій у вигляді алгеброалгоритмічних мов і моделей, що ставить своєю метою розвиток архітектурно- і мовно-незалежних засобів програмування для мультипроцесорних обчислювальних систем і мереж. У роботах [3–10] були запропоновані теорія, методологія та інструментарій для автоматизованого проектування паралельних програм, що ґрунтуються на засобах високорівневої алгеброалгоритмічної формалізації та автоматизації перетворень програм.

У роботах [3–5] розглядається розроблена системи синтезу програм – Онлайновий Діалоговий конструктор Синтаксично Правильних програм (ОДСП). Особливість інструментарію ОДСП полягає у використанні Інтернет-технологій та у розподіленій архітектурі системи. Система ОДСП призначена для діалогового проектування, генерації й запуску програм. Вона базується на сервісно-орієнтованій архітектурі та спрямована на багатокористувальницьке використання інструментарію через мережу Інтернет.

Система ОДСП призначена для проектування та генерації програм на основі високорівневих специфікацій алгоритмів. Вона ґрунтується на використанні апарату систем алгоритмічних алгебр (САА) та їх модифікацій [6]. Модифіковані САА (САА-М) призначені для формалізації процесів мультиобробки, що виникають при конструюванні програмного забезпечення в мультипроцесорних системах. Також в системі використано метод діалогового конструювання синтаксично правильних програм (ДСП-метод). На відміну від традиційних синтаксичних аналізаторів, ДСП-метод орієнтований не на пошук і виправлення синтаксичних помилок, а на виключення можливості їх появи в процесі побудови алгоритму. Основна ідея методу полягає в порівневому конструюванні схем зверху вниз шляхом суперпозиції мовних конструкцій САА-М. Система ОДСП забезпечує генерацію відповідних програм в цільових мовах програмування для сучасних паралельних та розподілених середовищ виконання.

У наступному розділі розглянуто розширення системи ОДСП для проектування та синтезу програм мовою Pig Latin з метою аналізу великих наборів даних на базі Apache Hadoop та Apache Pig і продемонстровано її використання на прикладах з області метеорології.

3. Розширення системи ОДСП з метою аналізу великих наборів даних

У попередній роботі [19] було запропоновано реалізацію Інтернет-порталу для надання послуг метеорологічного прогнозування, яка поєднує комплексність використання адекватних фізичних моделей атмосферних процесів з ефективними обчислювальними схемами та методами програмування високопродуктивних обчислень на мультипроцесорних системах, що дає змогу досягати належного ступеня точності, повноти та своєчасності інформації, необхідної для задоволення потреб широкого кола користувачів.

Науковцям, які ведуть дослідження в сфері метеорології необхідно аналізувати величезний обсяг історичних та прогностичних даних. Наприклад, Національне управління океанічних і атмосферних досліджень США (NOAA) [20] щодня видає близько 20 терабайт даних, включаючи дані сотень метеорологічних станцій, результати прогнозів, данні супутників і т. д. Для роботи з такими великими наборами даних науковцям необхідні інструменти, якими вони можуть користуватися без поглиблених знань програмування та архітектури обчислювальних систем.

З метою надання таких інструментів систему ОДСП було розширено та додано нові мовні конструкції для проектування та синтезу програм мовою Pig Latin, що дозволяє виконувати аналіз великих наборів даних на базі Apache Hadoop та Apache Pig. В перспективі це може бути використано для подальшого розвитку Інтернет-порталу і надання додаткових послуг пов'язаних з аналізом метеорологічних даних.

Для прикладу, було використано набор даних, що надається Національним центром кліматичних даних США NCDC [21]. Він містить показники метеостанцій та зберігається в текстових архівованих файлах за допомогою лінійно-орієнтованого формату ASCII, в якому кожний рядок є записом. Формат підтримує широкий набір метеорологічних елементів, багато з яких є необов'язковими або змінної довжини. Файли даних організовані за датою та метеостанцією, та зберігають інформацію починаючи з 1901 року. Загальний поточний обсяг набору даних складає близько 100 Гб.

Для проведення експерименту було розгорнуто інфраструктуру Apache Hadoop у псевдорозподіленому режимі. Набор даних NCDC було оброблено та перенесено у розподілену файлову систему HDFS.

Далі на рис. 1 показана побудована в системі ОДСП схема програми мовою Pig Latin для пошуку мінімальної та максимальної температури за місяцями.

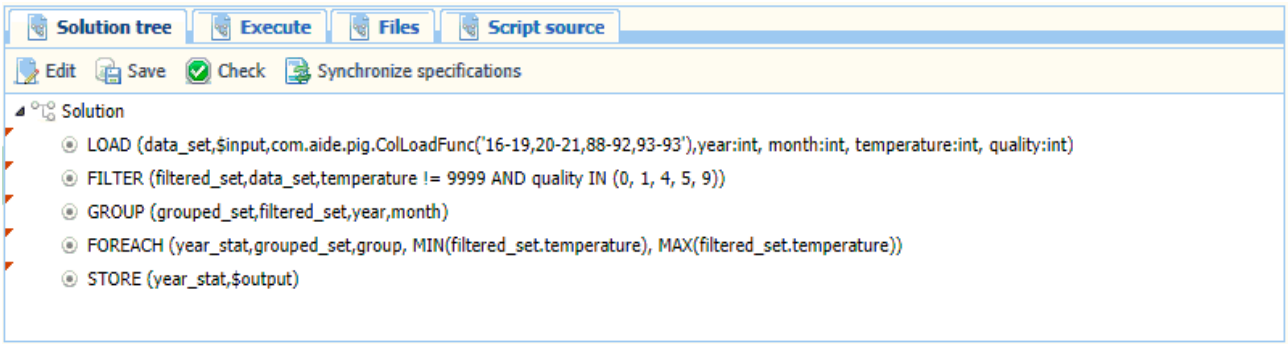

Рис. 1. Фрагмент копії екрану системи ОДСП з алгоритмом та списком конструкцій для обчислення максимальної та мінімальної температури за місяцями

За допомогою системи ОДСП було згенеровано програмний код мовою Pig Latin:

```
data_set = LOAD '$input'
USING com.aide.pig.ColLoadFunc('16-19,20-21,88-92,93-93')
     AS (year:int, month:int, temperature:int, quality:int);
filtered_set = FILTER data_set BY temperature != 9999 AND quality
     IN (0, 1, 4, 5, 9);
grouped_set = GROUP filtered_set BY (year,month);
year_stat = FOREACH grouped_set GENERATE group,
     MIN(filtered_set.temperature), MAX(filtered_set.temperature);
STORE year_stat into '$output';
```

Фрагмент результатів виконання програми (температура наведена у градусах Цельсія помножених на 10):

```
((2017,1),-597,592)
((2017,2),-653,600)
((2017,3),-714,600)
((2017,4),-746,545)
((2017,5),-764,570)
((2017,6),-822,540)
((2017,7),-814,580)
((2017,8),-786,540)
((2017,9),-777,530)
((2017,10),-687,506)
((2017,11),-600,489)
((2017,12),-600,500)
```

На рис. 2 показано побудовану в системі ОДСП схему програми мовою Pig Latin для пошуку мінімальної, максимальної мінімальної (найтепліша ніч року), мінімальної максимальної (найхолодніший день року) та максимальної температури за роками.

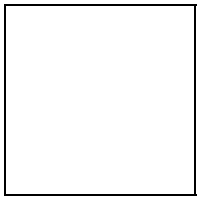

Рис. 2. Фрагмент копії екрану системи ОДСП з алгоритму та списком конструкцій
для обчислення більш детальних показників температури за роками

Даному алгоритму відповідає наступний код мовою Pig Latin, що було згенеровано за допомогою системи ОДСП:

```
data_set = LOAD '$input'
USING com.aide.pig.ColLoadFunc('16-19,20-21,16-23,88-92,93-93')
     AS (year:int, month:int, date:int, temperature:int, quality:int);
filtered_set = FILTER data_set BY temperature != 9999 AND quality
     IN (0, 1, 4, 5, 9);
group_by_date_set = GROUP filtered_set BY (year,date);
date_set = FOREACH group_by_date_set GENERATE group.year as year,
     MIN(filtered_set.temperature) as min_day_temperature,
     MAX(filtered_set.temperature) as max_day_temperature;
grouped_by_year_set = GROUP date_set BY year;
year_stat = FOREACH grouped_by_year_set GENERATE group,
     MIN(date_set.min_day_temperature), MAX(date_set.min_day_temperature),
     MIN(date_set.max_day_temperature), MAX(date_set.max_day_temperature);
STORE year_stat into '$output';
```

Фрагмент результатів виконання програми:

```
(1976,-820,-453,450,570)
(1977,-830,-420,398,580)
(1978,-930,-430,400,550)
(1979,-840,-460,407,600)
(1980,-780,-460,420,600)
(1981,-850,-420,415,580)
(1982,-930,-427,430,617)
(1983,-931,-453,400,616)
(1984,-932,-443,390,617)
(1985,-932,-446,397,611)
(1986,-901,-450,395,607)
```

З наведених прикладів видно, що програми для обробки великих обсягів даних на розподілених обчислювальних платформах, таких як Apache Hadoop, можна з легкістю проектувати за допомогою інструменту синтезу програм ОДСП. І, як зазначено вище, в перспективі дану роботу може бути використано для подальшого розвитку Інтернет-порталу і надання додаткових послуг пов’язаних з аналізом метеорологічних даних.

Висновки

В роботі виконано розширення онлайнового діалогового конструктору синтаксично правильних програм ОДСП для проектування та синтезу програм мовою потоків даних Pig Latin з метою аналізу даних на базі

програмного забезпеченням для зберігання і обробки великих наборів даних Apache Hadoop. Перевагою використовуваного у інструментарії підходу є застосування методу, який забезпечує синтаксичну правильність алгоритмів та програм, що проектуються. Проведено експерименти та проілюстровано роботу системи на прикладі проектування програми для аналізу великого набору метеорологічних даних. Даний підхід показав свою ефективність для виконання наукових досліджень, що потребують аналізу великих обсягів даних, зокрема у сфері метеорології.

Література

1. Lynch C. Big data: How do your data grow? Nature. 2008. № 455(7209). P. 28–29.
2. NIST Big Data Interoperability Framework: Volume 1, Definitions. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-1.pdf (дата звернення: 26.02.2018).
3. Дорошенко А.Ю., Бекетов О.Г., Іванів Р.Б., Іовчев В.О., Мироненко І.О., Яценко О.А. Автоматизована генерація паралельних програм для графічних прискорювачів на основі схем алгоритмів. *Проблеми програмування*. 2015. № 1. С. 19–28.
4. Андон Ф.И., Дорошенко А.Е., Бекетов А.Г., Иовчев В.А., Яценко Е.А. Инструментальные средства автоматизации параллельного программирования на основе алгебры алгоритмов. *Кибернетика и системный анализ*. 2015. № 1. С. 162–170.
5. Дорошенко А.Ю., Іваненко П.А., Овдій О.М., Яценко О.А. Автоматизоване проектування програм для розв'язання задачі метеорологічного прогнозування. *Проблеми програмування*. 2016. № 1. С. 102–115.
6. Андон Ф.И., Дорошенко А.Е., Цейтлин Г.Е., Яценко Е.А. Алгеброалгоритмические модели и методы параллельного программирования. Киев: Академпериодика, 2007. 631 с.
7. Дорошенко Е.А., Яценко Е.А. О синтезе программ на языке Java по алгеброалгоритмическим спецификациям. *Проблеми програмування*. 2006. № 4. С. 58–70.
8. Яценко Е.А. Интеграция средств алгебры алгоритмов и переписывания термов для разработки эффективных параллельных программ. *Проблеми програмування*. 2013. № 2. С. 62–70.
9. Дорошенко А.Ю., Бекетов О.Г., Яценко О.А, Вітряк Є.А., Павлючин Т.О. Розробка сервісно-орієнтованих засобів для запуску паралельних програм на мультипроцесорному кластері. *Проблеми програмування*. 2014. № 4. С. 3–14.
10. Дорошенко Е.А., Овдей О.М., Яценко Е.А. Онтологические и алгеброалгоритмические средства автоматизации проектирования параллельных программ для "облачных" платформ. *Кибернетика и системный анализ*. 2017. Т. 53, № 2. С. 181–192.
11. Apache Hadoop: сайт. URL: http://hadoop.apache.org/ (дата звернення: 26.02.2018).
12. White T. Hadoop: The Definitive Guide, 4th Edition. O'Reilly Media, Inc. 2015. 756 p.
13. Apache Pig: сайт. URL: http://pig.apache.org/ (дата звернення: 26.02.2018).
14. Gates A. Programming Pig Dataflow Scripting with Hadoop. O'Reilly Media. 2011. 224 p.
15. Olston Ch., Reed B., Srivastava U., Kumar R., Tomkins A. Pig latin: A not-so-foreign language for data processing. In Proc. ACM SIGMOD Int'l Conference on Management of Data. 2008. P. 1099–1110.
16. Dean J., Ghemawat S. Mapreduce: Simplified data processing on large clusters. In Proc. of the 6th USENIX OSDI. 2004. P. 137–150.
17. Atkinson M., Gesing S., Montagnat J., Taylor I. Scientific workflows: Past, present and future. Future Generation Computer Systems, Elsevier. 2017. N 75. P. 216–227.
18. Singh M.P., Vouk M.A. Scientific workflows: scientific computing meets transactional workflows. InProc. of the NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future Directions Univ. Georgia, Athens, GA, USA, 1996. P. 28–34.
19. Дорошенко А.Ю., Іваненко П.А., Овдій О.М., Павлючин Т.О., Вітряк Є.А. До створення Інтернет-порталу надання послуг метеорологічного прогнозування на мультипроцесорній платформі. *Проблеми програмування*. 2015. № 3. С. 24–32.
20. National Oceanic and Atmospheric Administration (NOAA): сайт. URL: http://www.noaa.gov/ (дата звернення: 26.02.2018).
21. National Climatic Data Center (NCDC): сайт. URL: https://www.ncdc.noaa.gov/ (дата звернення: 26.02.2018).

References

1. Lynch C. (2008) Big data: How do your data grow? *Nature*, 455(7209), P. 28–29.
2. NIST Big Data Interoperability Framework: Volume 1, Definitions. [online] Available at: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-1.pdf [Accessed 26 Feb. 2018].
3. Doroshenko, A.Yu., Beketov, O.G., Ivaniv R.B., Iovchev, V.O., Myronenko, I.O. & Yatsenko, O.A. (2015) Automated generation of parallel programs for graphics processing units based on algorithm schemes. *Problems in programming*. (1). P. 19–28. (in Ukrainian).
4. Andon, P.I., Doroshenko, A.Yu., Beketov, O.G., Iovchev, V.O. & Yatsenko O.A. (2015) Software tools for automation of parallel programming on the basis of algebra of algorithms. *Cybernetics and systems analysis*. (1). P. 162–170. (in Russian).
5. Doroshenko, A.Yu., Ivanenko, P.A., Ovdii, O.M., & Yatsenko, O.A. (2016) Automated design of programs for solving the task of meteorological forecasting. *Problems in programming*. (1). P. 102–115. (in Ukrainian).
6. Andon, P.I. et al. (2007) *Algebra-algorithmic models and methods of parallel programming*. Kiev: Academperiodika. (in Russian).
7. Doroshenko, A.Yu. & Yatsenko O.A. (2006) About the synthesis of Java programs by algebra-algorithmic specifications. *Problems in programming.* (4). P. 58–70. (in Russian).
8. Yatsenko O.A. (2013) Integration of algebra-algorithmic tools and term rewriting for efficient parallel programs development. *Problems in programming*. (2). P. 62–70. (in Russian).
9. Doroshenko, A.Yu., Beketov, O.G. Yatsenko, O.A., Pavliuchyn, T.O. & Vitriak, I.A. (2014) Development of the service-oriented soft-ware for launching parallel programs on a multiprocessor cluster. *Problems in programming*. (4). P. 3–14. (in Ukrainian).
10. Doroshenko A.Yu., Ovdii O.M., Yatsenko O.A. (2017) Ontological and algebra-algorithmic tools for automated design of parallel programs for cloud platforms. *Cybernetics and Systems Analysis*. 53(2). P. 181–192. (in Russian).
11. Hadoop.apache.org. *Apache Hadoop Official Website***.** [online] Available at: http://hadoop.apache.org/ [Accessed 26 Feb. 2018].
12. White T. (2015) *Hadoop: The Definitive Guide, 4th Edition.* O'Reilly Media, Inc.
13. Pig.apache.org. *Apache Pig Official Website.* [online] Available at: http://pig.apache.org/ [Accessed 26 Feb. 2018].
14. Gates A. (2011) *Programming Pig Dataflow Scripting with Hadoop*. O'Reilly Media.

15. Olston Ch., Reed B., Srivastava U., Kumar R. & Tomkins A. (2008) Pig latin: A not-so-foreign language for data processing. In: *ACM SIGMOD Int'l Conference on Management of Data*, P. 1099–1110.
16. Dean J. & Ghemawat S. (2004) Mapreduce: Simplified data processing on large clusters. In: *6th USENIX OSDI*, P. 137–150.
17. Atkinson M., Gesing S., Montagnat J. & Taylor I. (2017) Scientific workflows: Past, present and future. *Future Generation Computer Systems, Elsevier*. 75. P. 216–227.
18. Singh M.P. & Vouk M.A. (1996) Scientific workflows: scientific computing meets transactional workflows. In: *NSF Workshop on Workflow and Process Automation in Information Systems: State-of-the-Art and Future* Directions Univ. Georgia, Athens, GA, USA. P. 28–34.
19. Doroshenko, A.Yu., Ivanenko, P.A., Ovdii, O.M., Pavliuchyn, T.O. & Vitriak, I.A. (2015) Creation of an Internet portal providing meteorological forecasting services on multiprocessor platform. *Problems in programming*. (3). P. 24–32. (in Ukrainian).
20. Noaa.gov. *National Oceanic and Atmospheric Administration (NOAA)*. [online] Available at: http://www.noaa.gov/ [Accessed 26 Feb. 2018].
21. Ncdc.gov. *National Climatic Data Center (*NCDC*).* [online] Available at: https://www.ncdc.noaa.gov/ [Accessed 26 Feb. 2018].

Про автора:

Овдій Ольга Михайлівна,
молодший науковий співробітник
Інституту програмних систем НАН України.
Кількість наукових публікацій в українських виданнях – 21.
Кількість наукових публікацій в зарубіжних виданнях – 3.
http://orcid.org/0000-0002-8891-7002.

Місце роботи автора:

Інститут програмних систем НАН України,
03187, м. Київ-187, проспект Академіка Глушкова, 40.
Тел.: (38)(044) 526-60-33.
E-mail: olga.ovdiy@gmail.com