

Веб-сервіс доступу в ґрід для системи моделювання

Зеленюк Олексій, 07-05-2011

Структура роботи



Основні принципи SOA:

- Стандартизований контракт. Інтерфейс взаємодії сервісів описується документально;
- Слабка зв'язність. Взаємозв'язки між сервісами мають бути такими, що мінімізують взаємозалежності;
- Абстрагування сервісів. Внутрішня логіка сервісу має бути прихована від зовнішнього світу, який обізнаний лише з його контрактом;
- Повторне використання. Логіка розбивається на сервіси з думкою про вигоду повторного використання;
- Автоматизоване виявлення. Сервіси супроводжуються метаданими, що уможливають їх автоматизоване виявлення та ідентифікацію;
- Здатність до компонування. Сервіси мають бути добре придатними для поєднання, незалежно від складності композиціонування.

Ідеології створення веб-сервісів

SOAP чи REST ?

Елементи SOAP

- SOAP (Simple Object Access Protocol, простий протокол доступу до об'єктів) як протокол обміну XML-повідомленнями,
- WSDL (Web Service Description Language, мова опису веб-сервісів) як стандартна мова опису контрактів сервісів,
- UDDI (Universal Description Discovery & Integration, універсальний опис, виявлення, інтеграція) як стандартний механізм для пошуку та реєстрації сервісів.

Платформи створення веб-сервісів

- Borland
- Oracle
- Microsoft
- HP e-services
- IBM Web Services

Порівняння грид та веб-сервісів

Спільні риси:

- Модульність;
- Слабка зв'язність;
- Прагнення до повторного використання;
- Абстрагування;
- Здатність до поєднання;
- Моніторинг;
- Категоризація.

Порівняння грід та веб-сервісів

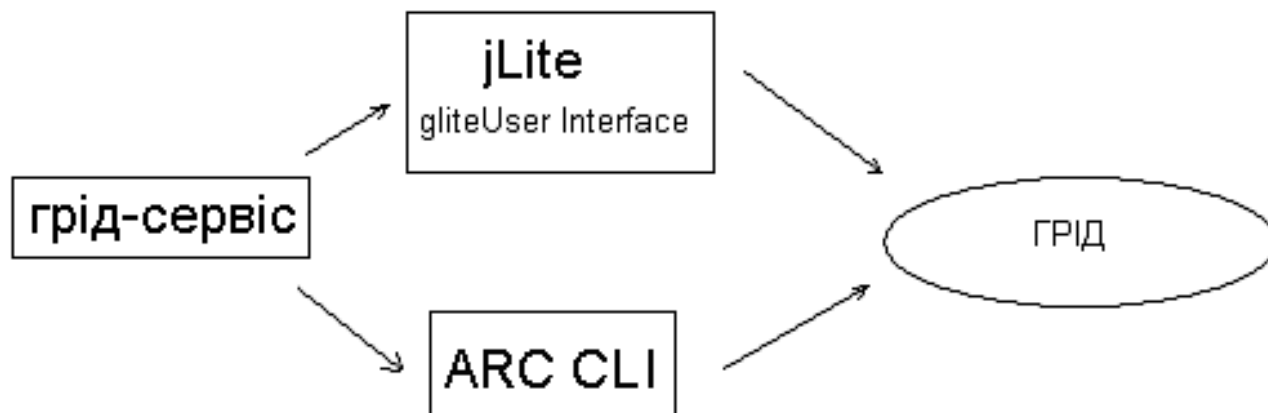
Відмінності:

	Веб-сервіси	Грід-сервіси
Життєвий цикл	Від запита до відповіді, збереження даних у БД	Існування між запитами, впродовж часу існування грід-ресурсу
Внутрішній стан	Відсутність спеціальної підтримки як ідеологічно, так і в стандартах	Має підтримуватись через особливості грід-ресурсів
Оповіщення	Прийнято додаткові стандарти WS-Notification	Мають підтримуватись через специфіку роботи Грід.
Безпека	Стандарти WS-Security	Підтримка GSI
Стандартизація	Визначені, зрілі стандарти	Наявність різних реалізацій проміжного програмного забезпечення грід не сприяє стандартизації

Порівняння грід-сервісів

	Libarcclient	CREAM	WMPoxy	Створений грід-сервіс
ППО	Arc, gLite, UNICORE	gLite	gLite	gLite, Arc
Формати опису завдань	xRSL, jSDL, JDL	JDL	JDL	JDL, xRSL
Платформи	Linux, MacOS, Windows	Linux	Linux	Windows , Linux, MacOS
Тип завдань	jobSubmission(resubmission, resuming), jobQuering, jobKilling	jobSumbission, jobKilling, GSI аутентифікація, VOMS авторизація	jobSumbission, jobKilling,	jobSumbission, jobStatus, VOMS авторизація, jobOutputRetriev
Наявність API	Python, Java	C++	Java	Немає, можливо додати
Готовність	запуск в 2013 році	готовий	готовий	Дотупна альфа-версія

Архітектура грід-сервіса



Тестування

The screenshot shows a web browser window with several tabs: 'GridSubmit.java', 'Result.jsp', 'Web Services Test...', 'Input.jsp', and 'Method.jsp'. The main content area is divided into three sections:

- Methods:** A list of methods with their signatures:
 - [jobSubmit](#) (java.lang.String, java.lang.S...
 - [getJobStautus](#) (java.lang.String, java.lang.S...
 - [getJobOutput](#) (java.lang.String, java.lang.S...
- Inputs:** A form for entering test parameters:
 - grid: gLite (dropdown menu)
 - password: [text input]
 - voms: [text input]
 - jdkPath: [text input] [Browse...]
 - inputDir: [text input] [Browse...]
 - Buttons: Invoke, Clear
- Result:** A section displaying the test outcome: result: N/A

Дякую за увагу