УДК 004.75

Є.А. Слюсар

Автоматизована служба реплікації файлів для організації високої доступності даних у грід-інфраструктурі

Описана архитектура и реализация автоматизированной службы репликации данных в грид-инфраструктуре, обеспечивающая поддержку нескольких виртуальных организаций и различных политик репликации, а также прозрачность доступа к данным благодаря применению каталога данных *LFC* для хранения политик. Автономность работы обеспечивается путем получения делегации владельцев файлов через службу *MyProxy* и удостоверением членства в виртуальных организациях службой *VOMS*.

Automatic data replication service architecture supporting multiple Virtual Organizations and different replication policies is presented. The transparent data access is accomplished by employing the LFC data catalog service for storage of replication policies. A non-interactive batch operation is achieved by integrating the credential delegation mechanisms for actual data owners, which employ the MyProxy credential repository and the Virtual Organizations membership service.

Описано архітектуру та реалізацію автоматизованої служби реплікації даних у грід-інфраструктурі, що забезпечує підтримку декількох віртуальних організацій та політик реплікації, забезпечує прозорість доступу до даних шляхом використання каталогу даних *LFC* для зберігання політик реплікації. Автономність роботи забезпечується шляхом отримання делегації власників файлів через службу *MyProxy* з подальшим засвідченням участі у віртуальних організаціях службою *VOMS*.

Вступ. Грід-система – це поєднання технологій, інфраструктури та стандартів. Технології включають в себе спеціальне програмне забезпечення проміжного рівня, що дозволяє організаціям надавати свої інформаційні та обчислювальні ресурси у загальне користування; інфраструктура складається з апаратних засобів та служб на основі інформаційних та обчислювальних ресурсів, а стандарти визначають формат та протоколи обміну як для взаємодії між самими службами, так і між службами та користувачами [1].

Доступ до ресурсів грід-інфраструктури здійснюється через звернення до відповідних служб з використанням визначених протоколів. Реалізація грід-служби приховує від користувачів, інших служб інфраструктури внутрішню архітектуру і особливості функціонування ресурсу, що нею обслуговується, та надає уніфікований стандартизований інтерфейс згідно типу цього ресурсу [2].

Постановка задачі

Для віртуальних організацій (ВО), що оперують великими обсягами даних, актуальною є проблема забезпечення високої доступності даних для обчислень та оптимізації їх розміщення між елементами зберігання даних (*Storage Element – SE*) для підвищення швидкості доступу з найближчих обчислювальних елементів (*Computing Element – CE*). Пакет програмного забезпечення проміжного рівня *Nordugrid Advanced Resource Connector* (*ARC*) [3] не містить у своєму складі автоматизованих засобів реплікації даних. До пакету *gLite* входить служба *File Transfer Service* [4], яка є вузько спеціалізованою та організує доставку наборів даних із одного центрального елемента на декілька розподілених сховищ без взаємодії з каталогом даних. ВО змушені розробляти власні засоби реплікації даних, що відповідають застосованим ними методам обробки. Як наслідок, розроблені засоби також мають вузьку спеціалізацію і не можуть без істотних модифікацій використовуватись як універсальний засіб в межах грід-інфраструктури.

Наведемо вимоги до реалізації автоматизованої служби реплікації:

- масштабованість на велику кількість елементів зберігання даних;
- підтримка декількох ВО та різних політик реплікації для різних ВО;
- автономність роботи – керування репліками без втручання користувачів;
- прозорість доступу до даних та інтеграція зі службою каталогу файлів;
- робота від імені користувачів ВО за допомогою делегації;
- можливість як централізованого, так і місцевого розгортання.

Єдина реалізація служби реплікації даних, що відповідає вимогам різних ВО і сприятиме

розвитку грід-технологій у складі національних грід-інфраструктур, зокрема й української. Це дозволить учасникам ВО сконцентруватись на галузі своїх досліджень та позбавить їх необхідності вручну керувати великими обсягами розподілених даних у грід-інфраструктурі.

Архітектура служби

Розроблена служба реплікації складається з декількох компонентів, що взаємодіють між собою. Для звернення до інших служб грід-інфраструктури використовуються утиліти та бібліотеки пакетів програмного забезпечення проміжного рівня *gLite* та *Nordugrid ARC*. Оскільки політики реплікації застосовуються до логічних імен у каталозі файлів, то доцільним є використання спільної бази даних для зберігання логічної ієрархії імен, а також відповідних метаданих, до складу яких долучається опис політик реплікації. Реалізація служби каталогу даних *LCG File Catalog* (*LFC*) [5], що застосовується в українському грід-сегменті, а також у світових інфраструктурах *WLCG* та *EGI*, не містить вбудованих засобів для зберігання спеціалізованих метаданих. Проте, кожен іменований об'єкт у каталозі має атрибут *коментар* – текст вільної форми, який може змінюватись користувачем згідно політик контролю доступу *LFC* і не впливає на будь-які інші аспекти роботи каталогу. Однією з особливостей запропонованої архітектури є зберігання політик реплікації в закодованому та стисненому вигляді у складі коментаря файлу або директорії у каталозі *LFC*. Така організація сховища метаданих для служби реплікації має певні переваги:

- ***відсутність власної бази даних***: усі політики реплікації зберігаються безпосередньо як метадані логічних імен у каталозі файлів, а служба реплікації звертається до нього для читання або запису цих політик;
- ***стандартний протокол зовнішнього інтерфейсу служби***: взаємодія користувачів із службою відбувається через внесення змін до коментарів логічних імен у каталозі *LFC*; при цьому використовується стандартний протокол *LFC*, який передбачає автентифікацію та авторизацію користувачів;
- ***мобільність реалізації інтерфейсу користувача***: засоби інтерфейсу користувача базуються на програмних бібліотеках клієнта служби *LFC*, не мають ніяких інших суттєвих залежностей та можуть бути реалізованими як у вигляді *Shell*-сценарію, так і у формі самостійної утиліти.

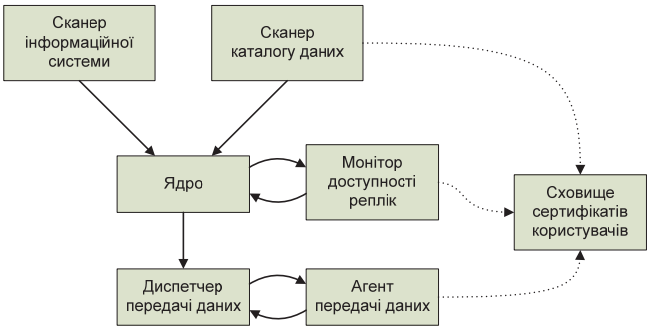

Рис.1. Взаємодія компонентів служби реплікації

Схема взаємодії компонентів служби реплікації показана на рис. 1. Ядро системи виконує зіставлення характеристик наявних у грід-інфраструктурі елементів зберігання даних та параметрів логічних імен у каталозі файлів, у тому числі кількість наявних реплік та політики реплікації, внаслідок чого формуються запити на перевірку доступності існуючих реплік та створення нових.

Сканер інформаційної системи – це компонент, що забезпечує ядро відомостями про наявні елементи зберігання даних та їх поточні характеристики. Один і той же елемент зберігання даних може підтримувати декілька ВО, а кожна ВО в свою чергу може мати декілька точок входу до сховища та резервації дискового простору (*Storage Space Reservations – SSR*) на них. Ці відомості, разом із обсягами резервацій та фактичними показниками вільного простору, публікуються у локальній інформаційній системі елемента зберігання даних. Опубліковані відомості з усіх грід-ресурсів передаються на центральний каталог ресурсів інфраструктури. Сканер інформаційної системи періодично опитує цей каталог та формує список усіх наявних сховищ даних та резервацій на кожну ВО. До списку входить також посилання на стандартний інтерфейс керування елементом зберігання даних *Storage Resource Manager* (*SRM*) [6], що підтримується більшістю поширених реалізацій грід-сховищ – *Disk Pool Manager* (*DPM*),

dCache, *Storage Resource Manager* (*StoRM*), *Berkeley Storage Manager* (*BeStMan*) та *CERN Advanced STORage manager* (*CASTOR*).

Сформований список зберігається у загальному кеші об'єктів служби реплікації, звідки його може отримати ядро системи. Сканер періодично поповнює кеш об'єктів новим списком на кожному циклі своєї роботи.

Сканер каталогу даних виконує обхід ієрархічної структури логічних імен у каталозі *LFC*, отримуючи метадані та список існуючих реплік на кожне ім'я, куди може входити опис політики реплікації. Оскільки один каталог *LFC* може обслуговувати декілька ВО, кожна з яких може встановлювати власні політики доступу до каталогу, звернення до нього відбувається з використанням проксі-сертифікатів учасників відповідних ВО. Для отримання таких сертифікатів, в процесі обходу каталогу, сканер звертається до сховища сертифікатів користувачів. Якщо у сховищі на момент запиту не зберігалось необхідного сертифіката, то відповідна гілка ієрархії імен пропускається, а на сховище направляється запит на отримання проксі-сертифіката. В результаті одного циклу обходу сканера формується список усіх логічних імен, що мають хоча б одну репліку та перебувають в полі дії хоча б одної з політик реплікації. Політики реплікації наслідуються від батьківського об'єкта вниз за ієрархією. Сформований список заноситься до загального кешу об'єктів служби реплікації та періодично оновлюється сканером.

Сховище сертифікатів користувачів слугує для зменшення навантаження на службу делегації *MyProxy* [7] та службу засвідчення участі у ВО (*Virtual Organization Membership Service*, *VOMS*) [8]. Агент сховища звертається до кешу об'єктів та отримує список необроблених запитів на отримання проксі-сертифіката. Використовуючи власний сертифікат служби, агент здійснює запити до служби *MyProxy* на отримання делегації користувачів, унікальні імена сертифікатів яких (*Distinguished Name – DN*) були вказані у необроблених запитах кешу. У разі успішного отримання делегації проксі-сертифікати завантажуються у кеш. На наступному етапі з використанням отриманих проксі-сертифікатів за допомогою служби *VOMS* генеруються нові, але вже зі спеціальним розширенням, що підтверджує участь користувача у відповідній ВО. Оптимізація полягає у зменшенні кількості запитів до служби *MyProxy* в тому випадку, коли один і той самий користувач є учасником декількох ВО. Результуючі проксі-сертифікати з *VOMS*-розширенням також завантажуються у кеш (рис. 2).

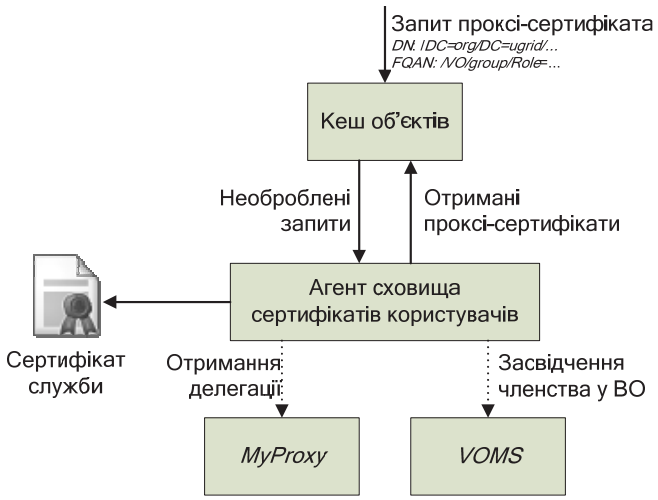

Рис. 2. Схема роботи сховища сертифікатів користувачів

Для успішної роботи описаного механізму необхідно, щоб користувачі служби реплікації завантажували на відповідний *MyProxy*-сервер довгострокові делегації та вказували у політиці делегування унікальне ім'я сертифіката служби. Це забезпечить отримання делегації та швидке реагування на зміну стану реплікації файлів у каталозі.

Сертифікат служби також використовується у випадках, коли неможливо визначити користувача-власника певного файлу чи директорії у каталозі файлів, а також при початковому скануванні для визначення піддерев, підтримуваних ВО. Якщо служба реплікації входить до складу ВО як учасник, то її сертифікат можна використовувати замість користувацького у всіх операціях, де дозволено доступ до файлів усім учасникам ВО. Це стосується загальних вхідних даних для подальшого аналізу учасниками ВО.

Ядро системи отримує список об'єктів та їх політики реплікації зі сканера каталогу даних. На першому кроці циклу відбувається перевірка кожного об'єкта на відповідність вказаним

політикам. Зокрема, перевіряється, щоб кількість реплік файла була у встановлених межах. Для зменшення навантаження на елементи зберігання даних перевірка доступності репліки відбувається за допомогою операції отримання списку файлів у відповідній директорії сховища, результати якої заносяться у загальний кеш. Спочатку перевіряється наявність у кеші відповідного елемента, і у випадку його відсутності генерується запит до монітору доступності реплік і ядро переходить до розгляду наступного об'єкта.

У випадку, коли результати перевірки доступності реплік файла були успішно отримані, визначається реальна кількість доступних реплік. На підставі цих даних приймається рішення про створення нової репліки у разі їх недостатньої кількості або про видалення зайвих реплік у разі їх надмірної кількості. Якщо кількість реплік залишається у діапазоні, вказаному в політиці, ядро переходить до розгляду наступного об'єкта.

Для створення нової репліки обирається резервація на елементі зберігання даних, де немає жодної репліки відповідного файла. Резервації є іменованими і можуть вказуватися у політиці реплікації. У випадку, коли резервацію не вказано, обирається простір за замовчуванням відповідного елемента зберігання даних. Отримані таким чином резервації впорядковуються за рейтингом, що обчислюється із показників надійності відповідного обчислювального елемента, що отримуються з системи тестування грід-інфраструктури або за внутрішніми результатами перевірки доступності служби реплікації, які формуються та поновлюються при перевірці доступності реплік. У політиці реплікації вказується порог відсічки – величина, що вказує з скількох елементів зберігання даних із верхівки рейтингу необхідно обрати місцезнаходження нової репліки. Кінцевий елемент обирається випадково.

На наступному кроці обирається існуюча репліка, з якої відбуватиметься копіювання вмісту файла для нової репліки. Також формується рейтинг серед існуючих реплік та обирається така пара елементів зберігання даних, для яких швидкість передачі була максимальною при попередніх операціях передачі даних, відомості про які також зберігаються у загальному кеші служби реплікації. Це забезпечить якнайшвидше створення нової репліки. Запит на передачу даних, сформований за описаними принципами, направляється до черги диспетчера передачі даних.

При видаленні репліки також будується рейтинг усіх використаних елементів зберігання даних, враховуючи показники надійності та вільного дискового простору. Із рейтингу обирається репліка з мінімальним рейтингом та формується відповідний запит на видалення даних та посилання на них із каталогу даних (рис. 3).

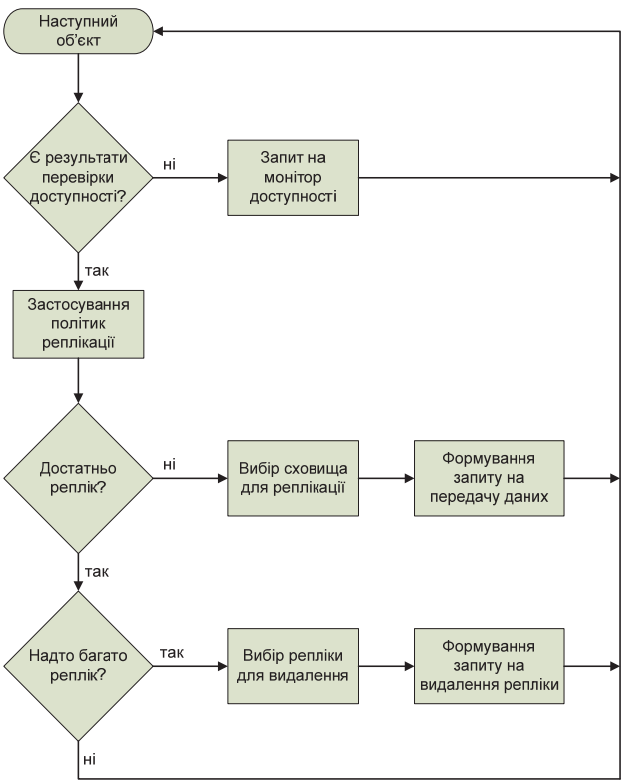

Рис.3. Схема роботи ядра служби реплікації

Монітор доступності реплік виконує перевірку наявності реальних даних за посиланнями, що вказуються як репліки файла. Для зниження навантаження на елементи зберігання даних запити консолідуються для кожного елемента шляхом отримання списку файлів певної директорії замість перевірки окремого файла. При автоматичному створенні реплік на елементах зберігання даних використовується проста дворівнева структура директорій, тому імовір-

ність того, що декілька файлів матимуть репліки в одній і тій самій директорії, досить висока. Списки файлів директорії заносяться до загального кешу на певний час, щоб виключити надто часті повторні запити. В результаті перевірки можливі наступні варіанти:

- необхідний файл наявний у списку файлів директорії – тоді поточна репліка вважається доступною;
- необхідний файл відсутній у списку – репліка вважається видаленою;
- елемент зберігання даних недоступний на момент запиту – репліка вважається недоступною.

Спроба безпосереднього отримання даних із репліки не виконується через те, що ці дані можуть зберігатися на стрічкових накопичувачах, і необхідний певний час та інші ресурси, щоб підготувати ці дані для завантаження. Якщо розмір файла досить суттєвий, то це може спричинити високе нецільове навантаження на елемент зберігання даних.

Перед зверненням до сховища, виконується спроба отримати проксі-сертифікат із сховища сертифікатів користувачів. Якщо сертифікат не знайдено, то залишається запит на генерацію такого сертифіката і логічне ім'я, репліки якого перевіряються у поточний момент, відкладається до наступного циклу роботи ядра.

Якщо файл відсутній у директорії на сховищі – це означає, що дані були видалені користувачем вручну або іншою службою грід-інфраструктури. Тому таку репліку необхідно видаляти із каталогу даних. Запит на видалення репліки формується автоматично та направляється безпосередньо до каталогу даних.

Якщо при зверненні до елемента зберігання даних виникла помилка, то не можна однозначно констатувати видалення репліки. Проте, можна констатувати тимчасову недоступність сховища в цілому. Успішність запитів до кожного сховища зберігається та поновлюється після кожного запиту у загальному кеші служби реплікації. Ці відомості у подальшому використовуються для обчислення рейтингу сховищ при виборі місцезнаходження нової репліки.

Диспетчер передачі даних застосовується для розподілу навантаження між елементами зберігання даних. Для цього використовується таблиця стану, де ведеться облік поточних сеансів обміну даними між елементами зберігання. Максимальна кількість одночасних сеансів читання та запису для одного сховища задається у глобальній конфігурації служби. Диспетчер вибирає з черги запит, який можна задовольнити у поточний момент, не порушуючи згаданих показників, та викликає агент передачі даних, який і обслуговує цей запит.

Агент передачі даних використовує стандартні засоби програмного забезпечення проміжного рівня для погодження тристоронньої передачі між елементами зберігання даних. Отже, дані передаються безпосередньо між двома сховищами, але процес передачі контролюється з вузла служби реплікації. Операція виконується від імені користувача-власника відповідних даних, проксі-сертифікат із необхідним *VOMS*-розширенням отримується з сховища сертифікатів користувачів. У разі невдачі при отриманні сертифіката або при налаштуванні передачі, лічильник спроб запиту збільшується і він залишається у черзі. При успішному створенні репліки вона реєструється у каталозі даних та запит видаляється з черги.

Загальний кеш об'єктів – це підсистема, яка забезпечує взаємодію усіх інших компонентів служби реплікації. Кеш являє собою довгострокове сховище типу «ключ-значення», об'єкти в якому мають обмежений час життя. Крім того, на кожен об'єкт вводиться лічильник звернень, і необхідною умовою для видалення об'єкта є рівність цього лічильника нулю. Замість об'єкта за заданим ключем може міститися запит на отримання об'єкта. Така схема дозволяє компонентам залишати запити на отримання об'єкта певного типу, а іншим – обробляти такі запити та замінювати їх сформованим об'єктом. Блокування та атомарний доступ до об'єктів забезпечується завдяки використанню стандартних примітивів синхронізації операційного середовища, таких як семафори та блокування читання-запису.

Підсистему кешу об'єктів реалізовано у вигляді окремого процесу, взаємодія з яким відбувається через стандартні засоби міжпроцесного зв'язку – загальну пам'ять та доменні гнізда *UNIX*. Для попередження втрати об'єктів при перезапуску служби, вони періодично із оперативної пам'яті записуються на диск. Система підтримує об'єкти різних типів та різні ключі відповідно до типів.

Реалізація та впровадження

Реалізація автоматизованої служби реплікації даних отримала назву *RAPTOR – Robot for Autonomous Precisely Tunable Operation of Replication*. Програмне забезпечення реалізовано мовою *C* та складається із наступних модулів:

- диспетчера кешу об'єктів (`raptor_ocs` – *Object Cache Service*);
- сканера інформаційної системи (`raptor_iss` – *Information System Scanner*);
- сканера каталогу даних (`raptor_fcc` – *File Catalog Crawler*);
- ядра системи (`raptor_combinator`);
- монітора доступності реплік (`raptor_ram` – *Replica Availability Monitor*);
- диспетчера та агента передачі даних (`raptor_tm` – *Transfer Manager*);
- сховища сертифікатів (`raptor_credman` – *Credential Manager*);
- утиліти адміністрування (`raptor_admin`).

Утиліта адміністрування дозволяє отримувати статистику роботи системи, а також здійснювати запуск та зупинку усіх модулів. Для доступу до інших служб грід-інфраструктури використано стандартні бібліотеки клієнтів засобів програмного забезпечення проміжного рівня, зокрема *LFC*, *MyProxy*, *VOMS* та *Grid File Access Layer* (*GFAL*) [9]. Усі компоненти, за виключенням утиліти адміністрування, працюють як фонові служби *UNIX* – демони, та взаємодіють між собою лише через диспетчер кешу об'єктів. Усі компоненти налаштовуються за допомогою загального файла конфігурації, що завантажується у кеш об'єктів при старті системи. До складу системи також входить бібліотека для роботи із двома форматами представлення політик реплікації – текстовим та бінарним. Текстовий формат можна застосовувати для встановлення політик реплікації безпосередньо через засоби інтерфейсу користувача служби каталогу файлів *LFC*.

Для більш зручного доступу до служби реплікації розроблено сценарій-обгортку мовою *Python*, що використовує стандартні клієнтські бібліотеки *LFC* та дозволяє встановлювати і декодувати опис політик реплікації, розміщений у полі коментаря до логічного імені файла чи директорії в каталозі *LFC*. Підтримуються наступні параметри політики:

- мінімальна кількість реплік;
- максимальна кількість реплік;
- спосіб вибору елементів зберігання даних – усі доступні для ВО, усі доступні з фіксованого списку, резервації за шаблоном, рейтингові коефіцієнти;
- атомарна реплікація каталогу або групи файлів за шаблоном імені;
- ознака наслідування/перевизначення.

Підтримується також надсилання сповіщення на вказану електронну адресу при зміні кількості реплік об'єкта.

Розроблена служба реплікації була впроваджена для обслуговування центрального каталогу даних української національної грід-інфраструктури, розміщеного в Інформаційно-обчислювальному центрі Київського національного університету імені Тараса Шевченка. Зокрема, службу реплікації інтегровано до віртуальної організації *MolDynGrid* [10], що працює з великими обсягами даних. Траєкторії молекулярної динаміки білків, що є результатом комп'ютерних симуляцій, займають сотні гігабайт. Вони є вхідними даними для різноманітних задач аналізу, тому забезпечення їх високої доступності є критичним для функціонування ВО.

Висновки. Проаналізовано існуючі засоби забезпечення високої доступності даних у грід-інфраструктурах та доведено необхідність запровадження служби реплікації даних в Українському національному грід-сегменті. Сформовано вимоги до реалізації служби – масштабованість, підтримку декількох ВО та політик реплікації, прозорість доступу до даних.

Представлено архітектуру автономної служби реплікації даних, що використовує каталог даних *LFC* як для зберігання списків реплік, так і для задання самих політик реплікації. Автономність служби забезпечується через використання механізмів отримання делегації користувачів-власників файлів із служби тимчасових посвідчень *MyProxy* та засвідчення участі користувачів у ВО через службу *VOMS*.

Реалізація служби складається із набору модулів, що виконуються паралельно і взаємодіють через загальний кеш об'єктів. Така архітектура забезпечує швидкий запуск, зупинку та відновлення роботи служби у випадку збою. Взаємодія з користувачами відбувається через модифікацію поля коментаря об'єкта у каталозі *LFC*. Підтримується дві схеми взаємодії – з використанням інтерфейсу користувача *LFC* та за допомогою власної утиліти керування політиками реплікації.

Розроблену реалізацію автономної служби реплікації інтегровано з центральним каталогом даних Українського національного гріду (УНГ). Інтеграція з віртуальною організацією *MolDynGrid* дозволила автоматизувати забезпечення високої доступності великих обсягів даних, розподілених між сховищами УНГ. Планується інтеграція служби реплікації з іншими віртуальними організаціями національної грід-інфраструктури.

1. *Демичев А.*, *Ильин В.*, *Крюков А.* Введение в грид-технологии. – 2007. – *http://www.sinp.msu.ru*
2. *Foster I.*, *Kesselman C.*, *Tuecke S.* The Anatomy of the Grid – Enabling Scalable Virtual Organizations // Intern. J. of Supercomputer Appl. – 2001. – **15**. – P. 2001.
3. *Advanced* Resource Connector middleware for lightweight computational Grids / M. Ellert, M. Gronager, A. Konstantinov et al. // Future Gener. Comput. Syst. – 2007. – **23**, N 1. – P. 219–240.
4. *The gLite* File Transfer Service / Paolo Badino, Ricardo Brito da Rocha, James Casey et al. // 1st EGEE User Forum, CERN, Geneva, Switzerland – 01–03 Mar. 2006. – P. 94.
5. *Calanducci T.* LFC: The LCG File Catalog // gLite Bratislava.– 27–30 Jun. 2005.
6. *Storage* resource manager version 2.2: design, implementation, and testing experience. / F. Donno, P. Badio, E. Corso et al. // J. of Physics: Conf. Series. – **119**. – IOP Publ., 2008.
7. *Basney J.*, *Humphrey M.*, *Welch V.* The MyProxy Online Credential Repository // J. Software: Practice and Experience. – 2005. – **35** (9). – P. 801–816.
8. *An Authorization* System for Virtual Organizations / R. Alfieri, R. Cecchini, V. Ciaschini et al. // Proc. of the 1st Europ. Across Grids Conf., Santiago de Compostela. – 2003. – P. 13–14.
9. *Jean-Philippe Baud*. Grid File Access Design // [Draft] LCG Design document. – 13th May 2003. – http://lcg.web.cern.ch/LCG/peb/GTA/GTA-ES/Grid-File-Access-Design-v1.0.doc
10. *Virtual* Laboratory MolDynGrid as a Part of Scientific Infrastructure for Biomolecular Simulations / A.O. Salnikov, I.A. Sliusar, O.O. Sudakov et al. // Computing. – 2010. – **9**, N 4. – P. 295–301.

Поступила 10.02.2012

Тел. для справок: (044) 526-1214 (Киев) +380 67 500-8124
E-mail: *slu@grid.org.ua*

●

Е.А. Слюсар

Автоматизированная служба репликации файлов для организации высокой доступности данных в грид-инфраструктуре

Введение. Грид-система – это сочетание технологий, инфраструктуры и стандартов. Технологии предусматривают специальное программное обеспечение промежуточного уровня, позволяющее организациям предоставлять свои информационные и вычислительные ресурсы в общее пользование. Инфраструктура состоит из аппаратных средств и служб на основе информационных и вычислительных ресурсов, а стандарты определяют формат и протоколы обмена как для взаимодействия между самими службами, так и между службами и пользователями [1].

Доступ к ресурсам грид-инфраструктуры осуществляется с помощью обращения к соответствующим службы с использованием определенных протоколов. Реализация грид-службы скрывает от пользователей и других служб инфраструктуры внутреннюю архитектуру и особенности функционирования ресурса, который ею обслуживается, и предоставляет унифицированный стандартизированный интерфейс по типу этого ресурса [2].

Постановка задачи

Для виртуальных организаций (ВО), работающих с большими объемами данных, актуальна проблема обес-

печения высокой доступности данных для своих вычислений и оптимизации их размещения среди элементов хранения данных (*Storage Element – SE*) для повышения скорости доступа из ближайших вычислительных элементов (*Computing Element – CE*). Пакет программного обеспечения промежуточного уровня *Nordugrid Advanced Resource Connector* (*ARC*) [3] не содержит в своем составе автоматизированных средств репликации данных. В пакет *gLite* входит служба *File Transfer Service* [4], узкоспециализированная и организующая доставку наборов данных с одного центрального элемента на несколько распределенных хранилищ без взаимодействия с каталогом данных. ВО вынуждены разрабатывать собственные средства репликации данных, соответствующие использованным ими методам обработки. Как следствие, разработанные средства также имеют узкую специализацию и не могут без существенных модификаций использоваться как универсальный инструмент в пределах грид-инфраструктуры.

Требования к реализации автоматизированной службы репликации можно сформулировать так:

- масштабируемость на большое количество элементов хранения данных;
- поддержка нескольких ВО и различных политик репликации для разных ВО;
- автономность работы – управление репликами без вмешательства пользователей;
- прозрачность доступа к данным и интеграция со службой каталога файлов;
- работа от имени пользователей ВО с помощью механизма делегации;
- возможность как централизованного, так и местного развертывания.

Единственная реализация службы репликации данных, соответствующая требованиям различных ВО, будет содействовать развитию грид-технологий в составе национальных грид-инфраструктур, включая украинскую. Это позволит участникам ВО сконцентрироваться на области своих исследований и избавит их от необходимости вручную управлять большими объемами распределенных данных в грид-инфраструктуре.

Архитектура службы

Разработанная служба репликации состоит из нескольких взаимодействующих компонентов. Для обращения к другим службам грид-инфраструктуры используются утилиты и библиотеки пакетов программного обеспечения промежуточного уровня *gLite* и *Nordugrid ARC*. Поскольку политики репликации применяются к логическим именам в каталоге файлов, целесообразно использование общей базы данных для хранения логической иерархии имен вместе с соответствующими метаданными, в состав которых входит описание политик репликации. Реализация службы каталога данных *LCG File Catalog* (*LFC*) [5], применяемая в Украинском грид-сегменте, а также в мировых инфраструктурах *WLCG* и *EGI*, не содержит встроенных средств для хранения специализированных метаданных. Однако каждый именуемый объект в каталоге имеет атрибут *комментарий* – текст свободной формы, который может изменяться пользователем согласно политикам контроля доступа *LFC* и не затрагивает другие аспекты работы каталога. Одна из особенностей предложенной архитектуры – хранение политик репликации в закодированном и сжатом виде в составе комментария к файлу или директории в каталоге *LFC*. Такая организация хранилища метаданных службы репликации имеет ряд преимуществ:

- ***отсутствие собственной базы данных***: все политики репликации хранятся непосредственно как метаданные логических имен в каталоге файлов, а служба репликации обращается к нему для чтения или записи этих политик;
- ***стандартный протокол внешнего интерфейса службы***: взаимодействие пользователей со службой происходит через внесение изменений в комментарии к логическим именам в каталоге *LFC*; при этом используется стандартный протокол обмена *LFC*, предусматривающий аутентификацию и авторизацию пользователей;
- ***переносимость реализации пользовательского интерфейса***: средства интерфейса пользователя базируются на программных библиотеках клиента службы *LFC*, не имеют никаких других существенных зависимостей и могут быть реализованы как в виде *Shell*-сценария, так и в форме самостоятельной утилиты.

Схема взаимодействия компонентов службы репликации показана на рис. 1. Ядро системы выполняет сопоставление характеристик, имеющихся в грид-инфраструктуре элементов хранения данных и параметров логических имен в каталоге файлов, в том числе количество имеющихся реплик и политики репликации, в результате чего формируются запросы на проверку доступности существующих реплик и создания новых.

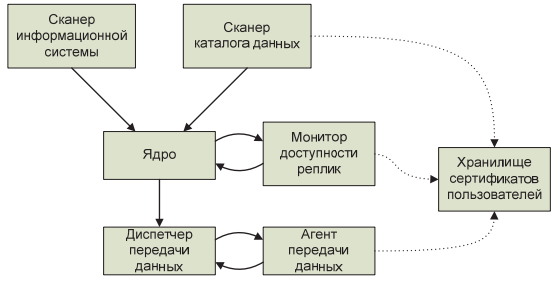

Рис. 1. Взаимодействие компонентов службы репликации

Сканер информационной системы – это компонент, предоставляющий ядру сведения об имеющихся элементах хранения данных и их текущих характеристиках. Один и тот же элемент хранения данных может поддерживать несколько ВО, а каждая ВО в свою очередь может иметь несколько точек входа в хранилище с резервациями дискового пространства (*Storage Space Reservations – SSR*). Эти сведения, вместе с объемами резерваций и фактическими показателями свободного пространства, публикуются в локальной информационной систе-

ме элемента хранения данных. Опубликованные сведения из всех грид-ресурсов передаются на центральный каталог ресурсов инфраструктуры. Сканер информационной системы периодически опрашивает этот каталог и формирует список всех имеющихся хранилищ данных и резерваций на каждую ВО. В списке также указываются ссылки на стандартный интерфейс управления элементом хранения данных *Storage Resource Manager* (*SRM*) [6], поддерживаемый большинством распространенных реализаций грид-хранилищ – *Disk Pool Manager* (*DPM*), *dCache*, *Storage Resource Manager* (*StoRM*), *Berkeley Storage Manager* (*BeStMan*) и *CERN Advanced STORage manager* (*CASTOR*).

Сформированный список заносится в общий кэш объектов службы репликации, откуда его может получить ядро системы. Сканер периодически обновляет кэш объектов, загружая новый список на каждом цикле своей работы.

Сканер каталога данных выполняет обход иерархической структуры логических имен в каталоге *LFC*, получая метаданные и список существующих реплик для каждого имени, где может содержаться описание политики репликации. Поскольку один каталог *LFC* может обслуживать несколько ВО, каждая из которых может устанавливать собственные политики доступа к каталогу, обращение к нему происходит с использованием прокси-сертификатов участников соответствующих ВО. Для получения таких сертификатов в процессе обхода каталога сканер обращается к хранилищу сертификатов пользователей. Если хранилище на момент запроса не содержало необходимого сертификата, то соответствующая ветвь иерархии имен пропускается, а в хранилище направляется запрос на получение прокси-сертификата. В результате одного цикла обхода сканера формируется список всех логических имен, имеющих хотя бы одну реплику и находящихся в поле действия хотя бы одной из политик репликации. Политики репликации наследуются от родительского объекта вниз по иерархии. Сформированный список заносится в общий кэш объектов службы репликации и периодически обновляется сканером.

Хранилище сертификатов пользователей применяется для уменьшения нагрузки на службу делегации *MyProxy* [7] и службу удостоверения участия в ВО (*Virtual Organization Membership Service – VOMS*) [8]. Агент хранилища обращается к кэшу объектов и получает список необработанных запросов на получение прокси-сертификата. Используя собственный сертификат службы, агент осуществляет запросы к службе *MyProxy* на получение делегации пользователей, отличительные имена сертификатов которых (*Distinguished Name – DN*) были указаны в необработанных запросах в кэше. В случае успешного получения делегации, прокси-сертификаты загружаются в кэш. На следующем этапе на основании полученных прокси-сертификатов с помощью службы *VOMS* генерируются новые, но уже со специальным расширением, подтверждающим участие пользователя в соответствующей ВО. Оптимизация заключается в уменьшении количества запросов в службу *MyProxy* в том случае, когда один и тот же пользователь является участником нескольких ВО. Полученные прокси-сертификаты с *VOMS*-разрешением также загружаются в кэш (рис. 2).

Для успешной работы описанного механизма необходимо, чтобы пользователи службы репликации загружали на соответствующий *MyProxy*-сервер долгосрочные делегации и указывали в политике делегирования отличительное имя сертификата службы репликации. Это обеспечит получение делегации и быстрое реагирование на изменение состояния репликации файлов в каталоге.

Сертификат службы также используется в случаях, когда невозможно определить пользователя-владельца какого-либо файла или директории в каталоге файлов, а также при первоначальном сканировании для определения поддеревьев поддерживаемых ВО. Если служба репликации входит в состав ВО как участник, то ее сертификат можно применять вместо пользовательского сертификата во всех операциях, где разрешен доступ к файлам всем участникам ВО. Это касается, например, общих входных данных для последующего анализа участниками ВО.

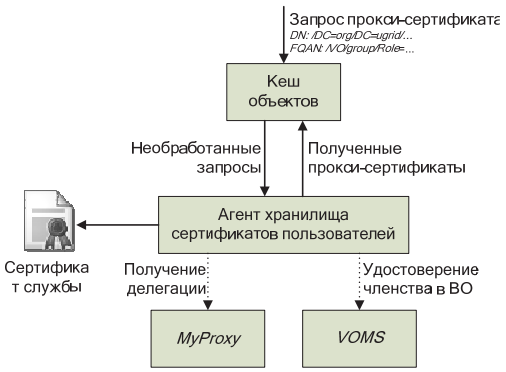

Рис. 2. Схема работы хранилища сертификатов пользователей

Ядро системы получает список объектов и их политики репликации со сканера каталога данных. На первом шаге цикла происходит проверка каждого объекта на соответствие указанным политикам. В частности, проверяется, чтобы количество реплик файла было в установленных пределах. Для уменьшения нагрузки на элементы хранения данных проверка доступности реплики происходит с помощью операции получения списка файлов в соответствующей директории хранилища, результаты которой заносятся в общий кэш. Сначала проверяется наличие в кэше соответствующего элемента, и в случае его отсутствия генерируется запрос к монитору доступности реплик и ядро переходит к рассмотрению следующего объекта.

В случае, когда результаты проверки доступности реплик файла были успешно получены, определяется реальное количество доступных реплик. На основании этих данных принимается решение о создании новой реплики в случае их недостатка или об удалении лишних реплик

в случае их чрезмерного количества. Если количество реплик находится в диапазоне, указанном в политике, то ядро переходит к рассмотрению следующего объекта.

Для создания новой реплики выбирается резервация на элементе хранения данных, где нет ни одной реплики соответствующего файла. Резервации различаются по имени и могут быть указаны в политике репликации. В случае, когда резервация не указана, выбирается пространство по умолчанию соответствующего элемента хранения данных. Полученные таким образом резервации упорядочиваются по рейтингу, который вычисляется исходя из показателей надежности соответствующего вычислительного элемента, получаемых из системы тестирования грид-инфраструктуры или по внутренним результатам проверки доступности, которые формируются и обновляются при проверке доступности реплик службой репликации. В политике репликации указывается порог отсечки – величина, указывающая, из скольких элементов хранения данных с верхушки рейтинга необходимо выбрать местоположение новой реплики. Конечный элемент хранения выбирается случайно.

На следующем шаге выбирается существующая реплика, с которой будет происходить копирование содержимого файла для новой реплики. Также формируется рейтинг среди существующих реплик и выбирается такая пара элементов хранения данных, для которых скорость передачи была максимальной при предыдущих операциях передачи данных, сведения о которых также хранятся в общем кэше службы репликации. Это обеспечит скорейшее создание новой реплики. Запрос на передачу данных, сформированный описанным способом, направляется в очередь диспетчера передачи данных (рис. 3).

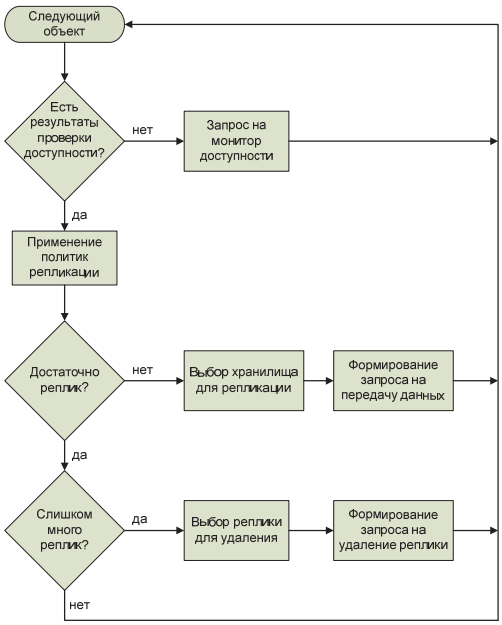

Рис. 3. Схема работы ядра службы репликации

При удалении реплики также строится рейтинг всех использованных элементов хранения данных, включая показатели надежности и свободного дискового пространства. Из него выбирается реплика с минимальным рейтингом и формируется соответствующий запрос на удаление данных и ссылки на реплику из каталога данных.

Монитор доступности реплик выполняет проверку наличия реальных данных по ссылкам, которые указываются как реплики файла. Для снижения нагрузки на элементы хранения данных запросы консолидируются для каждого элемента путем получения списка файлов определенной директории вместо проверки отдельного файла. При автоматическом создании реплик на элементах хранения данных используется простая двухуровневая структура директорий, поэтому вероятность того, что несколько файлов будут иметь реплики в одной и той же директории, достаточно высока. Списки файлов директории заносятся в общий кэш на определенное время, чтобы исключить слишком частые повторные запросы. В результате проверки возможны следующие варианты исхода:

- искомый файл присутствует в списке файлов директории – тогда текущая реплика считается доступной;
- искомый файл отсутствует в списке – реплика считается удаленной;
- элемент хранения данных недоступен на момент запроса – реплика считается недоступной.

Попытка непосредственного получения данных из реплики не выполняется исходя из того, что эти данные могут храниться на ленточных накопителях, и необходимо определенное время и другие ресурсы, чтобы подготовить эти данные для загрузки. Если файл имеет достаточно существенный объем, то это может создать высокую нагрузку на элемент хранения данных.

Перед обращением к хранилищу данных выполняется попытка получить прокси-сертификат из хранилища сертификатов пользователей. Если сертификат не найден, то в кеш вносится запрос на генерацию такого сертификата, и обработка логического имени файла, реплики которого проверяются в текущий момент, откладывается до следующего цикла работы ядра.

Если файл отсутствует в директории на хранилище, значит, данные были удалены пользователем вручную или другой службой грид-инфраструктуры. Поэтому такую реплику необходимо удалить и из каталога данных. Запрос на удаление реплики формируется автоматически и направляется непосредственно на каталог данных.

Если при обращении к элементу хранения данных возникла ошибка, то нельзя однозначно констатировать удаление реплики. Однако можно констатировать временную недоступность хранилища в целом. Успешность запросов к каждому хранилищу заносится и обновляется после каждого запроса в общий кэш службы репликации. Эти сведения в дальнейшем используются для вычисления рейтинга хранилищ при выборе местоположения новой реплики.

Диспетчер передачи данных применяется для распределения нагрузки между элементами хранения данных. Для этого используется таблица состояния, в кото-

рой ведется учет текущих сеансов обмена данными между элементами хранения. Максимальное количество одновременных сеансов чтения и записи для одного хранилища задается в глобальной конфигурации службы. Диспетчер выбирает из очереди запрос, который можно удовлетворить в текущий момент, не нарушая упомянутых показателей, и вызывает агента передачи данных для обслуживания этого запроса.

Агент передачи данных использует стандартные средства программного обеспечения промежуточного уровня для трехстороннего согласования передачи между элементами хранения данных. Таким образом, данные передаются непосредственно между двумя хранилищами, но процесс передачи контролируется третьей стороной – узлом службы репликации. Операция выполняется от имени пользователя-владельца соответствующих данных, прокси-сертификат с соответствующим *VOMS*-разрешением которого получается из хранилища сертификатов пользователей. В случае неудачи при получении сертификата или при согласовании передачи данных, счетчик попыток данного запроса увеличивается, и он остается в очереди. При успешном создании реплики она регистрируется в каталоге данных, и запрос удаляется из очереди.

Общий кэш объектов – это подсистема, обеспечивающая взаимодействие всех других компонентов службы репликации. Кэш представляет собой долговременное хранилище типа *ключ-значение*, объекты в котором имеют ограниченное время жизни. Также на каждый объект вводится счетчик обращений, и необходимым условием для удаления объекта будет равенство этого счетчика нулю. Вместо объекта с заданным ключом может содержаться запрос на получение объекта. Такая схема позволяет компонентам оставлять запросы на получение объекта определенного типа, а другим – обрабатывать такие запросы и заменять их полученным объектом. Блокировка и атомарный доступ к объектам обеспечивается благодаря использованию стандартных примитивов синхронизации операционной среды, таких как семафоры и блокировки чтения-записи.

Подсистема кэша объектов реализована в виде отдельного процесса, взаимодействие с которым происходит через стандартные средства межпроцессного взаимдоействия – общую память и доменные гнезда *UNIX*. Для предупреждения потери объектов при перезапуске службы они периодически из оперативной памяти записываются на диск. Система поддерживает объекты разных типов и разные ключи согласно определениям этих типов.

Реализация и внедрение

Реализация автоматизированной службы репликации данных получила название *RAPTOR – Robot for Autonomous Precisely Tunable Operation of Replication*. Программное обеспечение реализовано на языке С и состоит из следующих модулей:

- диспетчера кеша объектов (`raptor_ocs` – *Object Cache Service*);
- сканера информационной системы (`raptor_iss` – *Information System Scanner*);
- сканера каталога данных (`raptor_fcc` – *File Catalog Crawler*);
- ядроа системы (`raptor_combinator`);
- монитора доступности реплик (`raptor_ram` – *Replica Availability Monitor*);
- диспетчера и агента передачи данных (`raptor_tm` – *Transfer Manager*);
- хранилища сертификатов (`raptor_credman` – *Credential Manager*);
- утилиты для администрирования (`raptor_admin`).

Утилита администрирования позволяет получать статистику работы системы, а также осуществлять запуск и остановку всех модулей. Для доступа к другим службам грид-инфраструктуры были использованы стандартные клиентские библиотеки программного обеспечения промежуточного уровня, в частности, *LFC*, *MyProxy*, *VOMS* и *Grid File Access Layer* (*GFAL*) [9]. Все компоненты, за исключением утилиты администрирования, работают как фоновые службы среды *UNIX* – демоны, и взаимодействуют между собой только через диспетчер кэша объектов. Все компоненты настраиваются с помощью общего файла конфигурации, загружаемого в кэш объектов при старте системы. В состав системы также входит программная библиотека для работы с двумя форматами представления политик репликации – текстовым и двоичным. Текстовый формат может применяться для установки политик репликации непосредственно через пользовательский интерфейс службы каталога файлов *LFC*.

Для большего удобства доступа к службе репликации разработан сценарий-обертка на языке *Python*, использующий стандартные клиентские библиотеки *LFC* и позволяющий устанавливать и декодировать описание политик репликации, размещенное в поле комментария к логическому имени файла или директории в каталоге *LFC*. Поддерживаются следующие параметры политики:

- минимальное количество реплик;
- максимальное количество реплик;
- метод выбора элементов хранения данных – из всех доступных для ВО, из всех доступных из фиксированного списка, по шаблону имени резервации, по рейтинговым коэффициентам;
- атомарная репликация каталога или группы файлов по шаблону имени;
- признак наследования / переопределения.

Поддерживается также отправка уведомлений на указанный электронный адрес при изменении количества реплик объекта.

Разработанная служба репликации была внедрена для обслуживания центрального каталога данных Украинской национальной грид-инфраструктуры, расположенного в Информационно-вычислительном центре Киевского национального университета имени Тараса Шевченко. В частности, служба репликации интегрирована с

виртуальной организацией *MolDynGrid* [10], работающей с большими объемами данных. Траектории молекулярной динамики белков, получаемых с помощью компьютерных симуляций, занимают сотни гигабайт. Они служат входными данными для различных задач анализа, поэтому обеспечение их высокой доступности критично для функционирования данной ВО.

Заключение. Проанализированы существующие средства обеспечения высокой доступности данных в грид-инфраструктурах и доказана необходимость внедрения службы репликации данных в Украинском национальном грид-сегменте. Сформулированы требования к реализации службы – масштабируемость, поддержка нескольких ВО и политик репликации, прозрачность доступа к данным.

Представлена архитектура автономной службы репликации данных, использующая каталог данных *LFC* как для хранения списков реплик, так и для указания политик репликации. Автономность службы обеспечивается путем использования механизмов получения делегации пользователей-владельцев файлов с помощью службы временных удостоверений *MyProxy* и последующего удостоверения участия пользователей в ВО через службу *VOMS*.

Реализация службы состоит из нескольких модулей, выполняемых параллельно и взаимодействующих через общий кэш объектов. Такая архитектура обеспечивает быстрый запуск, остановку и возобновление работы службы в случае сбоя. Взаимодействие с пользователями происходит через модификацию поля комментария к объекту в каталоге *LFC*. Поддерживается две схемы взаимодействия – с использованием стандартного интерфейса *LFC* и с помощью собственной утилиты для управления политиками репликации.

Разработанная реализация автономной службы репликации интегрирована с центральным каталогом данных Украинского национального грида (УНГ). Интеграция с виртуальной организацией *MolDynGrid* позволила автоматизировать механизмы обеспечения высокой доступности больших объемов данных, распределенных между хранилищами в составе УНГ. Планируется дальнейшая интеграция службы репликации с другими виртуальными организациями национальной грид-инфраструктуры.