

## **ФРЕЙМВОРК ДЛЯ РОЗРОБКИ GRID-ЗАСТОСУВАНЬ З ПІДТРИМКОЮ РІЗНИХ ТИПІВ ОБЧИСЛЮВАЛЬНИХ ЗАДАЧ ВЕЛИКОЇ РОЗМІРНОСТІ**

**В.В. КАЗИМИР, О.А. ПРІЛА**

### **ВСТУП**

Незважаючи на розвиток та впровадження grid-технологій у різні галузі науки, створення метакомп'ютерного середовища на базі внутрішніх (корпоративних) обчислювальних ресурсів підприємства або вирішення прикладних задач в існуючому grid-середовищі являє собою складну технічну задачу та пов'язано із значними часовими та матеріальними затратами. Основними проблемами є відсутність високорівневих засобів, недостатність відповідної документації та спеціалістів, відсутність інтеграції між існуючим проміжним програмним забезпеченням грид (ППЗ, middleware) [1, 2].

На сьогодні існують віртуальні організації (ВО) [3], в межах яких організовані питання керування ресурсами та політикою доступу користувачів, однак більшість ВО не надають зручного інтерфейсу для використання спеціалізованих грид-сервісів та для співпраці учасників ВО над розробкою нових сервісів. Навіть базові грид-операції виконуються в консольному режимі, що робить неможливим використання грид-середовища прикладними спеціалістами.

Метою роботи є формування вимог до високорівневих засобів створення grid-застосувань, окреслення недоліків існуючих засобів та проектування фреймворку для створення grid-застосувань згідно окреслених вимог.

### **ПРОБЛЕМИ ЗАСТОСУВАННЯ GRID-СЕРЕДОВИЩА ДЛЯ ВИРІШЕННЯ ПРИКЛАДНИХ ЗАДАЧ**

Слід відмітити схожість інфраструктур ВО, створених для різних галузей науки, оскільки найбільш адаптованими до вирішення у грид-середовищі є задачі певного типу. Розглянемо типи задач, які вирішуються у grid-середовищі.

1. Задача являє собою задачу великої розмірності, її програмна реалізація може (але не обов'язково) характеризуватись внутрішнім паралелізмом, однак при виконанні задачі немає потреби у використанні всієї доступної grid-інфраструктури. Тобто необхідність використання grid-середовища зводиться лише до використання віддаленого потужного кластерного ресурсу. Причому ефективність використання кластерного ресурсу може значно знижуватись у зв'язку з затратами на пересилку даних для виконання обчислень. З точки зору розробника grid-порталу такі задачі являють собою найпростіший тип задач та ефективність їх вирішення залежить лише від ефективності програмної реалізації самої задачі. Подібні задачі будемо називати такими, що являють собою єдиний обчислювальний блок.
2. Задача являє собою єдиний обчислювальний блок та декомпонується на декілька підзадач шляхом розподілу вхідних даних на декілька порцій для паралельної обробки. Зазначена модель програмування має назву Single Program Multi Data

(SPMD) [4]. Подібні задачі будемо називати такими, яким властивий паралелізм за даними.

3. Задача, яка декомпонується на декілька паралельно-послідовних підзадач з певною схемою синхронізації обчислень. Особливості такого типу задач розглянемо на прикладі застосування паралельно-конвейєрних обчислень до задачі аналізу економічних ризиків.

Алгоритм статистичного методу аналізу ризиків включає наступні етапи: проведення кореляційно-регресійного аналізу; тестування моделі на контрольній вибірці; прогнозування значень факторів та імітаційне моделювання (рис. 1).

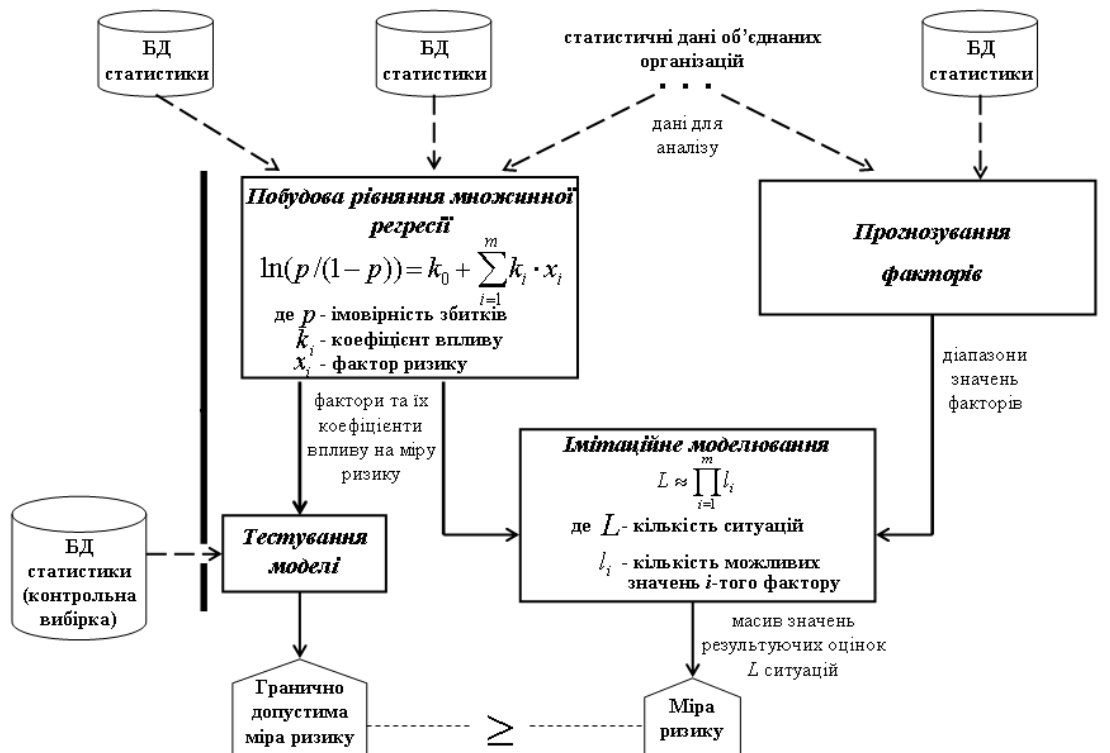


Рис. 1. Застосування паралельно-конвейєрних обчислень до задачі аналізу економічних ризиків

Перший етап полягає у визначенні факторів, під впливом яких формується ризик, і ступеня впливу кожного з обраних факторів на результуючу міру ризику, тобто проведення кореляційного аналізу статистичних даних і побудова рівняння багатфакторної логістичної регресії. Складність обчислень лінійно залежить від кількості факторів. Крім того, похибка оцінки параметра обернено пропорційна обсягу статистичної вибірки.

Другий етап аналізу полягає в тестуванні отриманої моделі. На контрольній вибірці здійснюється оцінка помилок 1-го й 2-го роду [5], на підставі чого робиться висновок про можливість застосування побудованої моделі та коректується значення гранично допустимої міри ризику.

Прогнозування факторів – це визначення їхніх значень у найближчому майбутньому. Значення факторів можуть бути задані характеристиками конкретної ситуації, розраховані на базі статистики або визначені виходячи із загальної макроекономічної

ситуації з урахуванням інших видів ризиків. Значення факторів можуть являти собою як певне значення або набір значень, так і ймовірнісний діапазон значень.

У ході процесу імітації будуються послідовні сценарії з використанням невизначених вхідних даних, які покладаються випадковими величинами. Результати імітації збираються й аналізуються статистично для того, щоб оцінити міру ризику. Залежність складності обчислень від числа факторів і діапазону їхніх значень – експоненційна.

Кореляційно-регресійний аналіз та обчислення щодо прогнозування факторів можна виконувати незалежно, так само як тестування побудованої моделі можна проводити паралельно з етапами прогнозування та імітаційного моделювання. Етап імітаційного моделювання можна розпочинати, як тільки підготовлено частковий набір значень факторів на етапі прогнозування (тобто не дочекуючись закінчення етапу прогнозування, паралельно йому). Оскільки складові задачі статистичного аналізу міри ризику є частково незалежними, то задачі властивий паралелізм за обчисленнями. Є доцільним забезпечення конвеєрної схеми організації обчислень у Grid-середовищі. Більше того, до кожної із складових підзадач може бути застосований паралелізм за даними.

Зазначена вище задача являє собою комбінований тип, оскільки також включає особливості другого типу задач.

Окремо необхідно виділити частковий випадок третього типу задач, коли задача декомпозується на повністю незалежні паралельні підзадачі. В цьому випадку процес адаптації до grid значно спрощується, оскільки синхронізація між блоками обчислень не потрібна.

Беручи до уваги принципи конвеєрної обробки [4], третій тип задач будемо називати такими, що характеризуються конвеєрною схемою обчислень.

Основними проблемами застосування grid-середовища для вирішення прикладних задач є наступні:

- 1) необхідність установки набору пакетів програмного забезпечення grid на клієнтській машині;
- 2) складність використання grid-середовища у консольному режимі прикладними користувачами;
- 3) складність формування низькорівневих специфікацій задачі (причому єдиного формату специфікації задачі для різного ППЗ на сьогодні не існує);
- 4) складність адаптації прикладної задачі до grid-середовища. Із зазначених вище типів задач, лише перші два частково підтримуються проміжним програмним забезпеченням grid на сьогодні.

Розробка високорівневих інструментів використання grid-середовища та розробки grid-порталів для вирішення прикладних задач є актуальною задачею.

## ОГЛЯД ІСНУЮЧИХ ЗАСОБІВ РОЗРОБКИ GRID-ЗАСТОСУВАНЬ

У таблиці 1 наведено результати дослідження вікритих фреймворків та бібліотек для створення високорівневих grid-застосувань: Ganga [6], Grid Portal Development Kit (GRDK) [7], GridSphere Portal Framework [8], SimpleGrid [9], Intel GPE API [10], Libarcclient [2]. Огляд існуючих засобів створення grid-застосувань проводився згідно зазначених у таблиці 1 критеріїв.

**Таблиця 1.** Порівняльний аналіз існуючих засобів створення grid-застосувань

| Критерій | Ganga | Grid Portal | GridSphere | SimpleGrid | Intel GPE | Libarc- |
|----------|-------|-------------|------------|------------|-----------|---------|
|----------|-------|-------------|------------|------------|-----------|---------|

| / фрейм-<br>ворк |   | <b>Develop-<br/>ment Kit</b> | <b>Portal<br/>Framework</b> |                     | <b>API</b>          | <b>client</b>                         |
|------------------|---|------------------------------|-----------------------------|---------------------|---------------------|---------------------------------------|
| API              | скрипто-<br>вий<br>інтерфейс<br>на мові<br>Python | Java-<br>бібліотеки          | Portlet API                 | Java-<br>бібліотеки | Java-<br>бібліотеки | C++-<br>бібліотека<br>Python,<br>Java |

Продовження таблиці 1

| Критерій /фрейм-ворк                    | Ganga  | Grid Portal Development Kit       | GridSphere Portal Framework                             | SimpleGrid   | Intel GPE API                 | Libarc-client                         |
|---|--|-----------------------------------|---|--|-------------------------------|---------------------------------------|
| Інтерфейс користувача                   | командна строка // desktop-застосування            | web-застосування з підтримкою SSL | web-застосування на базі технології портлетів (portlet) | командна строка // web-застосування на базі технології портлетів | desktop- // web-застосування  | desktop- // web-застосування          |
| Підтримка ППЗ                           | gLite, Condor                                      | Globus Toolkit                    | Globus Toolkit  | Globus Toolkit   | Globus Toolkit, GPE           | ARC Nordugrid, gLite, UNICORE, dCache |
| API для виконання базових grid-операцій | +<br>(неповний функціонал для desktop-застосувань) | +                                 | +   | +<br>(неповний функціонал)                                       | +<br>(складність API)         | +<br>(неповний функціонал)            |
| API для підтримки роботи ВО             | неповний функціонал                                |                                   |   |  |                               |                                       |
| API для роботи з grid-сховищами даних   | -  | -                                 | +   | -  | +                             | +/-<br>(неповний функціонал)          |
| API для запуску задач другого типу      | +<br>(Splitter Component)                          | -                                 | -   | -  | +<br>(складність API)         | +                                     |
| API для запуску задач третього типу     | -  | -                                 | -   | -  | -                             | -                                     |
| Розширюваність                          | +  | +                                 | +   | +  | +                             | +                                     |
| Зв'язок із іншими фреймворками          | -  | Java Community Grid (COG)         | Grid Portal Development Kit                             | COG, Globus API, GridSphere                                      | -                             | -                                     |
| Підтримка розробниками                  | на сьогодні підтримується                          | останній реліз від 25.02.02       | останній реліз від 21.09.2007                           | останній реліз від 31.07.2010                                    | останній реліз від 10.12.2007 | на сьогодні підтримується             |

За результатами досліджень окреслимо недоліки існуючих засобів розробки прикладних grid-застосувань:

- 1) відсутність інтеграції із різним проміжним програмним забезпеченням grid. Із зазначених, лише бібліотека Libarcclient спрямована на підтримку різного middleware;
- 2) недостатня реалізація базового функціоналу для роботи у grid-середовищі та підтримки роботи віртуальних організацій (управління користувачами та ресурсами ВО, управління доступом к ресурсам, облік часу використання та завантаження grid-ресурсів);
- 3) відсутність автоматизованого запуску для вирішення у grid-середовищі обчислювальних задач великої розмірності, яким властива конвеєрна схема організації обчислень;
- 4) відсутність зручного інтерфейсу для розробки спеціалізованих grid-сервісів ВО;
- 5) орієнтація на консольний режим роботи та використання скриптів. API для реалізації клієнтських застосунків з графічним інтерфейсом є не повнофункціональним;
- 6) часта зміна API, для більшості проектів відсутня підтримка на сьогодні.

Проведене дослідження підтверджує актуальність розробки фреймворку для створення грід-застосунків.

## **ФОРМУВАННЯ ВИМОГ ДО ФРЕЙМВОРКУ**

Оскільки інфраструктури віртуальних організацій, створених для різних галузей науки, є схожими, то є можливим окреслити базові вимоги до розроблюваного фреймворку для створення високорівневих grid-застосунків. Функціональні вимоги наведено нижче.

1. Надання API для виконання базових грід-операцій: створення проксі-сертифікату, перегляд доступних обчислювальних ресурсів та сховищ даних, запуск задачі, моніторинг виконання в динамічному режимі, отримання результатів та ін.
2. Автоматизована генерація файлу опису задачі згідно визначених користувачем вимог до ресурсів.
3. Підтримка наведених вище трьох типів задач з наданням графічних компонентів для визначення обчислювальних блоків задачі та схеми синхронізації між ними.
4. Надання API для підтримки роботи віртуальних організацій:
  - 1) управління користувачами віртуальної організації;
  - 2) управління ресурсами ВО: обмеження власником доступу до ресурсу; визначення власником тарифікації оплати за користування ресурсом; моніторинг стану ресурсів;
  - 3) білінгова система обліку використання ресурсів та нарахування оплати.
5. Надання API для розгортання нових грід-сервісів ВО.

Функціональні вимоги представлені нижче з точки зору користувачів грід-порталу віртуальних організацій (рис. 2 та рис. 3).

Окремо слід виділити нефункціональні вимоги до фреймворку.

1. Орієнтація на підтримку різного ППЗ (гнучкість реалізації для забезпечення можливості розширення). Підтримка ППЗ, що використовується в межах української національної грід-інфраструктури, а саме ARC Nordugrid та gLite.
2. Орієнтація як на desktop, так і web-застосування.
3. Кросплатформеність.
4. Розширюваність.
5. Масштабованість.
6. Забезпечення безпеки грід-порталу та ресурсів ВО.

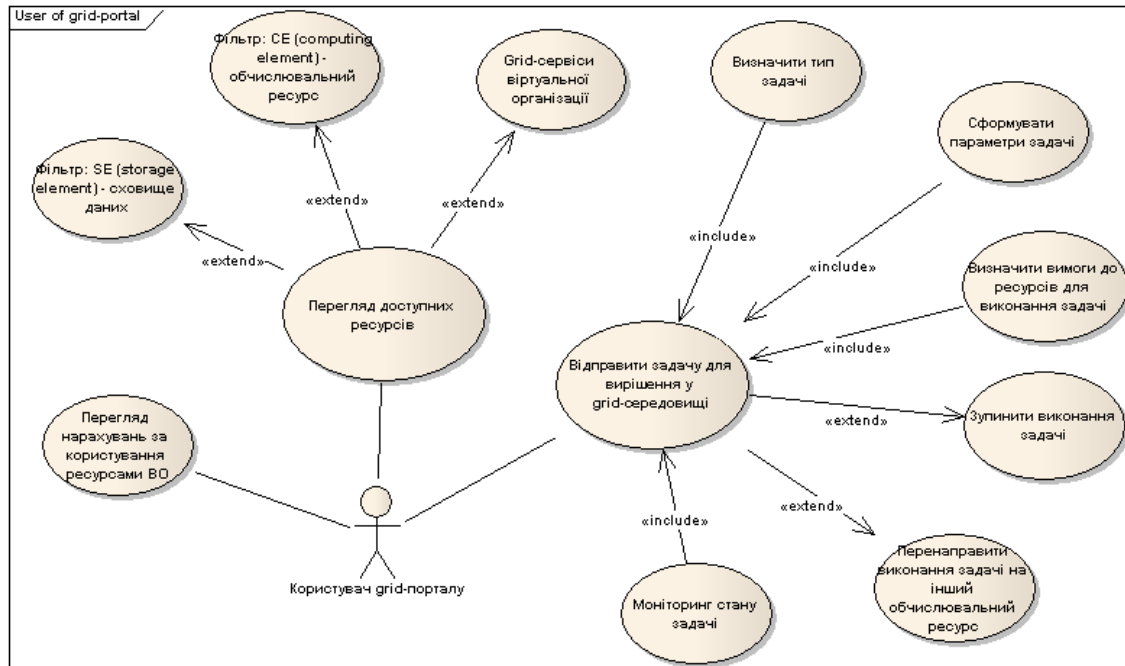


Рис. 2. Діаграма варіантів застосування грід-порталу користувачем

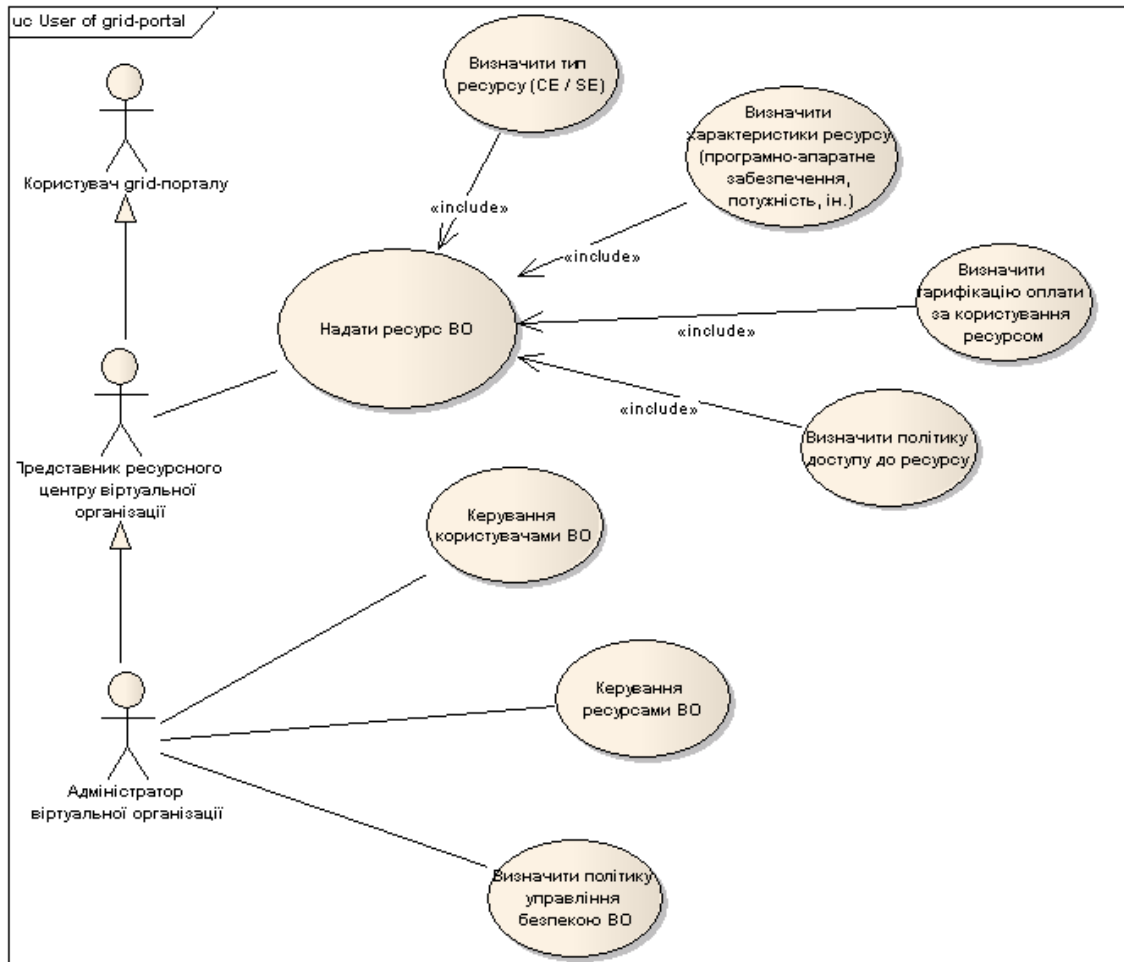


Рис. 3. Діаграма варіантів застосування grid-порталу адміністратором віртуальної організації та представником ресурсного центру

## АРХІТЕКТУРА ФРЕЙМВОРКУ ДЛЯ СТВОРЕННЯ GRID-ЗАСТОСУВАНЬ

Фреймворк розробляється на базі бібліотеки Libarcclient згідно наведених вище функціональних та нефункціональних вимог.

Архітектура фреймворку представлена нижче (рис. 4).



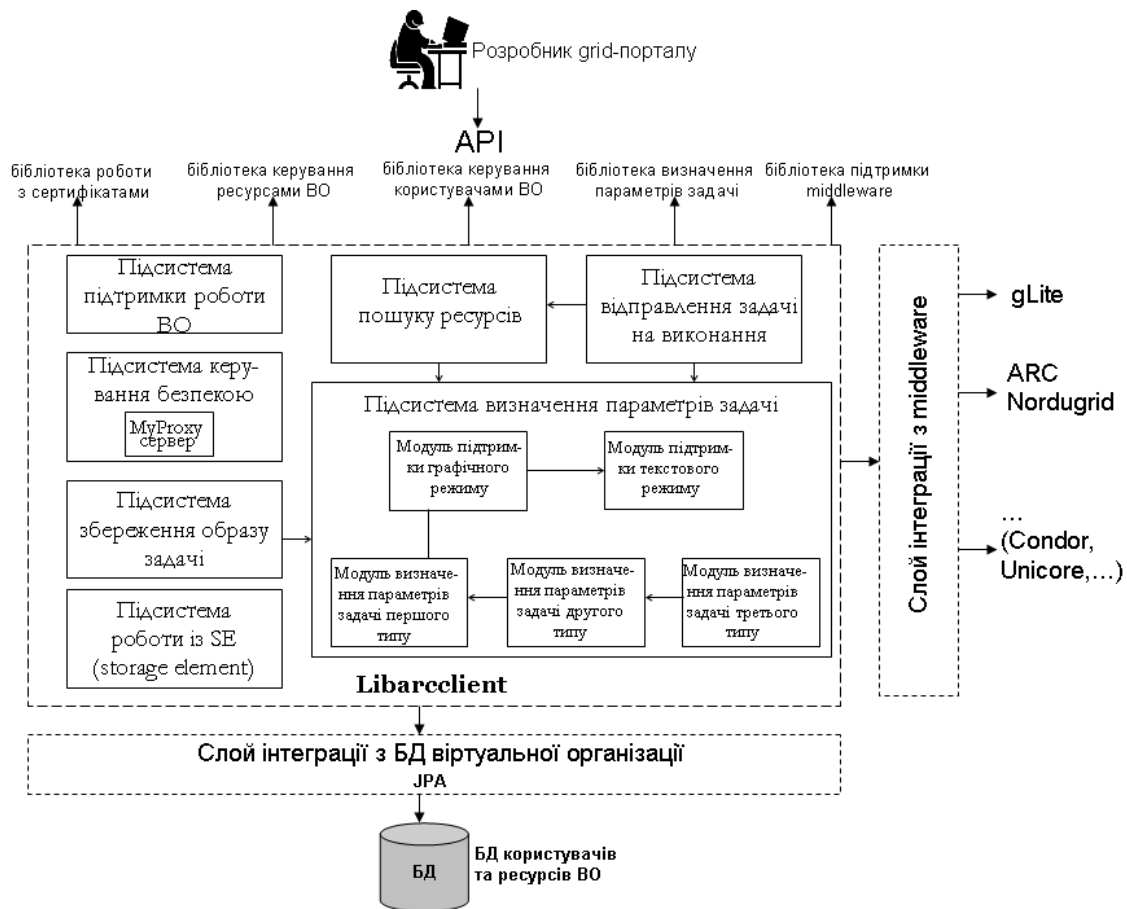


Рис. 4. Архітектура фреймворку для створення grid-застосунків

Використання grid-порталу для вирішення прикладних задач у grid-середовищі дозволяє значно знизити вимоги до програмного забезпечення клієнтської машини: достатнім є лише наявність встановленого веб-браузера. Тому фреймворк орієнтується на підтримку як desktop-застосунків, так і web-застосунків.

В межах підсистеми підтримки роботи BO реалізовано засоби керування користувачами та ресурсами BO, політикою доступу до ресурсів, обліку часу використання ресурсів та обчислення нарахунків за користування ресурсами. Для зберігання облікових даних користувачів, аутентифікації та авторизації використовується LDAP-сервер з підтримкою SSL-протоколу.

Для моніторингу стану ресурсів використовується відкрите рішення Ganglia Monitoring System – розподілена система моніторингу кластерів паралельних та розподілених обчислень, що надає інструменти для збору та перегляду статистики обчислень у режимі реального часу для кожної з машин [11].

**Підсистема керування безпекою.** У випадку grid-порталів нетривіальним є питання щодо створення проксі-сертифікату, оскільки на сьогодні браузери не надають підтримки проксі-сертифікатів [12]. Існує декілька підходів щодо делегування прав при запуску користувачем задачі на обчислювальний грід-ресурс.

1. На сервері віртуальної організації ведеться журнал запуску задач від імені користувача, однак на грід-ресурс задача запускається із сертифікатом сервера (шляхом створення проксі-сертифікату сервера). Така схема є простою в плані реалізації, однак не надає власнику ресурсу детальної інформації з приводу

використання ресурсу (який саме користувач grid-інфраструктури використовує ресурс). Можливість застосування такої схеми визначається домовленістю учасників віртуальної організації (насамперед власників ресурсів).

2. Проксі-сертифікат для запуску задачі створюється на клієнтській стороні (реалізація у вигляді java-applet). Недоліком використання такого підходу є необхідність установки додаткових java-плагінів та додаткових налаштувань браузера з метою відкриття доступу до локальних даних клієнтської машини.
3. Сертифікати користувачів зберігаються у репозиторії на сервері MyProху [13] та періодично оновлюються. Проксі-сертифікат для запуску задачі створюється на стороні сервера засобами сервісів MyProху.

На даний момент фреймворк реалізує перший варіант створення проксі-сертифікатів, планується також реалізація підтримки останніх двох з наданням адміністратору ВО можливості вибору схеми делегування прав при запуску задачі.

Для використання довгострокових проксі-сертифікатів, що є необхідним при запуску задач із значним терміном виконання, використовуються також сервіси MyProху.

Для забезпечення гнучкості розширювання системи з метою підтримки різного ППЗ було прийнято архітектурне рішення щодо винесення *слою інтеграції з middleware* в окремий незалежний слой. На даний момент фреймворк реалізує підтримку лише ARC Nordugrid 11.05, в подальшому планується підтримка gLite. Для забезпечення підтримки нового middleware мають бути реалізовані певні інтерфейси з метою визначення правил формування формату специфікації задачі. При цьому вносити зміни в решту фреймворку непотрібно, за рахунок чого процес реалізації підтримки нового middleware значно спрощується.

В межах *підсистеми визначення параметрів задачі* забезпечується підтримка трьох типів задач шляхом реалізації засобів визначення workflow задачі з метою виділення послідовних та паралельних блоків обчислень, потоків даних та схеми синхронізації обчислень між блоками. Користувачеві надається можливість визначення workflow задачі як у графічному, та і у текстовому режимі.

При запуску задачі з конвеєрною схемою обчислень система надає механізм оптимального вибору ресурсів з урахуванням параметрів мережі з метою мінімізації витрат на передачу даних між обчислювальними блоками з одного грід-вузла на інший. Якщо час на пересилку даних на інший кластер наближається до часу виконання підзадачі, то система запропонує варіант виконання двох підзадач на одному кластері.

Механізми оптимізації вибору ресурсів реалізовані у межах *підсистеми пошуку ресурсів*. Окрім цього, підсистема надає стандартні засоби вибору grid-ресурсів згідно визначених користувачем параметрів.

*Підсистема відправлення задачі на виконання* забезпечує автоматизований запуск визначених користувачем обчислювальних блоків задачі на обрані ресурси. Оскільки формат файлу опису задачі відрізняється для різного ППЗ, то в межах підсистеми реалізовано засоби автоматичного вибору створюваного формату специфікації задачі в залежності від параметрів ресурсів, куди відправляється задача для виконання. Підсистема надає також засоби моніторингу стану виконання задачі в динамічному режимі.

*Підсистема збереження образу задачі* надає засоби для збереження визначених користувачем або розробником параметрів та налаштувань прикладної задачі (тип задачі, обчислювальні блоки, схема синхронізації) для повторних запусків без налаштувань або з метою надання іншим користувачам ВО можливості запуску підготовленої та

адаптованої до grid-середовища задачі також без проведення повторних налаштувань. Збережений образ задачі може бути представлений у вигляді grid-сервісу VO.

Підсистема роботи із grid-сховищами даних (SE) надає інструментарій для пошуку відповідних ресурсів, відправки та отримання даних на/з ресурсу.

Фреймворк надає розробнику grid-застосувань набір API у вигляді java-бібліотек.

## ПРИКЛАД ВИКОНАННЯ БАЗОВИХ GRID-ОПЕРАЦІЙ З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ

На базі проміжної версії фреймворку було розроблено desktop-застосування, яке надає можливість відправки задачі для виконання на обчислювальний кластер, пошуку ресурсів, моніторингу стану задачі та отримання результатів виконання.

На рис. 5-9 наведено приклади використання розробленого grid-застосування.

The screenshot shows a desktop application window titled 'Открыть'. It has two main panes. The left pane contains a 'Выполняемый файл' (Executable file) field with '/bin/echo', a 'Задание' (Task) field with 'Hello World', and a 'Список кластеров' (Cluster list) with several entries like 'jade-cms.hip.fi', 'master.ri.kharkov.ua', etc. Below the list are buttons: 'Создать XRSL', 'Выполнить', 'Формируем набор кластеров', 'Проверить состояние задачи', 'Добавить кластер', 'Загрузить результат', 'Удалить кластер', and 'Вывести результат на экран'. The right pane has input fields for 'Свободные процессоры' (10), 'Вычислительные узлы' (10), and 'Свободное место' (20000000000).

Рис. 5. Генерация списка доступных кластерів

The screenshot shows a desktop application window titled 'Идентификатор задачи' (Task identifier). It contains several input fields: 'Идентификатор задачи' (hello), 'Выходной файл (результат)' (out.txt), 'Выходной файл (ошибки)' (err.txt), 'Выполняемый файл' (hello.sh), 'Максимальное время выполнения' (5), 'Свободное место на диске' (500), and 'Количество вычислительных узлов' (5). There is a checkbox for 'Кэш' (checked) and a button 'Создать образ файла' (Create file image).

Рис. 6. Визначення параметрів задачі

Результат формирования списка кластерів, что удовлетворяют указанным параметрам наведено на рис. 5.

На рис. 6 приведена форма назначения параметров задачи. Кнопка «выполняемый файл» позволяет указать путь до выполняемого файла задачи у файловой системе пользователя.

После отправки задачи на выполнение у нижнему полю окна программы будет назначено идентификатор, который получила задача. На рис. 7 изображено уведомление про стан задачи, что отвечает ситуации, когда задача находится у черзі на выполнения. Через деякий час стан задачи змінився: виконання задачі закінчилось успішно (рис. 8).

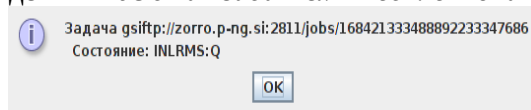


Рис. 7. Задача находится в черзі на выполнения

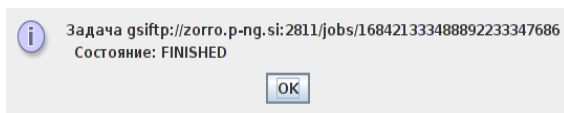


Рис. 8. Виконання задачі завершилось успішно

Після успішного виконання задачі, можна завантажити результат виконання (кнопки «Завантажити результат» та «Відобразити результат на екран»). Приклад отримання результату виконання тестової задачі Hello World зображено на рис. 9.

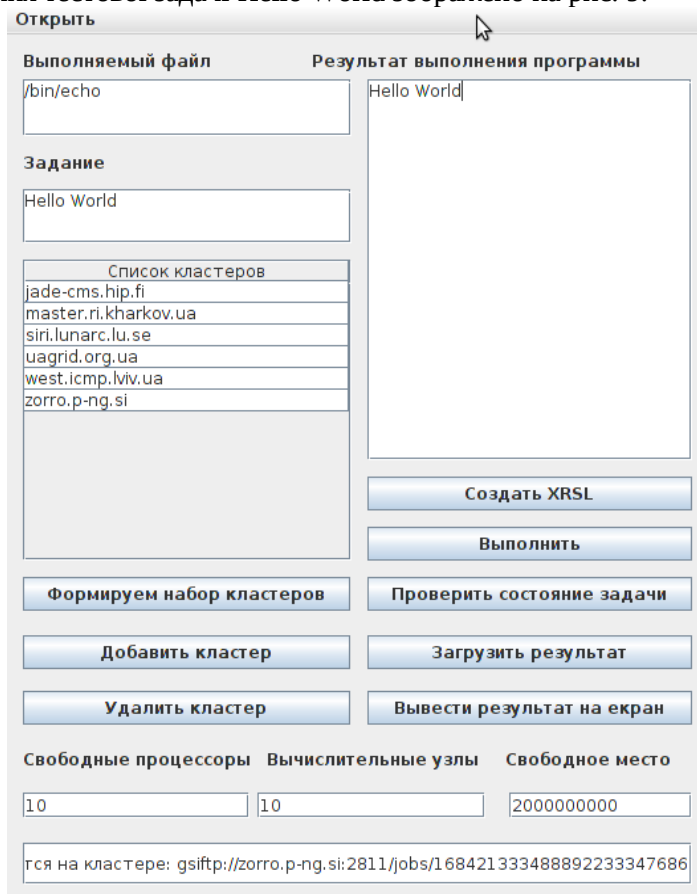


Рис. 9. Відображення результату виконання задачі

На базі фреймворку було розроблено також web-застосування (grid-портал) для виконання базових grid-операцій. Web-застосування використовує базу даних для збереження даних користувачів (сертифікати, налаштування), ведення журналу запуску задач, збереження образу задачі та ін..

На рис. 10 зображено приклад реєстрації користувача на grid-порталі.

Рис. 10. Реєстрація користувача на grid-порталі

При запуску задачі користувач може створити свій файл задачі або завантажити збережений образ задачі (рис. 11).

Рис. 11. Управління файлом опису задачі

Після формування параметрів задачі, так само як і у випадку desktop-застосування, формується список доступних кластерів для виконання задачі. Після відправлення задачі на виконання користувач може переглянути список та характеристики власних задач (рис. 12).

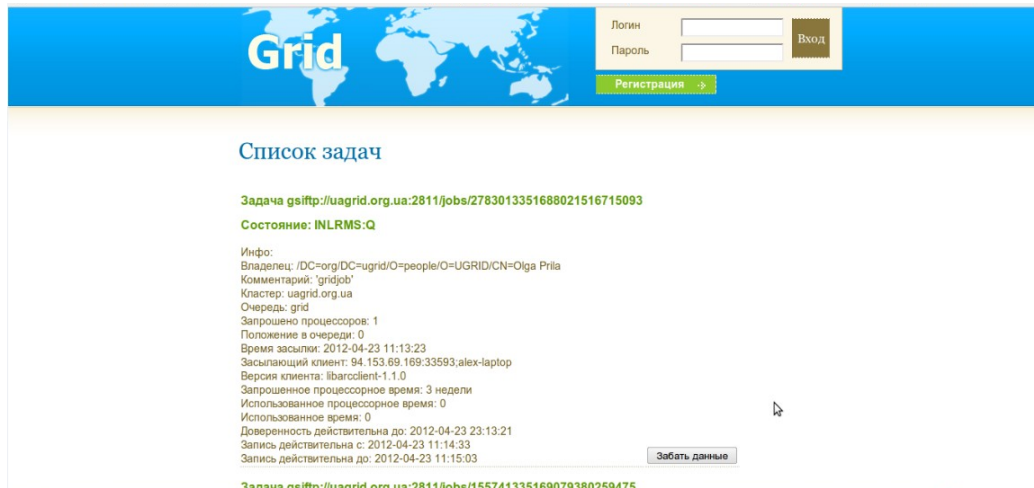


Рис.12. Список власних задач користувача

## РОЗРОБКА ЗАСОБІВ РОЗГОРТАННЯ GRID-СЕРВІСІВ З ОБРОБКИ ДАНИХ ВІДДАЛЕНИХ ОРГАНІЗАЦІЙ В УМОВАХ ПОЛІТИКИ НЕПОВНОЇ ДОВІРИ

В межах фреймворку планується реалізація засобів розгортання грід-сервісів вирішення задач, яким властива наступна схема організації обчислень: використання для обчислень статистичних (або інших) даних віддалених підприємств (що часто є конкуруючими) без відкриття прямого доступу до інформації. Зазначена схема організації обчислень характерна для ВО, які характеризуються політикою неповної довіри між об'єднуваними організаціями. Тобто окремою задачею фреймворку є надання API для створення такого виду грід-ресурсів як сховищ даних, що можуть використовуватись для виконання обчислень іншими учасниками ВО, однак лишаються закритими від прямого доступу.

Приклади подібних задач – задачі аналізу економічних ризиків підприємств та економічного прогнозування; вирішення медичних задач; обробка екологічних даних регіонів та ін.

Зазначена методика організації обчислень на прикладі задачі аналізу економічних ризиків банківських структур приведена нижче.

Статистичні методи побудови моделі оцінки міри ризику потребують наявності достатніх вибірок статистичних даних, оскільки точність оцінки параметра прямо пропорційна обсягу вибірки. Але існує коло задач області ризик-менеджменту банківської діяльності, дуже залежних від регіонального або інших критеріїв, які на даних одного банку не можуть бути вирішені навіть з мінімально-достатньою точністю оцінки.

Об'єднання статистичних даних банків дозволить значно підвищити точність обчислень. Однак пряме об'єднання даних неможливе через політику неповної довіри між організаціями. Для подібної схеми організації безпеки пропонується зазначена нижче методика організації обчислень у grid-середовищі.

Всю сукупність статистичних даних, що являє собою простір несумісних подій, можна поділити на  $n$  вибірок і проводити кореляційний аналіз кожної з цих вибірок паралельно на  $n$  обчислювальних вузлах з наступним розрахуванням результуючого рівняння регресії на головному вузлі. Можливість застосування такого механізму

розподілення обчислень забезпечується виконанням наступних обмежень: обсяг кожної вибірки та загальний обсяг статистичних даних достатньо великий; кожний ОВ повинен надавати спроможні й незміщені оцінки; функція розподілу величини, оцінку якої шукаємо, однакова для всіх вибірок у межах певної похибки. Останнє обмеження виконується за рахунок того, що вибірка однорідна, більше того, досить великий обсяг вибірки дозволяє вважати, що вибірковий розподіл прагне до істинного, а істинним для більшості процесів є нормальний розподіл [5].

Ефективність оцінки прямо пропорційна обсягу вибірки [5], отже, для об'єднання оцінок, отриманих по різних фрагментах вибірки з генеральної сукупності, необхідно використовувати вагові коефіцієнти, які розраховуються по формулі:

$$\mu_i = \frac{N_i}{\sum_{i=1}^n N_i}, \quad (1)$$

де  $\mu_i$  – вага оцінки, отриманої на  $i$ -тому вузлі;

$N_i$  – обсяг вибірки на  $i$ -ому вузлі;

$n$  – число обчислювальних вузлів.

Така методика організації обчислень полягає у виконанні локальних обчислень на даних окремих підприємств з отриманням проміжних оцінок та подальшому об'єднанні результатів. Фреймворк надає API для налаштування користувачем алгоритмів проміжних обчислень та об'єднання результатів.

## ВИСНОВКИ

Фреймворк для розробки грід-порталів розроблюється у межах відкритого проекту. Результат буде представлений у вигляді java-бібліотеки та прикладу застосування фреймворку для створення грід-порталу віртуальної організації економічного прогнозування. Застосування фреймворку дозволить зменшити витрати при розробці грід-порталів та вирішити проблему складності використання грід-середовища прикладними користувачами.

## ЛІТЕРАТУРА

1. gLite - Lightweight Middleware for Grid Computing [Електронний ресурс]. – Режим доступу: <http://glite.cern.ch>.
2. Nordugrid. Advanced Resource Connector [Електронний ресурс]. – Режим доступу: <http://www.nordugrid.org/arc>.
3. ARC Грід-монитор [Електронний ресурс]. – Режим доступу: <http://www.nordugrid.org/monitor/loadmon.php?display=vo=Ukraine>.
4. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2004. – 608 с.
5. Розанов Ю.А. Теория вероятностей, случайные процессы и математическая статистика. – М.: Наука, 1989. – 320 с.
6. Ganga. Simplifying use of the Grid [Електронний ресурс]. - Режим доступу: <http://ganga.web.cern.ch/ganga>.

7. DOE Science Grid: Grid Portal Development Kit (GPDК) [Электронный ресурс]. - Режим доступа: <http://doesciencegrid.org/projects/GPDК>.
8. Install GridSphere Portal Framework [Электронный ресурс]. - Режим доступа: [http://technical.bestgrid.org/index.php/Install\\_GridSphere\\_Portal\\_Framework](http://technical.bestgrid.org/index.php/Install_GridSphere_Portal_Framework).
9. Projects:simplegrid:index [CyberInfrastructure and Geospatial Information Laboratory] [Электронный ресурс]. - Режим доступа: <https://www.cigi.uiuc.edu/doku.php/projects/simplegrid/index>.
10. GPE4GTK project [Электронный ресурс]. – Режим доступа: <http://gpe4gtk.sourceforge.net/>.
11. [Ganglia Monitoring System](http://ganglia.sourceforge.net) [Электронный ресурс]. – Режим доступа: <http://ganglia.sourceforge.net>.
12. GridPP – UK Computing for Particle Physics [Электронный ресурс]. – Режим доступа: <http://www.gridpp.ac.uk/deployment/users/certhints.html#proxies>.
13. MyProxy Credential Management Service [Электронный ресурс]. – Режим доступа: <http://grid.ncsa.illinois.edu/myproxy>.