

О. Лебедєва, М. Лобур, О. Матвійків
Національний університет “Львівська політехніка”,
кафедра систем автоматизованого проектування

СИСТЕМА ПІДТРИМКИ КОЛЕКТИВНОЇ РОБОТИ НА ОСНОВІ GRID-ТЕХНОЛОГІЙ

© Лебедєва О., Лобур М., Матвійків О., 2007

Розглянуто аспекти створення системи підтримки спільної колективної роботи при виконанні складних проектів, запропоновано ієрархічні рівні розподіленої САПР на основі GRID-технології.

The system support creation aspects of collaborative work are examined at project implementation in the article. The hierarchical levels of distributed CAD are offered on the basis of GRID-technologies.

Вступ. Зазвичай при виконанні складних та масштабних проектів виникає необхідність взаємодії наукових колективів та установ різної спеціалізації. Подібна розподілена робота потребує створення єдиного інформаційного простору, в межах якого можна забезпечити ефективне спілкування учасників, обмін та узгодження результатів проектування та вільний доступ до інформації. Водночас поточна інформація про розробку повинна бути закритою для сторонніх осіб та захищеною від несанкціонованого доступу.

Потужним механізмом створення інформаційного простору для розподіленого проектування складних систем є GRID-технологія (Global Resource and Information Database). Вона забезпечує створення розподіленої проектною установи, надає можливість спільного використання обчислювальних та інформаційних ресурсів та дозволяє окреслити простір (закритий для доступу ззовні) для спілкування і спільного розв'язання різних проектних задач. GRID-технологія забезпечує можливість масштабування розподіленої системи, функціонування в гетерогенному середовищі та прозорість для користувачів.

У статті розглянуто аспекти створення розподіленої проектною групи для спільного виконання проектів територіально розділеними колективами і організаціями. Реалізація такого єдиного інформаційного середовища значно підвищить ефективність розподіленої розробки та інформованість учасників проекту.

Опис GRID-технології. Формально авторами концепції GRID вважають Яна Фостера з Чиказького університету і Карла Кессельмана з Університету Південної Каліфорнії. Саме вони 1998 року вперше запропонували термін GRID-комп'ютинг для позначення універсальної програмно-апаратної інфраструктури, що об'єднує комп'ютери та суперкомп'ютери в територіально-розподілену інформаційно-обчислювальну систему. За їхнім визначенням, “GRID – узгоджене, відкрите й стандартизоване середовище, яке забезпечує гнучкий, безпечний, скоординований розподіл ресурсів у рамках віртуальної організації”.

GRID – програмне забезпечення середнього рівня, що дозволяє інтегрувати розподілені та різномірні інформаційні та обчислювальні ресурси [1]. GRID є проміжною ланкою між користувачем та апаратним і програмним забезпеченням. Важливою властивістю GRID-технології є можливість авторизації доступу до мережі, що дає змогу реалізовувати на цій базі розподілені проектні наукові лабораторії і організації та створювати високоефективні системи дистанційного навчання.

Робота GRID основана на використанні Web-служб, що робить GRID-систему легко масштабованою, прозорою для використання та забезпечити можливість функціонування у

гетерогенному середовищі. Сьогодні така технологія активно розвивається, вважається найперспективнішою для розв'язання комплексних задач та набуває все більшого поширення.

Технологія управління розподіленими ресурсами – одне з найважливіших завдань. Вона спрямована насамперед на забезпечення керуваності інформаційної інфраструктури в умовах зростання навантаження і збільшення кількості компонентів мережі. Принципи роботи системи управління завданнями добре відомі: це черга, пошук вільних ресурсів, диспетчеризація, політики та пріоритети. Мережева система управління завданнями була реалізована досить давно, однак застосування технологій GRID дає змогу будувати систему управління розподіленими обчислювальними ресурсами. У такій ситуації користувачу вже не важливо, на якому конкретному вузлі мережі виконується його завдання; він просто споживає певну кількість віртуальної процесорної потужності, наявної в мережі.

Є декілька причин, які спонукають вчених використовувати GRID-технології.

1. Необхідність обробити величезну кількість даних, які зберігаються в різних організаціях;
2. Необхідність під час досліджень виконати величезну кількість обчислень;
3. Бажання наукової команди, члени якої працюють у різних куточках земної кулі, спільно використовувати великі масиви даних, швидко й інтерактивно їх комплексно аналізувати, візуалізувати та обговорювати результати в онлайн-режимі.

Важливу роль в GRID технологіях відіграє концепція віртуалізації, яка підтримує узгоджений доступ до ресурсів на множині гетерогенних платформ, дає змогу визначити відображення множини логічних екземплярів ресурсів на один і той самий фізичний рівень та допомагає управляти ресурсами в віртуальних організаціях, забезпечує композицію з ресурсів нижчого рівня, об'єднує базові сервіси, дає змогу формувати складніші сервіси. Віртуалізацією GRID-сервісів можна легко відображати загальну семантичну поведінку сервісів на оригінальні механізми платформи. Своєю чергою, віртуальні проектні організації є формою об'єднання ресурсів та користувачів, орієнтованих на виконання певних задач та надання певних сервісів.

Серед основних чинників, котрі сприяють впровадженню GRID до систем розподіленого проектування, треба зазначити підвищення ефективності використання наявних обчислювальних та людських ресурсів, оскільки, спільно працюючи над різними проектами, фахівці використовують одну й ту саму інфраструктуру [3].

Відомо, що існує декілька типів GRID-систем. Всі ці системи надають ресурси за допомогою сервісів, але кожна з цих GRID-систем оптимізована для надання різної функціональності. При цьому розрізняють:

- **Обчислювальні GRID** – для об'єднання обчислювальної потужності тисяч персональних комп'ютерів та серверів і створення так середовища з суперкомп'ютерним рівнем продуктивності;
- **GRID даних** – надають ресурси для розподіленої обробки даних, що дає змогу виконувати глибокий аналіз розподілених, більших за обсягом та часто різнотипних баз даних без їхнього фізичного переміщення;
- **Семантичні GRID** – надають інфраструктуру для виконання обчислювальних задач на основі розподіленого мета-інформаційного оточення, що дає змогу оперувати даними з різнотипних баз даних, різноманітних форматів, надаючи результат у форматі, що визначається додатком.

Прикладами GRID-систем є Information PowerGRID (створена для розподіленої інтелектуальної обробки складних космічних даних), DataGRID (розроблялася для розв'язання комплексних задач екологічного моніторингу та моделювання фізичних і біологічних процесів у космосі) та TerraGRID (створена для розв'язання складних наукових задач на основі паралельних обчислень) [4].

Вимоги до архітектури системи. У межах розподіленої проектної організації необхідно забезпечити відкритість спілкування учасників проекту та закритість інформації для доступу ззовні. Зазвичай розробка систем підтримки колективної проектної роботи потребує розв'язання таких основних задач:

- Створення служби сертифікації ключів для авторизованого доступу до системи та забезпечення їх поширення між учасниками;
- Створення служби для інтерактивного обміну інформацією;
- Створення служби контролю версій документації з проекту, що виконується.

Розглянемо основні вимоги до реалізації різних рівнів розподіленої САПР, а також оцінимо ступінь відповідності цим вимогам технологій GRID, що використовуються для побудови інфраструктури проектних сервісів (middleware):

- **Об'єднання різноманітних апаратних засобів САПР до єдиної інфраструктури**

Інфраструктура та технології GRID створювалися саме для сумісного використання ресурсів в динамічних розподілених проектних організаціях з метою ефективного використання наявних ресурсів в процесі вирішення будь-яких складних завдань. Проте основною є наявність швидкісних і надійних комунікаційних каналів.

- **Масштабованість, що динамічно надає обчислювальні потужності для вирішення поставлених завдань з врахуванням граничних умов**

Можливість динамічної конфігурації "комп'ютерного елемента" дає змогу використовувати рівно стільки ресурсів, скільки необхідно для вирішення конкретного завдання з врахуванням додаткових умов, а також динамічно перерозподіляти ресурси, звільнити або задіяти за необхідності додаткові обчислювальні потужності.

- **Забезпечення надійності і відмовостійкості**

Технології GRID підтримують створення динамічного середовища і можуть відстежувати стан завдання так, щоб у разі виходу з ладу одного або декількох вузлів в обчислювальному пулі можна було достатньо швидко заповнити його структуру за рахунок реконфігурації і введення додаткових вузлів або переміщення завдання на інший обчислювальний ресурс.

- **Забезпечення безпеки та конфіденційності даних**

Технологію GRID було розроблено для створення безпечного середовища з метою виконання різних за вимогами безпеки завдань. Відмінність інфраструктура безпеки GRID від мережевої системи безпеки, яка в основному забезпечує безпечні канали передачі даних між мережевими вузлами, що спілкуються, або комп'ютерами, в тому, що GRID-безпека повинна гарантуватися для завдання і даних, які можуть оброблятися на багатьох комп'ютерних вузлах і переміщатися від одного комп'ютера до іншого в процесі виконання. В GRID безпеці контекст безпеки повинен бути прив'язаний до самого завдання або даних і забезпечувати для них такі сервіси безпеки, як цілісність, конфіденційність, аутентифікація і авторизація. Всі ці сервіси і контекст безпеки не повинні порушуватися при переміщенні даних або завдань від однієї обчислювальної системи до іншої [7, 8].

- **Зберігання і обробка великих обсягів інформації**

Спеціальний тип GRID даних, які спеціалізуються на зберіганні, наданні доступу і обробці величезних об'ємів інформації в багатьох додатках не передбачають їхнє фізичне переміщення між обчислювальними ресурсами.

- **Гетерогенність, стандартне представлення даних обчислювальних ресурсів**

Обмін даними між різними компонентами системи відбувається з використанням стандартних протоколів веб-сервісів, побудованих на основі XML-протоколу або SOAP (Simple Object Access Protocol). Використання XML-технологій також дає змогу ефективно керувати семантичним простором імен і типів даних, які також можуть бути прив'язані до стандартного опису у форматі WSDL [5, 6].

Наведені вище результати аналізу дозволяють зробити висновок, що архітектура GRID надає всі необхідні технологічні компоненти для побудови розподілених, масштабованих, гетерогенних, надійних САПР, які охоплюють не тільки обчислювальний рівень, але й рівні, що забезпечують сумісне проектування географічно віддалених команд інженерів, що покриває всі види забезпечення САПР. Більш того, з використанням стандартних протоколів і технологій Веб-сервісів можна інтегрувати компоненти САПР в інші системи, які підтримують ці технології [4].

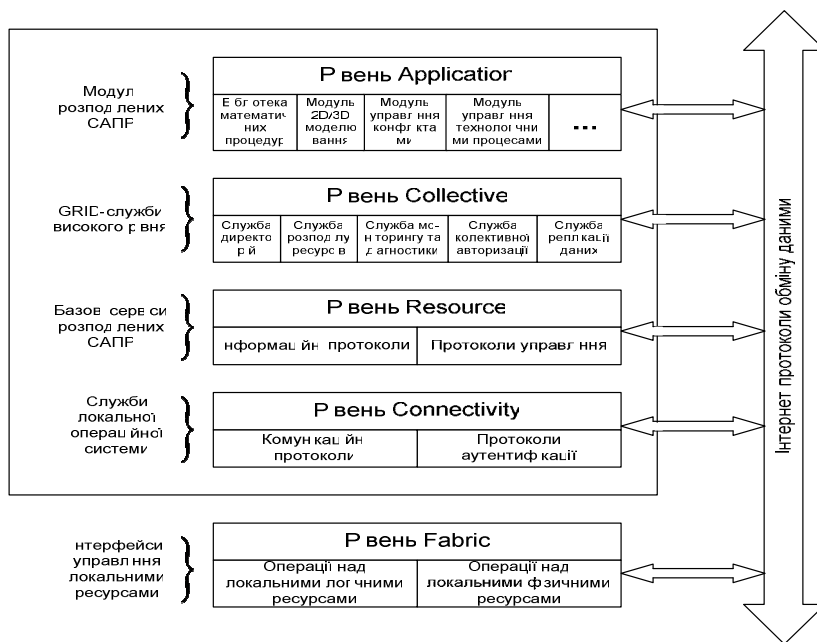
Як типова система підтримки спільної колективної роботи, запропонована GRID-система повинна містити такі програмні компоненти: реєстр організацій-учасників проекту, служби обміну повідомленнями між користувачами (чат, Інтернет-пейджер, форум), служба каталогів для доступу до документів, служба для пошуку в інформаційних ресурсах системи, служби контролю версій документів.

Система повинна забезпечувати відкритість спілкування між учасниками проекту, прозорий доступ до сховища інформаційних ресурсів організацій-учасників, можливість доступу до обчислювальних ресурсів організацій-учасників та до існуючих інформаційно-обчислювальних додатків у вигляді GRID-служб. Система також повинна надавати Web-інтерфейс, доступний з будь-якого пристрою, що підтримує стандарти HTTP і HTML. Крім того, треба забезпечити можливість контролю версій документів для запобігання випадкової або зловмисної модифікації даних у системі, аутентифікацію і авторизацію користувачів перед доступом до ресурсів системи, безпечний спосіб передачі даних загальнодоступними каналами зв'язку для запобігання перехопленню інформації третіми особами.

У системі повинні використовуватися протоколи, що ґрунтуються на XML. Базові компоненти системи необхідно реалізовувати на основі технології Java для можливості їх використання незалежно від типу операційної системи. GRID-служби мають бути реалізовані згідно з специфікацією OGSA/OGSI. Інтерфейси GRID-служб повинні відповідати стандарту GWSDL; інтерфейс користувача має сенс реалізувати у вигляді Web-порталу з використанням технології портлетів.

При створенні єдиного інформаційного простору програмне забезпечення системи повинно відповідати таким вимогам: сервери програмних додатків мають забезпечувати функціонування контейнера для GRID-служб; сервери баз даних повинні забезпечувати функціонування СКБД; Web-сервер повинен забезпечувати функціонування Web-інтерфейсу.

GRID-архитектура насамперед повинна бути здатна встановлювати, управляти і використовувати відносини розподілу ресурсів між будь-якими потенційними учасниками спільного проекту. Центральною проблемою є забезпечення взаємодії між різними платформами, мовами і програмним середовищем. У мережевому середовищі взаємодія означає роботу за загальними протоколами. Саме протоколи регламентують взаємодію елементів розподіленої системи, а також структуру інформації, що передається. Отже, GRID-архитектура – це архітектура протоколів, які повинні визначати базовий механізм взаємодії. На основі стандартних протоколів можуть будуватися стандартні сервіси, розроблятися інтерфейси прикладного програмування (API) та інструментальні засоби розробки (Software Development Kits - SDK). Технологія і архітектура утворюють служби, необхідні для підтримки загального набору додатків в розподіленому мережевому середовищі [3].



Ієрархічні рівні розподіленої САПР на базі GRID-технології

Ієрархічні рівні розподіленої САПР на базі GRID-технології наведено на рисунку. Деталізуючи цей рисунок, архітектуру GRID можна представити у вигляді ієрархічної структури, що складається з декількох рівнів [2]. На кожному з наведених рівнів існують свої сервіси, що взаємодіють за допомогою певних протоколів.

Ядром багаторівневої моделі є протоколи Resource і Connectivity, на яких покладені функції забезпечення розподілу індивідуальних ресурсів. Рівень Collective відповідає за координацію використання наявних ресурсів, доступ до них здійснюється за допомогою протоколів Fabric. Опишемо тепер детальніше функції кожного з рівнів:

- **Fabric: управління локальними ресурсами**

Рівень Fabric забезпечує доступ до ресурсів, що розподіляються, який опосередкований протоколами вищого рівня. При цьому ресурс може бути логічною суттю, наприклад розподіленою файловою системою, або фізичною – кластером або пулом комп'ютерів. Реалізація такого ресурсу може містити внутрішні протоколи, проте подібні протоколи залишаються поза GRID-архітектурою. Компоненти рівня Fabric реалізують локальні, специфічні для кожного даного ресурсу (логічного або фізичного) операції. Як видно з рисунка, цей рівень представляє набір інтерфейсів для управління локальними ресурсами. Цей рівень є базовим при оперативній побудові великих інтегрованих систем.

- **Connectivity: легкість і безпека комунікацій**

Цей рівень є ядром комунікацій і містить протоколи аутентифікації, потрібні для виконання специфічних мережових транзакцій. Комунікаційні протоколи відповідають за обмін даними з рівнем Fabric. До їх функцій входять транспорт, маршрутизація і привласнення імен. Зазвичай використовується стек протоколів TCP/IP.

- **Resource: розділення одиничних ресурсів**

В ієрархії рівнів Resource розташовується над рівнем Connectivity і використовує ті, що входять в рівень Connectivity, комунікаційні та аутентифікаційні протоколи для узгодження використовуваних методів безпеки, ініціалізації, моніторингу і управління ресурсами. Для доступу і управління локальними ресурсами Resource викликає відповідні функції рівня Fabric. Протоколи рівня Resource призначені винятково для роботи з локальними ресурсами, не враховують глобальний стан системи і не відстежують атомарні операції, що виконуються об'єднаними ресурсами. Рівень містить 2 основні класи протоколів: *інформаційні протоколи*, призначені для отримання інформації про структуру і стан ресурсу, його конфігурацію, поточне завантаження тощо, та *протоколи управління*, що забезпечують узгодженість доступу до ресурсу, що розподіляється, а також визначають операції, які необхідно виконати.

- **Collective: координація ресурсів;**

Рівень Collective містить протоколи і сервіси, не пов'язані з будь-яким специфічним ресурсом. Це глобальні протоколи, які відповідають за взаємодію всіх елементів пулу ресурсів.

Деякі глобальні функції і сервіси, що реалізуються протоколами цього рівня, наведемо нижче:

- Служба директорій – дає користувачам змогу виявити ресурси і визначити їхні властивості, виконувати запити, використовуючи логічні імена та атрибути ресурсів;
- Розподіл ресурсів, планування і брокерські послуги – виділяє за запитом користувача необхідну кількість ресурсів для специфічного завдання і планує його виконання;
- Служба моніторингу та діагностики – відстежує несправності, перевантаження, спроби несанкціонованого доступу тощо;
- Служба реплікації даних – забезпечує управління пристроями зберігання, необхідними мережевими і обчислювальними ресурсами;
- Служба колективної авторизації – санкціонує доступ до керованих ресурсів.

- **Applications: рівень додатків**

Це вищий рівень GRID-архітектури. Він містить призначені для користувача додатки, які виконуються в середовищі об'єднаних ресурсів. В процесі свого виконання вони викликають служби протоколів, що пролягають нижче. На кожному з рівнів є цілком певні протоколи, що забезпечують доступ до необхідних служб [9, 10].

Висновки. Отже, для створення єдиного інформаційного середовища взаємодії розподілених проектних організацій при виконанні складних та масштабних проектів пропонується використовувати GRID-технологію, яка забезпечить різноманітні сервісні функції, пов'язані з обміном інформаційними ресурсами, працюватиме у гетерогенному середовищі, буде ліцензійно «чистою», легко масштабованою, з належним рівнем безпеки і може бути адаптована майже до будь-якого процесу колективної розробки. Використання такої системи дасть змогу значно підвищити ефективність розподіленої розробки та інформованість розробників.

1. Foster I., Kesselman C., Nick J.M., Tuecke S. *The Physiology of the GRID - An Open GRID Services Architecture for Distributed Systems Integration.* – <http://www.globus.org/research/papers/ogsa.pdf>. 2. I Foster, C Kesselman, S Tuecke, *The Anatomy of the GRID; Enabling Scalable Virtual Organisations*; 2001. 3. Куссиль Н.М., Лавренюк А.М., Шелестов А.Ю., Лобунець О.Г., Скакун С.В. *Розробка віртуального інформаційного середовища для спільного виконання космічних проектів // Проблеми програмування.* – 2006. – № 2–3. 4. С. В. Велічкєвич, А.І. Петренко. *Розподілена САПР з використанням технології GRID сервісів // Науковий Вісник НТУУ "Київський політехнічний інститут".* – 2004. – N3. – с.30–37. 5. P. Steinberg, S. Agarwal, G. Vorobiov. *Java and .NET Both Bring Something to the Party* – <http://www.devx.com/Intel/Article>. 6. *The Globus Toolkit.* - <http://www-unix.globus.org/toolkit>. 7. *Security in a Web Services World: A Proposed Architecture and Roadmap, Version 1.0, A joint security whitepaper from IBM Corporation and Microsoft Corporation, 2002.* – <http://www-106.ibm.com/developerworks/library/ws-secmap/> 8. Demchenko, Y., L. Gommans, C. de Laat, B.Oudenaarde, A. Tokmakoff, M. Snijders, R. Buuren. *Security Architecture for Open Collaborative Environment // European Grid Conference, Amsterdam, The Netherlands, Proceedings. Series: Lecture Notes in Computer Science, Volume 3470, 2005.* 9. Каменщиков М.А. *Сервисы GRID, как объекты стандартизации // ЖУРНАЛ РАДИОЭЛЕКТРОНИКИ.* – 2002. – N 12. 10. Бараш Л. *GRID Computing – новая парадигма Internet-вычислений // Компьютерное обозрение.* – 2001. – № 32.