

Прототип автоматизованої системи тестування грід-ресурсів на базі Nordugrid ARC. Технічна документація

1 Системні вимоги

Автоматизована система тестування являє собою набір сценаріїв для PHP, Bourne-shell та програм мовою C. Відповідно, на цільовій машині має бути встановлено веб-сервер із підтримкою PHP-сценаріїв, sh-сумісний інтерпретатор сценаріїв (наприклад, `bash`), а також компілятор мови C (зазвичай `gcc`). Для компіляції утиліти отримання сертифікатів необхідна наявність бібліотек та заголовкових файлів пакету OpenSSL версії 0.9.8 і вище, а також утиліти GNU Make. Автоматизована система тестування спирається на програмне забезпечення промислового рівня грід-інфраструктури Nordugrid ARC версії не нижче 0.8 та не вище 1.0, яке також має бути встановлене на цільовій машині. Якщо буде застосовуватись отримання делегації користувача від MuProху-сервера, то для цільової машини має бути згенеровано сертифікат, за допомогою якого буде відбуватись звернення для сервера делегації.

2 Підготовка до інсталяції

Необхідно розпакувати отриманий архів із набором файлів автоматизованої системи тестування. Це можна зробити наступною командою:

```
tar -zxvf ast-0.2.tar.gz
```

Всередині архіву розгашовані наступні теки:

- `c` – вихідні коди утиліт мовою C;
- `contrib` – приклади сценаріїв та конфігурації;
- `scripts` – Shell-сценарії системи тестування;
- `web` – веб-інтерфейс системи тестування.

2.1 Компіляція утиліт мовою С

У директорії `s` міститься файл автоматизованої компіляції за допомогою утиліти GNU Make – `Makefile`. В залежності від системи, утиліта GNU Make може викликатись командами `make` чи `gmake`. Якщо бібліотека OpenSSL розміщена у нестандартному каталозі, то необхідно внести зміни до `Makefile`. Далі необхідно скомпілювати утиліти, викликавши GNU Make у каталозі `s`.

2.2 Автоматичне отримання делегації

Автоматизована система тестування працює від імені користувача грид-інфраструктури, який вповноважений відправляти завдання на усі грид-ресурси, що тестуються. Для роботи утиліт Nordugrid ARC необхідний проксі-сертифікат – делегація повноважень такого користувача. Із міркувань безпеки отримання делегації користувача винесено у зовнішній сценарій. Приклад такого сценарію, що реалізує отримання делегації із сервера MyProxy, приведено у файлі `proxugen.sh` із каталогу `contrib`.

У випадку отримання делегації із MyProxy-сервера, на цільовій машині має бути сертифікат вузла та його закритий ключ, доступний для локального облікового запису користувача, від імені якого будуть запускатися веб-сценарії системи тестування. Довгострокова делегація уповноваженого користувача має бути завантажена на обраний MyProxy-сервер, із дозволом на отримання короткострокової делегації вузлом системи тестування. Також, у конфігурації MyProxy-сервера DN сертифіката вузла системи тестування має бути внесений до списку довірених одержувачів (`trusted_retrievers`).

Для завантаження довгострокової делегації на сервер MyProxy може використовуватись така команда:

```
myproxy-init -x --dn_as_username -s myproxy-server \  
--retrievable_by_cert "/DC=org/DC=ugrid/..."
```

Сценарій отримання делегації у загальному випадку має приймати параметр командного рядка, що вказує на файл, у який у випадку успішного отримання делегації, необхідно записати проксі-сертифікат. Якщо отримати делегацію не вдалось, сценарій має завершитися із кодом повернення, відмінним від нуля.

3 Інсталяція

3.1 Веб-інтерфейс

Щоб автоматизована система тестування стала доступною через веб-інтерфейс, необхідно розмістити файли із каталогу `web` на веб-сервері. На веб-сервері має бути підтримка серверних сценаріїв мовою PHP, а у налаштуваннях PHP має бути вимкнений режим Safe Mode, оскільки сценарії системи тестування будуть запускати фонові процеси (по одному на кожен грид-ресурс, що тестується). Перевірити налаштування PHP можна за допомогою сценарію `phpinfo.php` із теки `contrib`.

Необхідно відредагувати файл `config.php.sample` та перейменувати у `config.php`. Файл містить наступні параметри:

- `STATE_DIR` – повний шлях до каталогу стану системи тестування;
- `TESTER_BE` – повний шлях до сценарію запуску операцій тестування;
- `PROXY_GEN` – повний шлях до сценарію отримання делегації;
- часовий пояс для виведення часу запуску тестів (не обов'язково).

3.2 Допоміжні сценарії

Допоміжні сценарії можуть розміщуватись як у окремому каталозі, так і у каталозі веб-інтерфейсу. Бажано каталог сценаріїв тестування винести за межі кореневого каталогу веб-сервера, або, якщо цього зробити не вдалось, то необхідно вжити заходів, щоб заборонити скачування цих сценаріїв випадковими користувачами.

До каталогу допоміжних сценаріїв необхідно скопіювати скомпільовані утиліти мовою C, зокрема, утиліту `get-cert`. Також, необхідно перевірити, що обліковий запис користувача, від імені якого працюють веб-сценарії, має доступ на виконання до цього каталогу.

Необхідно відредагувати файл `testconf.sh.sample` та перейменувати у `testconf.sh`. Файл містить наступні параметри:

- `NORDUGRID_PROFILE` – повний шлях до файлу сценарія підвантаження оточення пакету Nordugrid ARC. Він має забезпечувати виконання стандартних клієнтських утиліт без вказання шляху та додаткових динамічних бібліотек.
- `TRUSTED_CA_DIR` – повний шлях до каталогу із корневими сертифікатами довірених центрів сертифікації, за допомогою яких буде здійснюватись перевірка сертифікатів вузлів ґрід-ресурсів, що тестуються.
- `INFOSYS_COUNT` – кількість каталогів ґрід-ресурсів, у яких має реєструватись ґрід-ресурс. Параметри кожного каталогу ресурсів задаються у вигляді елементів відповідного масиву із індексом каталога ресурсів, починаючи з 1. Наприклад:

```
INFOSYS_HOST[1]="giis.grid.org.ua"  
INFOSYS_PORT[1]="2135"  
INFOSYS_ROOT[1]="Mds-Vo-name=Ukraine,o=Grid"
```

- `TRUSTED_CA_COUNT` – кількість корневих центрів сертифікації, яким мають довіряти ґрід-ресурси, які тестуються. Список DN корневих сертифікатів довірених центрів задається у вигляді масиву. Наприклад:

```
TRUSTED_CA_DN[1]="/DC=org/DC=ugrid/CN=UGRID CA"
```

- **SUPPORTED_VO_COUNT** – кількість віртуальних організацій, які мають підтримувати грід-ресурси, що тестуються. Імена віртуальних організацій задаються у вигляді масиву. Наприклад:

```
SUPPORTED_VO[1]="ukraine"
```

- **SUPPORTED_RTE_COUNT** – кількість додаткових середовищ виконання, які мають підтримувати грід-ресурси, що тестуються. Назви середовищ виконання задаються у вигляді масиву. Наприклад:

```
SUPPORTED_RTE[1]="ENV/GROMACS"
```

Стандартні середовища **ENV/X509_USER_PROXY** та **ENV/ARC** перевіряються автоматично. Якщо додаткові середовища виконання перевірити непотрібно, то необхідно вказати кількість 0.

- **JOB_POLL_MINUTES** – термін у хвиликах, протягом якого тестове завдання має повністю завершитись.
- **JOB_CHECK_SLEEP** – інтервал між перевірками стану завдання.
- **AST_HTTP_ROOT** – повне посилання до веб-інтерфейсу системи автоматичного тестування. Наприклад, якщо систему встановлено у кореневий каталог веб-сервера **ast.grid.org.ua**, то визначення посилання буде мати вигляд:

```
AST_HTTP_ROOT="http://ast.grid.org.ua/"
```

- **GRIDFTP_EXTERNAL_SERVER** – посилання на каталог зовнішнього GridFTP-сервера, що буде використовуватись для тестів. Конфігурація сервера має дозволяти створювати, завантажувати та видаляти файли у вказаному каталозі користувачеві, делегацію якого використовує система тестування.

Всі параметри є обов'язковими.

3.3 Каталог стану

Відомості про стан тестів грід-ресурсів зберігаються у директорії стану, до якої має бути доступ із веб-сценаріїв – на читання, а із shell-сценаріїв – на читання і запис. Крім того, обраний каталог бажано не публікувати для доступу через веб, оскільки він містить файли, із вмісту яких можна виявити внутрішні налаштування вузла та системи тестування.

Якщо каталог стану не вдалось розмістити за межами кореневого каталогу веб-сайту, то, у випадку веб-сервера Apache, можна скористатися файлом **htaccess-denyall** із каталогу **contrib**. Його необхідно розмістити безпосередньо у каталозі стану під іменем **.htaccess**.

4 Налаштування списку ресурсів

Список грід-ресурсів, що мають тестуватися, задається у файлі `sitelist.php` у каталозі сценаріїв веб-інтерфейсу. Приклад наведено у файлі `sitelist.php.sample`. Щоб додати новий грід-ресурс до списку, необхідно згенерувати для нього GUID. Для цього можна скористатися утилітою `uuidgen`. Массив сайтів складається із елементів, кожен із яких є асоціативним масивом. У якості ключів мають бути вказані такі параметри:

- `name` – назва грід-ресурсу, яка буде відобразитися у веб-інтерфейсі. Щоб уникнути проблем із екрануванням символів, рекомендується не використовувати прогалини.
- `ce_host` – DNS-ім'я вузла обчислювального елемента відповідного грід-ресурсу.

Сховища даних додаються до конфігурації аналогічним чином. Використовуються наступні параметри:

- `name` – назва сховища, яка буде відобразитися у веб-інтерфейсі. Щоб уникнути проблем із екрануванням символів, рекомендується не використовувати прогалини.
- `se_name` – назва елемента зберігання даних відповідного грід-ресурсу так як вона вказана у інформаційній системі.

Приклад опису списку грід-ресурсів:

```
$itelist = array(
    "a30a71dc-9d63-4ba2-bc39-282d16039120" => array(
        "name" => "IPM",
        "ce_host" => "grid.ipm.lviv.ua",
    ),
);

$selist = array(
    "f821cf2f-76cd-4b70-8fa3-6f5525665fa9" => array(
        "name" => "KNU_1",
        "se_name" => "pub:arc.univ.kiev.ua"
    ),
);
```

5 Веб-інтерфейс системи тестування

5.1 Інтерфейс користувача

Веб-інтерфейс складається із головної сторінки, де відображається список усіх обчислювальних елементів та сховищ, вказаних у конфігурації, із вказанням часу останнього тестування та його результатів. За допомогою

посилання «Test now» виконується запуск процедури тестування обраного грід-ресурсу. За допомогою посилання із результатом тесту викликається сторінка із детальним протоколом останнього тестування відповідного грід-ресурсу.

Система автоматичного тестування дозволяє паралельно тестувати декілька ресурсів. Сценарії тестів побудовані так, що паралельно запуснені тести не створюють конфліктів та не залежать один від одного.

5.2 Програмний інтерфейс

Програмний інтерфейс використовується самою системою тестування, але може бути використаний сторонніми засобами. Звернення до інтерфейсу відбувається через HTTP-запит до сценарію `jgl.php` із вказанням параметру запити «1». Наприклад, якщо система тестування встановлена на вузлі `ast.grid.org.ua`, то звернутись до програмного інтерфейсу можна за наступним посиланням:

```
http://ast.grid.org.ua/jgl.php?l=mun_запиту
```

У відповідь на запит сценарій системи тестування згенерує текстовий файл, вміст якого залежить від типу запити. Загальнодоступні типи запитів:

- `giislist` – отримати список центральних каталогів грід-ресурсів, на використання яких налаштований даний примірник системи автоматизованого тестування. Отриманий таким чином файл можна застосовувати без змін у якості вхідного файлу списку каталогів грід-ресурсів для утиліт Nordugrid ARC.
- `sites_ok` – отримати список імен вузлів обчислювальних елементів, що у поточний момент часу успішно пройшли усі тести. Отриманий таким чином файл можна без змін застосовувати у якості списку дозволених грід-ресурсів при направленні задачі засобами утиліти `ngsub` пакету Nordugrid ARC.
- `se_ok` – отримати список імен елементів зберігання даних, що у поточний момент часу успішно пройшли усі тести.

6 Опис сценаріїв тестування

При запуску тестів для обраного обчислювального елемента чи сховища сценарії веб-інтерфейсу викликають сценарій мовою `bash` – `tester.sh`. Цей сценарій переводить виконання у фоновий режим та створює журнал тестування. Після цього викликається сценарій `testlogic.sh`, що реалізує послідовне виконання відповідних тестів. Сценарії тестів містяться у каталозі `scripts/tests` та викликаються у порядку згідно логіки тестування.

6.1 Локальні тести

Локальні тести виконуються безпосередньо на вузлі, де мають запускатись служби грід-ресурсу на основі Nordugrid ARC. Сценарії локальних тестів можуть викликатися вручну адміністратором грід-ресурсу.

Перевірка файлу конфігурації ARC: 01_local_conf_check.sh

У даному тесті аналізується вміст файлу конфігурації служб грід-ресурсу `/etc/arc.conf`. Перевіряється коректність синтаксису – структура файлу, допустимі блоки та директиви, їх параметри.

Локальні тести оточення: 02_local_env_check.sh

Перевіряється коректність роботи утиліт системи керування ресурсами шляхом запуску одно- та багатоядерних тестових завдань та подальшого моніторингу їх стану.

Локальний тест інтерфейсу користувача: 03_local_ui_check.sh

Сценарій тестування із використанням проксі-сертифіката отриманого із центрального сервера системи тестування, або, за бажанням адміністратора грід-ресурсу – із використанням його власного проксі-сертифіката, виконує постановку завдання на локальний грід-ресурс. При цьому викликається той самий сценарій, що й застосовувався при тестуванні локальної системи керування ресурсами – `02_local_env_check.sh`. Виконується постановка, опитування стану та відміна завдання.

6.2 Віддалені тести грід-ресурсу

Тест інформаційної системи: 04_infosys.sh

Опитуються центральні каталоги грід-ресурсів на предмет наявності реєстрації у них грід-ресурсу, що тестується. Визначається розміщення інформаційної служби грід-ресурсу та відбувається перевірка відомостей про грід-ресурс, що опубліковані в ній. Перевіряється підтримка довірених центрів сертифікації та віртуальних організацій, а також середовищ виконання грід-завдань у випадку служби обчислювального елемента.

Тест сертифікації вузла: 05_hostcert.sh

Аналізується термін дії та входження центру сертифікації, що видав сертифікат даному вузлу, до списку довірених центрів IGTF.

Тест GridFTP-інтерфейсу: 06_gridftp.sh

Для перевірки служби обчислювального елемента ARC відбувається звернення до GridFTP-сервера, вказаного у інформаційній системі.

Простий тест завдання: 07_simple_job.sh

На сервер направляється просте завдання, що містить перевірку можливості доступу до глобальної мережі, а саме – звернення до програмного інтерфейсу системи тестування. Шаблон сценарію тестового завдання знаходиться у файлі `scripts/jobdata/07_simple_job_survey.sh`.

Тест відділеного вводу/виводу даних: 08_remote_io_job.sh

На обраний грід-ресурс направляється завдання, шаблон сценарію якого міститься у файлі `scripts/jobdata/08_remote_io_job_survey.sh`. Сценарій перевіряє доступність клієнтських утиліт пакету ARC в контексті завдання та виконує звернення до зовнішнього GridFTP-сервера.

Тест запуску завдання із віддаленими даними: 09_gate_io_job.sh

На сервер направляється завдання, опис якого містить посилання на вхідні та вихідні файли, розміщені на віддаленому сервері.

Тест сховища даних: 12_storage_putget.sh

Відбувається звернення до відповідного GridFTP чи HTTPg-сервера та перевіряється його сертифікат. Далі отримується список файлів сховища. Виконується спроба завантажити файл на сховище та знову отримується список файлів. Перевіряється наявність щойно завантаженого файлу у списку. Щойно завантажений файл отримується на вузол системи автоматичного тестування та порівнюється із оригіналом. Далі надсилається запит на видалення файлу із сховища та знову отримується список файлів. Перевіряється успішність видалення файлу.

6.3 Колективні тести

Тест запуску завдання із результатами іншого завдання у якості вхідних даних: 10_collective_inout_job.sh

Для перевірки передачі вихідних даних одного завдання на вхід іншому, на один із обчислювальних елементів із списку, отриманого із програмного інтерфейсу системи тестування, ставиться завдання, що генерує файл випадкового вмісту. Шаблон сценарію завдання знаходиться у файлі `scripts/jobdata/10_producer_job.sh`. Після завершення роботи цього завдання на обчислювальний елемент, що тестується, направляється інше завдання, що у своєму описі вхідних файлів має посилання на gsiftp-адресу попереднього завдання. Шаблон сценарію завдання знаходиться у файлі `scripts/jobdata/10_consumer_job.sh`.

Тест направлення нового завдання на грід-ресурс із контексту поточного завдання: 11_collective_submit_job.sh

На грід-ресурс направляється завдання, яке містить команди постановки іншого завдання на інший обчислювальний елемент, його моніторингу та завантаження вхідних і вихідних файлів. Шаблон сценарію завдання міститься у файлі `scripts/jobdata/11_submitter_job.sh`. Список доступних грід-ресурсів для направлення завдання отримується через програмний інтерфейс системи тестування.

Тест тристоронньої передачі даних: 13_storage_transfer.sh

На зовнішній сервер із вузла системи автоматичного тестування завантажуються тестовий файл. Далі, за допомогою утиліти `ngtransfer` налашто-

вується та проводиться передача файлу із зовнішнього сервера на сховище, що перевіряється. Потім відповідний файл із сховища за допомогою утиліти `ngtransfer` знову завантажується на зовнішній сервер, але під іншим іменем. Із зовнішнього сервера файл завантажується на вузол системи автоматичного тестування та порівнюється із оригінальним. Далі тестові файли видаляються із сховища та зовнішнього сервера.