КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет радіофізики, електроніки та комп'ютерних систем

Кафедра комп'ютерної інженерії

Тестування системи керування ресурсами обчислювального кластера OpenPBS

Дипломна робота бакалавра

студента 4 року навчання

Спеціальність: 123 «Комп'ютерна інженерія»

Станіслава ЛОЖКИ

Науковий керівник

канд. фіз.-мат. наук Олександр СУДАКОВ,

доцент кафедри медичної радіофізики

Рецензент

канд. фіз.-мат. наук Андрій НЕТРЕБА

Декан факультету радіофізики,

електроніки та комп'ютерних систем

До захисту допускаю:

Завідувач кафедрою

Доктор технічних наук,

проф. ПОГОРІЛИЙ Сергій Дем'янович

Ухвалено на засіданні кафедри "____" _____2022 р., протокол № ____

Київ 2022

РЕФЕРАТ

Дипломна робота бакалавра 49 с., 22 рис., 1 таб., 17 джерел. Показана можливість встановлення та налаштування системи керування ресурсами OpenPBS на кластері Київського національного університету імені Тараса Шевченка. Налаштовано підтримку функцій Preemptive Scheduling, Fair Share та ін. Продемонстровано працездатність встановленої системи. Наведено порівняння функціоналу планувальника MAUI та OpenPBS за результатами тестування. Зроблено висновки про можливість заміни MAUI на OpenPBS на кластері Київського національного університету імені Тараса Шевченка

Ключові слова КЛАСТЕР, Torque, MAUI, PBSPro, OpenPBS.

Зміст

Вступ

На сьогодні технології дуже швидко розвиваються. І іноді програмне і апаратне забезпечення встановлене у освітніх закладах, компаніях, навіть на персональних комп'ютерах застаріває, через що її потрібно оновлювати. Така проблема виникла і на кластері Київського національного університету імені Тараса Шевченка, а точніше застаріли система керування ресурсами, що виконувала свої задачі, але вже не встигала за прогресом. Тому стала необхідність в оновленні програмного забезпечення на кластері.

Система керування ресурсами виконує задачі з розподілу завдань по вузлам кластеру. На кластері Київського національного університету імені Тараса Шевченка на даний момент встановлене програмне забезпечення Torque і Maui. Вони вже не підтримуються та не оновлюються, через що нове апаратне та програмне забезпечення вступає з ними в конфлікт та заважає оновленням кластеру.

Для вирішення цієї проблеми в даній роботі виконано встановлення нового програмного забезпечення системи керування ресурсами, а саме OpenPBS. Окрім демонстрації можливості встановлення виконано налаштування та тестування роботи нової системи. Налаштування відповідають потребам університету, для можливості апаратного та програмного оновлення кластеру, що дозволить у подальшому розвивати кластер Київського національного університету імені Тараса Шевченка, робота якого забезпечує складні розрахунки для наукових робіт викладачів і студентів усього університету.

1. Кластер

Обчислювальний кластер являє собою набір комп'ютерів, які працюють разом, так щоб їх можна розглядати як єдину систему. На відміну від мережевих комп’ютерів , всі вузли комп’ютерних кластерів налаштовані на виконання одного й того ж завдання і працюють як складові однієї великої системи, яка керується програмним забезпеченням. Компоненти кластера здебільшого з’єднуються один з одним через швидкі локальні мережі , причому кожен вузол (комп’ютер, що використовується як сервер) має свій власний екземпляр операційної системи. У більшості випадків усі вузли використовують одне й те саме обладнання та одну операційну систему, хоча в деяких налаштуваннях можна використовувати різні операційні системи на кожного комп’ютера або іншого обладнання. Кластери зазвичай розгортаються для підвищення продуктивності та доступності порівняно з одним комп’ютером. Окрім цього зазвичай вони набагато рентабельніші, ніж окремі комп’ютери. Комп'ютерні кластери з'явилися в результаті зближення ряду обчислювальних тенденцій, включаючи доступність недорогих мікропроцесорів, високошвидкісних мереж і програмного забезпечення для високопродуктивних розподілених обчислень . Вони мають широкий діапазон застосування та розгортання, починаючи від кластерів малого бізнесу з кількома вузлами до деяких із найшвидших суперкомп’ютерів у світі. На відміну від високонадійних мейнфреймів, кластери дешевше масштабувати, але вони також мають підвищену складність у обробці помилок.

1.1. Кластер Київського національного університету

Кластер - частина проекту "Комп'ютерна мережа Київського університету" і створений для вирішення прикладних задач, що оперують великими об'ємами інформації і потребують великих витрат машинного часу (Рис 1.1).

Рисунок1.1.Фотографія кластеру (взято з сайту університету http://cluster.univ.kiev.ua/ukr/?history)

1.1.1. Опис архітектури кластеру

Перший кластер було створено у 1999 році на ІОЦ з двох персональних комп'ютерів. Була встановлена бібліотека для роботи з матрицями, яка легко розпаралелюється, та програму GAMESS, що використовувала дану бібліотеку.

Протягом 2000 року навантаження на кластер зросло й досягло досить високого рівня. Було вирішено провести модернізацію системи, що й було

зроблено в березні-квітні 2001 року в рамках програми підтримки вищих навчальних закладів корпорації Intel.

У 2005 році за ініціативою Інституту Теоретичної фізики Національної Академії Наук України та Інформаційно-обчислювального центру Київського національного університету в Україні створюється сегмент обчислювальних ресурсів для участі в міжнародних проектах з GRID-обчислень. Кластер підключений до декількох GRID систем.

З того часу кластер постійно розширюється силами інформаційно-обчислювального центру Київського національного університету імені Тараса Шевченка і використовується у наукових дослідженнях та навчальному процесі багатьох наукових та освітніх установ України.

1.1.2. Технічні дані кластеру Київського Національного Університету

Кластер Київського національного університету імені Тараса Шевченка належить до гетерогенних кластерів типу BEOWULF. Gigabit Ethernet використовується як мережа для зв’язку. На вузлах встановлено операційну систему Linux на основі поставки Centos 7 rhel fedora. Вузли кластера мають однакову структуру каталогів та містить, по можливості, однаковий набір програмного забезпечення, встановленого в одні й ті ж каталоги (рис1.2). Основна модель програмування - розподілена пам'ять, проте в межах одного вузла підтримується модель спільної пам'яті.

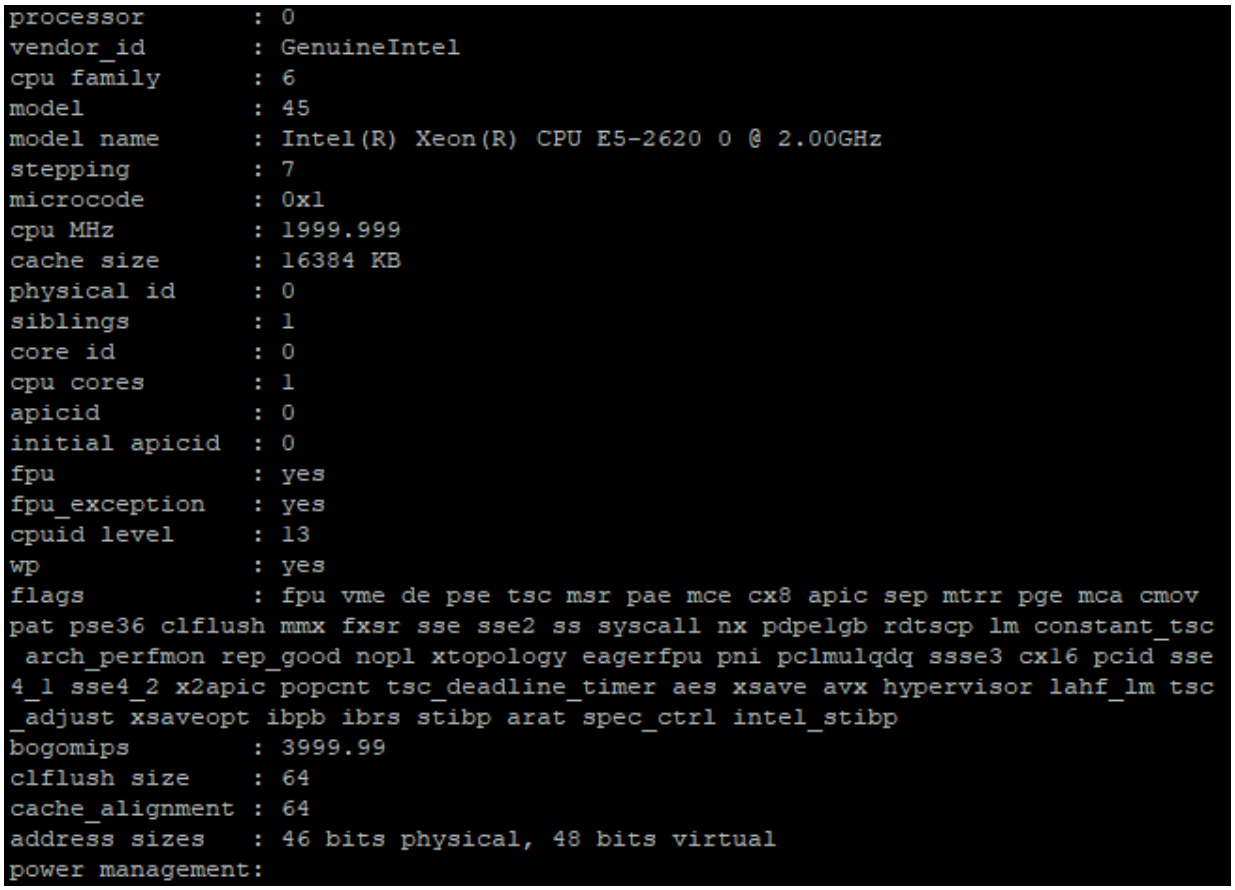

Рисунок 1.2 Конфігурація вузла. Сім процесорів Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz

Комутатор

Mellanox 1016X

Операційна система

Кластер працює під керуванням операційної системи Linux (рис 1.3).

На сьогодні використовуються: поставки Centos 7 rhel fedora, з вузлами Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz

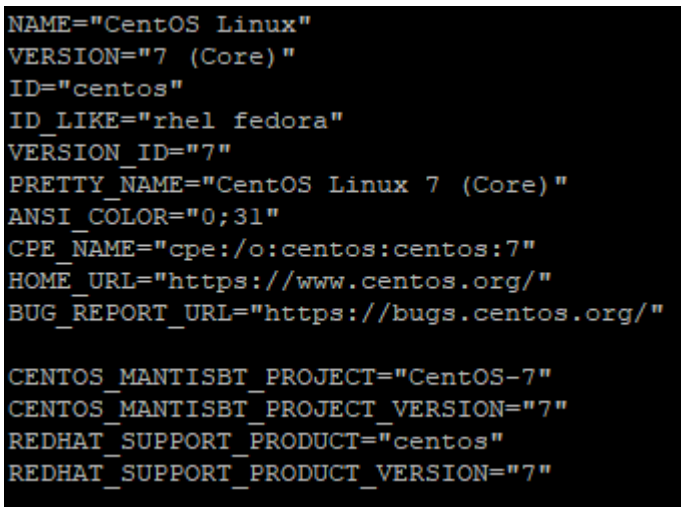

Рисунок 1.3. Операційна система

СИСТЕМА РОЗПОДІЛУ НАВАНТАЖЕННЯ

Система керування ресурсами Torque - 2.1.6

Планувальник MAUI - 2.3.6p17

ІНТЕРФЕЙСИ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ

З інтерфейсів паралельного програмування доступні OpenMP, MPI, а також PVM.

На сьогодні встановлено:

LAM/MPI 6.5.9, 7.0.1, 7.1.1

MPICH 1.2.4.

PVM 3.4.3.

IntelMPI

ОСНОВНІ МАТЕМАТИЧНІ БІБЛІОТЕКИ

MKL, ATLAS, SCALAPACK 1.7, FFTW 2.1.3, доступні також інші бібліотеки, які звичайно входять в поставку Linux.

МОВИ ПРОГРАМУВАННЯ

C / C++ (gcc/gcc-c++ 2.96-85, 3.4, icc 5.0, 6.0, 7.0, 8.0), Fortran-77 (gcc-g77 2.96-85, 3.4, ifc 5.0, 6.0, 7.0, 8.0), Fortran-90/95 (ifc - 5.0, 6.0, 7.0, 8.0)

ЗАСОБИ АВТОМАТИЧНОГО РОЗПАРАЛЕЛЮВАННЯ ТА МОВИ ПАРАЛЕЛЬНОГО ПРОГРАМУВАННЯ

OpenMP, cilk 5.3.1, adaptor 7.0, mpC 2.2.0

ПРИКЛАДНІ ПРОГРАМИ

GAMESS-US, PCGAMESS, NWCHEM, GAUSSIAN2002, GROMACS, NAMD, AMBER, AUTODOCK, 3DDOCK, FLO+, MATHEMATICA, GNU OCTAVE, GATE.

ОСОБЛИВОСТІ КОНФІГУРАЦІЇ КЛАСТЕРА

Вузли кластера мають однакову структуру каталогів та містять однаковий набір програмного забезпечення, встановленого в одні й ті ж каталоги. Це дозволяє забезпечити максимальну незалежність вузлів та зменшити об'єм інформації, що передається мережею. Всі вузли кластеру мають однаковий простір імен, у тому числі імен та ідентифікаторі користувачів і груп, що забезпечується службою каталогів на базі LDAP.

Крім постійно працюючих вузлів до кластера можливе динамічне підключення будь-якої кількості додаткових вузлів. Нові вузли можуть бути бездисковими і завантажуватись через мережу.

2. Система керування ресурсами

Portable Batch System (PBS) – це комп'ютерне програмне забезпечення, яке виконує планування завдань. Його основне завдання - розподілити обчислювальні завдання, тобто пакетні завдання серед доступних обчислювальних ресурсів.

PBS дозволяє виконувати три основні дії:

- Додати завдання до черги;
- Видалити завдання із черги;
- Подивитись, де робота у черзі (статистика).

2.1. Torque

TORQUE - менеджер розподілення ресурсів для обчислювальних кластерів управлінням Linux та інших Unix-подібних операційних систем. Поширюється під вільною ліцензією OpenPBS Software License. TORQUE розробляється та підтримується спільнотою на базі проекту OpenPBS. Для менеджера існує більше 1200 патчів та розширень, написаних найбільшими організаціями та лабораторіями, серед яких US DOE, USC, PNLL та ін., це дозволяє досягти високого ступеня масштабованості та відмовостійкості менеджера як системи.

Основна функція TORQUE – розподіл обчислювальних завдань серед доступних обчислювальних ресурсів. TORQUE містить власний планувальник завдань, що визначає момент запуску завдань. Існує також сторонній планувальник завдань Maui, який має значно більшу

функціональність у порівнянні зі стандартним планувальником, і тому часто використовується спільно з TORQUE.

2.2. Maui

Maui cluster scheduler - планувальник завдань у паралельних та розподілених обчислювальних системах (кластерах). Зазвичай використовується спільно з менеджером розподілених ресурсів TORQUE.

Maui дозволяє вибирати різні політики планування, підтримує динамічну зміну пріоритетів, виняток. Все це покращує керованість та ефективність машин, починаючи від простих кластерів до суперкомп'ютерів (Рис 2.1).

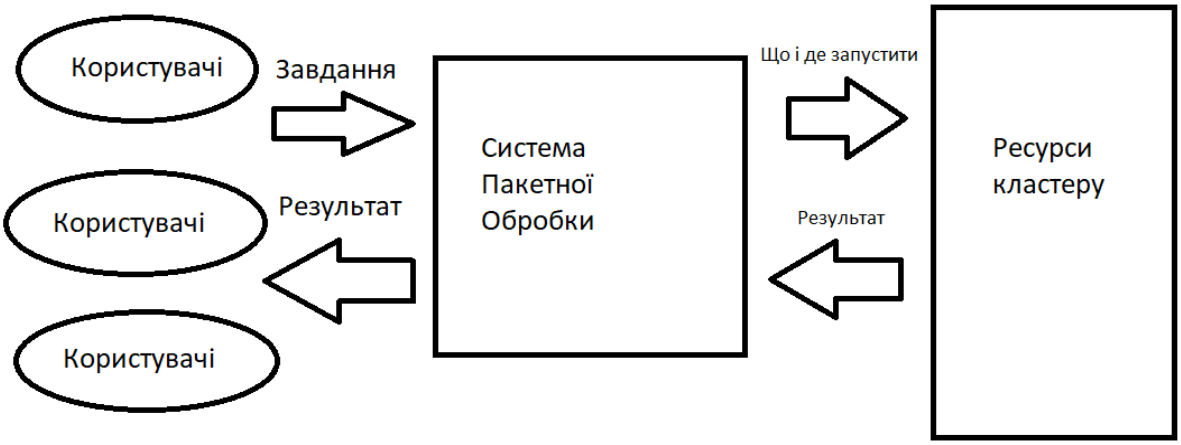

Рисунок 2.1. Схема планування

2.2.1. Важливі функції у Maui і Torque

Backfill — це оптимізація планування, яка дозволяє планувальнику краще використовувати доступні ресурси, запускаючи завдання. Коли Maui планує, він визначає пріоритети завдань у черзі відповідно до ряду факторів, а потім впорядковує завдання в відсортований список “спершу з найвищим пріоритетом”. Він запускає завдання по черзі, проходячи через список пріоритетів, доки не досягне завдання, яке не може почати. Оскільки всі вакансії та бронювання мають час початку та wallclock ліміт, Maui може визначити час завершення всіх завдань у черзі.

Fairshare дозволяє включати інформацію про використання ресурсів у доцільність роботи та пріоритетність рішень. Ця функція дозволяє адміністраторам сайту встановлювати цілі використання системи для користувачів, груп, облікових записів і класів. Адміністратори також можуть вказати часові рамки, протягом яких оцінюється використання ресурсів, щоб визначити, чи досягнута ціль. Параметри дозволяють сайтам вказувати показники використання, спосіб агрегування інформації та вплив стану справедливої частки на поведінку планування. Цілі Fairshare можна вказати для будь-яких облікових даних (наприклад, користувача, групи, класу тощо), на які адміністратори хочуть, щоб ця інформація вплинула.

Функція QoS дозволяє надавати особливу увагу різним класам робіт, користувачам, групам тощо. Кожен об’єкт QoS можна розглядати як контейнер особливих привілеїв, починаючи від винятків політики справедливого розподілу ресурсів до спеціального визначення пріоритетів завдань і закінчуючи доступом до спеціальної функціональності. Кожен об’єкт QoS також має широкий список доступу користувачів, груп та облікових записів, які можуть отримати доступ до цих привілеїв.

Standing reservation базується на можливостях попереднього бронювання, щоб сайт міг ефективно застосовувати розширені політики використання. Standing reservation надає додатковий набір можливостей, які зазвичай містяться в класі або архітектурі черги пакетної системи обслуговування. Наприклад, черги можна використовувати так, щоб дозволити лише певним типам завдань доступ до певних обчислювальних ресурсів. Крім того, деякі пакетні системи дозволяють налаштувати ці черги так, щоб вони дозволяли цей доступ лише в певний час дня або тижня. Постійне резервування надає ті самі можливості, але з більшою гнучкістю та ефективністю, ніж зазвичай у звичайній системі керування чергою.

Standing reservation забезпечує механізм, за допомогою якого сайт може виділяти певний блок ресурсів для спеціального використання на регулярній щоденній або щотижневій основі. Наприклад, вузол X може бути призначений для виконання завдань лише від користувачів у групі обліку щоп’ятниці з 16:00 до 22:00.

Maui дозволяє розставляти пріоритети вакансій на основі низки факторів, пов’язаних з роботою. Ці фактори розбиті на дворівневу ієрархію пріоритетних факторів і підфакторів, кожному з яких може бути присвоєна вага. Такий підхід надає адміністратору детальний, але простий контроль процесу вибору вакансій.

GRES (generic consumable resources). Кожного разу, коли завдання призначається обчислювальному вузлу, воно споживає один або кілька типів ресурсів. Стандартні ресурси, такі як процесор, пам'ять, диск, пропускна здатність мережевого адаптера та підкачка, автоматично відстежуються та споживаються. Однак у багатьох випадках додаткові ресурси можуть надаватися вузлами і споживатися роботами, які необхідно відстежувати. Мета цього відстеження може включати облік, виставлення рахунків або запобігання надмірного споживання ресурсів.

Загальні споживані ресурси можуть використовуватися для керування ліцензіями на програмне забезпечення, використанням вводу-виводу, пропускною здатністю, підключеннями до програм або будь-яким іншим аспектом більшого обчислювального середовища; вони можуть бути пов'язані з обчислювальними вузлами, мережами, системами зберігання даних або іншими реальними чи віртуальними ресурсами.

Цими додатковими ресурсами можна керувати, визначивши один або кілька загальних ресурсів. Першим кроком у визначенні загального ресурсу є назва ресурсу. Доступність загальних ресурсів може бути пов’язана з різними обчислювальними вузлами, а загальні вимоги до використання ресурсів можуть бути пов’язані з завданнями.

2.3. PBSPro

PBSPro або Portable Batch System Professional — це розподілене програмне забезпечення для керування робочим навантаженням, яке забезпечує уніфікований інтерфейс пакетної черги та керування завданнями для набору обчислювальних ресурсів (Рис 2.2).

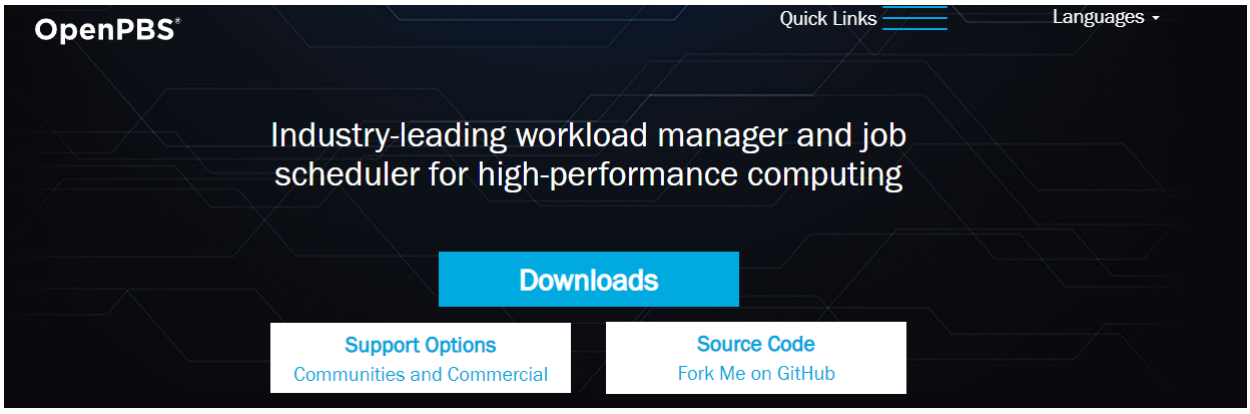

Рисунок 2.2. Сайт OpenPBS

PBS відповідає за управління ресурсами, планування завдань, оптимізацію суперкомп'ютера, програмування передачі повідомлень, паралельні обчислення та розподілені високопродуктивні обчислення.

PBS Professional надає багато функцій і переваг як для користувача комп'ютерної системи, так і для організації.

Спільний доступ до ресурсів у масштабі підприємства забезпечує прозоре планування завдань у будь-якій системі PBS будь-яким авторизованим користувачем. Вакансії можна надсилати з будь-якої клієнтської системи як локальної, так і віддаленої, перетинаючи домени, де це необхідно.

Кілька інтерфейсів користувача надають графічний інтерфейс користувача на додаток до стандартного інтерфейсу командного рядка для подання пакетних та інтерактивних завдань; запитувати роботу, чергу та статус системи; і моніторинг ходу роботи.

Parallel Job Support працює з бібліотеками паралельного програмування, такими як MPI, PVM і HPF. Програми можна запланувати для виконання на одному багатопроцесорному комп’ютері або в кількох системах. Взаємозалежність завдань дозволяє користувачеві визначати широкий діапазон взаємозалежностей між роботами. Такі залежності включають порядок виконання та виконання, обумовлене успіхом або невдачею іншого конкретного завдання (або набору завдань). Система моніторингу включає графічний інтерфейс користувача для моніторингу системи. Відображає статус вузла, розміщення роботи та інформацію про використання ресурсів як для автономних систем, так і для кластерів. Підтримка обчислювальної сітки надає технологію для мета-обчислювальних і обчислювальних сіток, включаючи підтримку Globus Grid Toolkit. Автоматичне вирівнювання навантаження надає численні способи розподілити навантаження між кластером машин на основі

конфігурації обладнання, доступності ресурсів, активності клавіатури та політики локального планування. Розподілена кластеризація дозволяє користувачам використовувати фізично розподілені системи та кластери навіть у глобальних мережах.

2.4. Причини переходу на систему PBSPro

Torque і maui – це програмне забезпечення, яке добре виконує свої функції, але морально застарілі. Вони не підлаштовані під нові види обладнання, вимагають старих компіляторів і бібліотек. PBSPro, на відміну від них, підтримується до цього часу, що дозволяє кластеру спокійно еволюціонувати, ставити нові бібліотеки і т.д.

Серед важливих функцій, які є у Maui і Torque у PBSPro виконує кожну з них, окрім QoS. Вона недоступна за замовчуванням, але її можна налаштувати самостійно.

2.5. Різниця між OpenPBS та PBSPro

Основна різниця між цими продуктами полягає у підтримці. OpenPBS – версія з відкритим кодом, що підтримується спільнотою, а PBSPro – має офіційну підтримку. Тобто використання OpenPBS є безкоштовним, але з нестабільною підтримкою.

Окремим пунктом можна зазначити, що PBSPro має рішення для освітніх закладів з підтримкою електронною поштою за річну плату.

3. Встановлення OpenPBS

3.1. Налаштовування середовища

Для встановлення OpenPBS на систему Linux, було виконано початкове налаштування системи, а саме встановлено необхідні пакети для OpenPBS (рис 3.1)

```
dnf install -y gcc make rpm-build libtool hwloc-devel \
  libX11-devel libXt-devel libedit-devel libical-devel \
  ncurses-devel perl postgresql-devel postgresql-contrib python3-devel tcl-devel \
  tk-devel swig expat-devel openssl-devel libXext libXft \
  autoconf automake gcc-c++
```

Рисунок 3.1. Пакети необхідні для встановлення

Ці пакети відповідають за коректне встановлення та побудови OpenPBS. Окрім цього, необхідно було встановити пакети для роботи (рис 3.2)

```
yum install -y expat libedit postgresql-server postgresql-contrib python3 \
  sendmail sudo tcl tk libical
```

Рисунок 3.2. Пакети необхідні для роботи

Серед них виділяється postgresql-server. Він стане базою даних для кластеру.

3.2. Завантаження

Для встановлення необхідно було завантажити архів з сайту github.com, версію можна вибрати опціонально, на даний момент v20.0.1.tar.gz є найсучаснішою. Це необхідно робити як (не-рут)користувач

https://github.com/openpbs/openpbs/archive/refs/tags/v20.0.1.tar.gz

У подальшому потрібно розпакувати архів

tar -xpvf v20.0.1.tar.gz

Та зайти у розпакований каталог

cd openpbs-20.0.1

У цьому каталозі можна побачити файли для скрипту (Рис 3.1)

```
drwxrwxr-x. 13 lozhka lozhka   4096 тра 24 21:30 .
drwx------. 17 lozhka lozhka   4096 тра 24 22:05 ..
-rw-rw-r--.  1 lozhka lozhka  59230 тра 24 21:29 aclocal.m4
drwxrwxr-x.  2 lozhka lozhka    168 чер 13  2020 .appveyor
-rw-rw-r--.  1 lozhka lozhka   2148 чер 13  2020 .appveyor.yml
-rwxrwxr-x.  1 lozhka lozhka   1877 чер 13  2020 autogen.sh
drwxr-xr-x.  2 lozhka lozhka    150 тра 24 21:29 autom4te.cache
drwxrwxr-x.  2 lozhka lozhka    253 тра 24 21:30 buildutils
drwxrwxr-x.  2 lozhka lozhka    170 чер 13  2020 ci
-rw-rw-r--.  1 lozhka lozhka    605 чер 13  2020 CODE_OF_CONDUCT
-rw-rw-r--.  1 lozhka lozhka 194516 тра 24 21:30 config.log
-rwxrwxr-x.  1 lozhka lozhka  75986 тра 24 21:30 config.status
-rwxrwxr-x.  1 lozhka lozhka 608903 тра 24 21:29 configure
-rw-rw-r--.  1 lozhka lozhka   8452 чер 13  2020 configure.ac
-rw-rw-r--.  1 lozhka lozhka    750 чер 13  2020 CONTRIBUTING.md
-rw-rw-r--.  1 lozhka lozhka   1584 чер 13  2020 COPYRIGHT
drwxrwxr-x.  5 lozhka lozhka     96 тра 24 21:30 doc
drwxrwxr-x.  2 lozhka lozhka     72 чер 13  2020 .github
-rw-rw-r--.  1 lozhka lozhka   4494 чер 13  2020 .gitignore
-rw-rw-r--.  1 lozhka lozhka   9465 чер 13  2020 INSTALL
-rwxrwxr-x.  1 lozhka lozhka 293050 тра 24 21:30 libtool
-rw-rw-r--.  1 lozhka lozhka  49638 чер 13  2020 LICENSE
drwxrwxr-x.  2 lozhka lozhka   4096 тра 24 21:29 m4
-rw-rw-r--.  1 lozhka lozhka  30581 тра 24 21:30 Makefile
-rw-rw-r--.  1 lozhka lozhka   1825 чер 13  2020 Makefile.am
-rw-rw-r--.  1 lozhka lozhka  29996 тра 24 21:29 Makefile.in
-rw-rw-r--.  1 lozhka lozhka   2336 чер 13  2020 openpbs-rpmlint
-rw-rw-r--.  1 lozhka lozhka  18092 тра 24 21:30 openpbs.spec
-rw-rw-r--.  1 lozhka lozhka  18086 чер 13  2020 openpbs.spec.in
-rw-rw-r--.  1 lozhka lozhka   1584 чер 13  2020 PBS_License.txt
-rw-rw-r--.  1 lozhka lozhka   4594 чер 13  2020 README.md
drwxrwxr-x. 15 lozhka lozhka    247 тра 24 21:30 src
drwxrwxr-x.  4 lozhka lozhka     83 тра 24 21:30 test
drwxrwxr-x.  2 lozhka lozhka     46 чер 13  2020 .travis
-rw-rw-r--.  1 lozhka lozhka   2796 чер 13  2020 .travis.yml
-rw-rw-r--.  1 lozhka lozhka  13879 чер 13  2020 valgrind.supp
drwxrwxr-x.  5 lozhka lozhka     84 чер 13  2020 win_configure
```

Рисунок 3.3. Файли, які знаходяться у каталозі для встановлення

3.3. Створення скрипту для встановлення

Для створення скрипту необхідно було ввести команду

./autogen.sh

Якщо при роботі випадково зміниться configure.ac або часові мітки у будь яких файлах, що створяться автоматично, то можна буде їх згенерувати заново повторним запуском autogen.sh

Щоб побачити доступні параметри збірки, необхідно ввести

./configure --help

У параметрах можна подивитись документацію, змінити каталог встановлення, встановити дозволи для каталогів і файлів, які будуть встановленні, налаштувати хоста, специфікації, прапори.

Для того, щоб відкрити конфігуратор необхідно було ввести

./configure --prefix=/opt/pbs

Коли налаштування виконано, необхідно було запустити скрипт командою

Make

3.4. Встановлення OpenPBS за допомогою скрипта

Коли скрипт згенеровано, можна встановлювати OpenPBS командою

sudo make install

Необхідно було переконатись, що OpenPBS встановився коректно. Якщо це так, то залишилось запустити post-install скрипт

sudo /opt/pbs/libexec/pbs_postinstall

3.5. Конфігурування OpenPBS

Необхідно було відредагувати /etc/pbs.conf, щоб налаштувати служби pbs, які будуть працювати. Якщо PBS встановлюється на одну систему, то необхідно було встановити параметр PBS_START_MOM з 0 на 1(рис 3.2) .

Щоб це зробити, потрібно було виконати

sudo vi /etc/pbs.conf

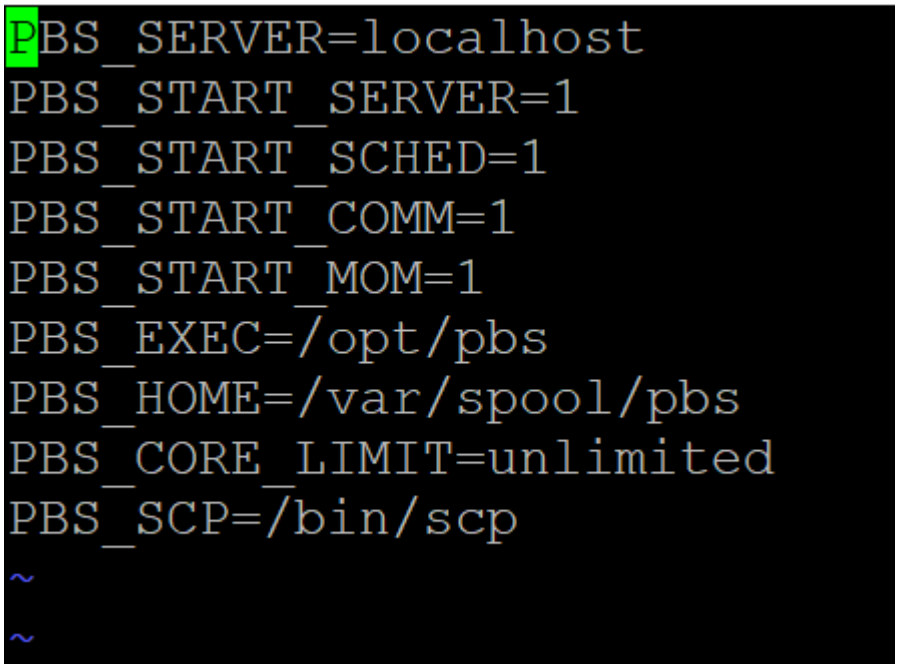

Рисунок 3.2. Вміст файлу pbs.conf (PBS_START_MOM вже встановлений на 1)

Деякі права доступу до файлів необхідно було змінити, щоб додати привілеї SUID

sudo chmod 4755 /opt/pbs/sbin/pbs_iff /opt/pbs/sbin/pbs_rcp

3.6. Запуск сервісів PBS

Коли це все виконано, можна запустити сервіси PBS

sudo /etc/init.d/pbs start

Всі налаштовані служби PBS тепер повинні працювати. Оновлювати змінні PATH і MANPATH можна знайшовши відповідні профілі PBS, або виходити з системи та заходити назад.

4. Налаштування OpenPBS

Для коректної і зручної роботи OpenPBS необхідно було налаштувати сервер, так як налаштований на даний момент кластер Київського Національного Університету.

4.1 Створення черг для роботи кластеру

Для створення черги використовується команда –

create queue назва_черги

Після створення, черга налаштовується:

set queue назва_черги queue_type = тип

– ця команда дозволяє обрати тип для черги.

set queue назва_черги Priority = число

– вказує пріоритет черги.

set queue назва_черги resources_min.walltime = час

-вказує мінімальні часові рамки для завдання, що подаються в чергу

set queue назва_черги resources_max.walltime = час

-вказує максимальні часові рамки для завдання, що подаються в чергу

set queue назва_черги resources_default.walltime = час

-визначає вимоги до ресурсів за замовчуванням для завдань, які надіслані на сервер

set queue назва_черги resources_max.cput = час

-визначає максимальний процесорний час на виконання завдання

set queue назва_черги resources_default.cput = час

-вказує вимоги для процесорного часу на виконання завдання

set queue назва_черги resources_min.nodect = число

-вказує мінімальне значення вузлів для виконання завдання

set queue назва_черги resources_max.nodect = число

-вказує максимальне значення вузлів для виконання завдання

set queue назва_черги resources_default.nodes =число

-вказує вимоги до кількості вузлів за замовчуванням

set queue назва_черги resources_default.file = розмір_файлу

-вказує вимоги до розміру файлу завдання за замовчуванням

set queue назва_черги resources_default.mem = кількість_пам’яті

-вказує вимоги до кількості пам’яті на завдання за замовчуванням

set queue назва_черги max_queuable = число

-вказує максимальну кількість завдань, які можуть одночасно знаходитись в черзі

set queue назва_черги max_running = число

-вказує максимальну кількість завдань у черзі, які виконуються

set queue назва_черги route_destinations = назва_черги

-визначає потенційні черги, на які перенаправляються завдання, які надіслані на дану чергу

set queue назва_черги route_retry_time = число

-визначає час на спробу перенаправлення, після невдалого перенаправлення

set queue назва_черги route_lifetime = число

-визначає час тривалості перенаправлення, на іншу чергу

set queue назва_черги enabled = True\False

-вказує чи приймає черга нові завдання

set queue назва_черги started = True\False

-вказує чи дозволено виконання завдань у черзі

4.2. Налаштування черг згідно кластеру Київського Національного Університету

create queue stereo_short

set queue stereo_short queue_type = Execution

set queue stereo_short Priority = 6

set queue stereo_short resources_max.walltime = 64:00:00

set queue stereo_short resources_min.nodect = 1

set queue stereo_short enabled = True

set queue stereo_short started = True

---

create queue quick_serial

```
set queue quick_serial queue_type = Execution
set queue quick_serial Priority = 10
set queue quick_serial resources_max.nodect = 1
set queue quick_serial resources_max.walltime = 00:30:00
set queue quick_serial resources_default.walltime = 00:30:00
set queue quick_serial enabled = True
set queue quick_serial started = True
```

```
create queue interactive
set queue interactive queue_type = Execution
set queue interactive resources_default.walltime = 00:05:00
set queue interactive enabled = True
set queue interactive started = True
```

```
create queue mono_short
set queue mono_short queue_type = Execution
set queue mono_short Priority = 5
set queue mono_short resources_max.nodect = 1
set queue mono_short resources_max.walltime = 36:00:00
set queue mono_short resources_min.walltime = 00:30:00
set queue mono_short resources_default.walltime = 12:00:00
set queue mono_short enabled = True
set queue mono_short started = True
```

```
create queue moldyngrid
```

set queue moldyngrid queue_type = Execution

set queue moldyngrid Priority = 2

set queue moldyngrid max_queuable = 50

set queue moldyngrid max_running = 10

set queue moldyngrid resources_max.cput = 34560:00:00

set queue moldyngrid resources_max.walltime = 1440:00:00

set queue moldyngrid enabled = True

set queue moldyngrid started = True

---

create queue grid

set queue grid queue_type = Execution

set queue grid Priority = 2

set queue grid max_queuable = 100

set queue grid max_running = 50

set queue grid resources_max.cput = 17280:00:00

set queue grid resources_max.walltime = 720:00:00

set queue grid resources_default.walltime = 72:00:00

set queue grid enabled = True

set queue grid started = True

---

create queue mono_long

set queue mono_long queue_type = Execution

set queue mono_long Priority = 4

set queue mono_long resources_max.nodect = 1

set queue mono_long resources_min.walltime = 36:00:00

set queue mono_long resources_default.walltime = 72:00:00

set queue mono_long enabled = True

set queue mono_long started = True

---

create queue stereo_long

set queue stereo_long queue_type = Execution

set queue stereo_long Priority = 6

set queue stereo_long resources_min.nodect = 1

set queue stereo_long enabled = True

set queue stereo_long started = True

---

create queue alien_batch

set queue alien_batch queue_type = Execution

set queue alien_batch resources_default.file = 4095mb

set queue alien_batch resources_default.mem = 3400mb

set queue alien_batch resources_default.nodes = 1:ppn=1

set queue alien_batch resources_default.walltime = 72:00:00

set queue alien_batch enabled = True

set queue alien_batch started = True

---

create queue grid_rt

set queue grid_rt queue_type = Execution

set queue grid_rt Priority = 10

set queue grid_rt max_running = 4

```
set queue grid_rt resources_max.cput = 02:00:00
set queue grid_rt resources_max.nodect = 2
set queue grid_rt resources_max.walltime = 01:00:00
set queue grid_rt resources_default.cput = 00:30:00
set queue grid_rt resources_default.nodes = 1:ppn=1
set queue grid_rt resources_default.walltime = 00:30:00
set queue grid_rt enabled = True
set queue grid_rt started = True
```

```
create queue default
set queue default queue_type = Route
set queue default route_destinations = mono_short
set queue default route_destinations += mono_long
set queue default route_destinations += stereo_short
set queue default route_destinations += stereo_long
set queue default route_destinations += quick_serial
set queue default route_destinations += interactive
set queue default route_lifetime = 86400
set queue default enabled = True
set queue default started = True
```

```
create queue alien
set queue alien queue_type = Route
set queue alien resources_max.cput = 72:00:00
```

```
set queue alien resources_max.walltime = 72:00:00
set queue alien resources_default.cput = 72:00:00
set queue alien resources_default.walltime = 72:00:00
set queue alien route_destinations = grid_rt
set queue alien route_destinations += alien_priv
set queue alien route_destinations += alien_batch
set queue alien route_retry_time = 10
set queue alien route_lifetime = 0
set queue alien enabled = True
set queue alien started = True
```

```
create queue alien_priv
set queue alien_priv queue_type = Execution
set queue alien_priv max_queuable = 8
set queue alien_priv max_running = 7
set queue alien_priv resources_default.file = 4095mb
set queue alien_priv resources_default.mem = 3400mb
set queue alien_priv resources_default.nodes = 1:ppn=1
set queue alien_priv resources_default.walltime = 72:00:00
set queue alien_priv enabled = True
set queue alien_priv started = True
```

4.3 Налаштування атрибутів серверу

Окрім черг необхідно налаштувати сервер згідно з кластером Київського Національного університету

set server scheduling = True

-контролює, чи буде сервер запитувати планування завдань зі стороні PBS. Так як, стоїть «Так», то планувальник буде викликатись за необхідністю

set server acl_host_enable = True

-атрибут, який вказує серверу – використовувати список acl_hosts, чи ні. Так як, стоїть «Так», то вони будуть використовуватись

set server acl_hosts = localhost

set server acl_hosts += bs.cluster.univ.kiev.ua

set server acl_hosts += *.cluster.univ.kiev.ua

set server acl_hosts += arc.univ.kiev.ua

-атрибут, який вказує на список хостів, які можуть користуватись послугами сервера

set server managers = root@*.cluster.univ.kiev.ua

-атрибут, який вказує на хостів, які можуть користуватись правами менеджера

set server operators = root@*.cluster.univ.kiev.ua

-атрибут, який вказує на хостів, які можуть користуватись правами пакетного оператора

set server default_queue = default

-атрибут, який вказує чергу, на яку буде надходити завдання у випадку, коли у ньому не вказано конкретну чергу.

set server log_events = 511

-атрибут, який визначає, який тип подій буде регіструватись

set server mail_from = pbs

-атрибут, який вказує ідентифікатор пошти, з якої буде надходити повідомлення користувачам

set server query_other_jobs = True

-атрибут, який визначає можливість користувачів вибирати і дивитись статус не своїх завдань

set server scheduler_iteration = 60

-атрибут, який визначає час в секундах між спробами сервера планувати роботу. На кожній ітерації планувальник перевіряє доступні ресурси и завдання, щоб перевірити можливість ініціювання роботи нової задачі. Перевірка також відбувається, коли певна робота завершається або нове завдання поміщається в чергу.

4.4 Налаштування планувальника

При налаштуванні планувальника можна задіяти певні функції, що покращують роботу кластеру. Більшість з налаштувань виконуються у файлі /var/spool/pbs/sched_priv/ sched_config

4.4.1 Preemptive Scheduling

Ця функція дозволяє ставити на паузу виконання запущених завдань для задач з вищим пріоритетом. Це робиться шляхом присвоєння одній черзі пріоритету вище, ніж у інших і вказуванням планувальнику порогу для роботи функції. Якщо завдання знаходиться у такій черзі і, при його надходженні до планувальника, воно не може бути запущеним, то планувальник знайде завдання з нижчим пріоритетом і призупинить його виконання.

Є п’ять параметрів цієї функції:

preemptive_sched – атрибут, який вказує, які черги можуть витісняти інших, при більшому пріоритеті.

preempt_queue_prio – атрибут, який вказує, який поріг пріоритету мінімальний, для витіснення.

preempt_suspend, preempt_checkpoint, preempt_requeue – ці атрибути, вказують на те, яким чином буде відбуватись зупинка менш пріоритетних черг. Їх можна виставляти незалежно один від одного, але в такому випадку вони будуть виконуватись у порядку – зупинка, контрольна точка, очікування запиту.

Налаштування на кластері:

preemptive_sched: true ALL

preempt_queue_prio: 1

preempt_suspend: TRUE

preempt_checkpoint: FALSE

preempt_requeue: FALSE

Ці налаштування означають, що при надходженні завдання, яке має пріоритет на 1 вище, ніж те, яке виконується – воно його замінить для роботи. Коефіцієнт, черги та спосіб заміщення можна змінювати за необхідністю.

4.4.2 Fair Share

Функція Fair Share дозволяє розподілити ресурси кластеру між групами користувачів. Користувачі знаходяться у груповому файлі, він зчитується і створюється дерево, яке складається із груп і користувачів (групи можуть мати у собі інші групи). Ці групи мають пріоритети і ресурси розподіляються між ними.

Файл з групами знаходиться в

/var/spool/pbs/sched_priv/resource_group

Основні параметри Fair Share:

Name – ім’я специфічного користувача (або групи)

unique_id – атрибут, який вказує на унікальний ідентифікатор для користувача або групи

parent_group – Ім’я групи, до якої належить користувач, група root створюється автоматично

shares – атрибут, що вказує на пріоритет, який користувач має у вказаній групі. Якщо користувача немає у групах, то такому користувачу можна задати пріоритет.

Налаштування Fair Share у resource_group:

grp1 50 root 10

grp2 51 root 20

usr1 52 root 5

usr2 53 grp1 10

usr3 54 grp2 10

usr4 55 grp1 5

usr5 56 grp2 5

Результатом цього налаштування буде таким, що usr1 матиме 15% потужності, usr2 - 19%, usr3 - 38%, usr4 - 10%, usr5 – 18%

Налаштування Fair Share у sched_priv:

fair_share: true ALL – вмикає функцію fair_share

unknown_shares: 1 - задає пріоритети невідомим користувачам

fairshare_usage_res: cput – атрибут, який задає ресурс, який буде розподілятись

fairshare_entity: euser – атрибут, який буде використовуватись для справедливого розподілу ресурсів, це може бути користувач, група і, навіть, черга.

fairshare_decay_time: 48:00:00 – атрибут, який вказує інтервал між зруйнуванням і створенням нового дерева розподілу.

За допомогою цього налаштування у pbspro зникає необхідність у функції QoS. Ці групи і є класами.

4.4.3 STARVING JOB OPTIONS

У випадку коли є завдання з низьким пріоритетом і завдання з більшим пріоритетом постійно надходять, воно може затриматись в черзі навічно. Для цього є налаштування «голодних» завдань. При досягненні певного часу простою, пріоритет цих завдань стає вище, і вони отримують шанс на виконання.

help_starving_jobs: true ALL – атрибут, який вмикає функцію, допомоги «голодним» завданням

max_starve: 24:00:00 – атрибут, який задає після якого часу завдання стає «голодним»

Також складовою функції STARVING JOB є backfill. Ця функція ввімкнена за замовчуванням, при потребі можна додатково її визначити за допомогою команди backfill: True all . Але вона працює не так, як у maui (точніше не зовсім так). Планувальник відслідковує, яке завдання з «голодних» повинне мати виконане наступним, але, крім цього, достатньо мале завдання поміщається в слот «невелика робота» та виконується, доки це не змінить час виконання більш великого «голодного завдання». Це працює схоже на backfill у Maui, але там не потрібно чекати, доки завдання стануть «голодними».

4.4.4 Resources

Ця функція дозволяє налаштувати ліміти ресурсів, яким повинен дотримуватись PBS. Тобто PBS буде слідкувати за вказаними ресурсами і не запускати в роботу завдання, якщо якийсь із них буде завантаженим

resources: "ncpus, mem, arch, host, vnode, aoe, eoe"

Ця функція є аналогом GRES. Для додавання нових типів ресурсів можна використати файл $PBS_HOME/server_priv/resourcedef file.

4.4.5. Primetime

Ця функція дозволяє розподіляти час на prime і nonprime. При використанні цієї функції, завдання з черг, які вказані як prime можуть виконуватись тільки у цей визначений час. Черги які визначені як nonprime – навпаки. Ця функція дозволяє, наприклад, виділити сервер на роботу певної черги протягом певного часу. Може використовуватись як аналог функції Standing reservation

Налаштування:

backfill_prime: false ALL – атрибут, який вказує на можливість переходу завдань із Primetime в nonprimetime і навпаки.

primetime_prefix: p_ - черги з таким атрибутом можуть бути запущені тільки в Primetime

nonprimetime_prefix: np_ - черги з таким атрибутом можуть бути запущені тільки в nonprimetime

Всі інші черги, які не мають таких префіксів, запускаються в будь який час.

У файлі /var/spool/pbs/sched_priv/holidays можна налаштувати час Primetime

YEAR 2022 – вибрати рік

weekday 0600 1730 – вибирає будній день та час початку primetime і закінчення (початок nonprimetime)

saturday none all – вибирає субботу і позначає, що вона повністю nonprimetime

sunday none all вибирає неділю і позначає, що вона повністю nonprimetime

1 Jan 1 New Year's Day – можна вибирати день у році і назначати його nonprimetime

5. Тестування роботи

Після налаштування черг, сервера та планувальника була виконана робота з тестування роботи кластеру. Перевірена можливість задавання роботи у різні черги, виконання цих робіт на кластері, роботу планувальника та його функцій, таких як Fair Share, primetime, starving jobs та ін. При тестуванні була підтверджена робота цих функцій, які необхідні для роботи на кластері Київського національного університету імені Тараса Шевченка.

5.1. Запуск задачі

Для цього було запущено завдання для перевірки роботи сервера, завдання було виконано, працездатність сервера підтверджена(Рис 5.1). Ми можемо побачити, що запущений скрипт був відправлений в чергу під номером 32.

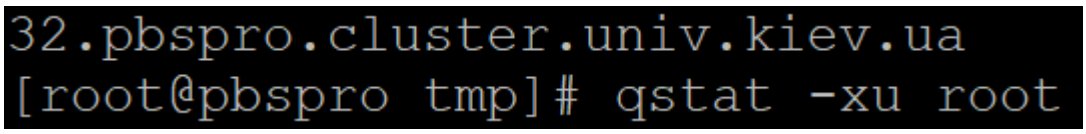

Рисунок 5.1 Запуск скрипту

Для відправлення завдання на конкретну чергу використовувало -q і назва черги (Рис 5.2). Можемо побачити, що завдання відправлено у чергу під номером 31

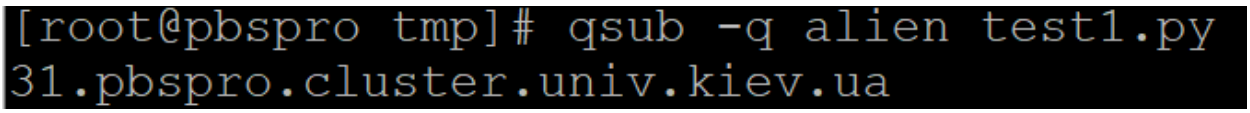

Рисунок 5.2 Запуск скрипту у черзі

Для перегляду робіт можемо скористатись функцією qstat. Так як, скрипт простий, він буде виконанний дуже швидко , то qstat необхідно запустити з параметром -xu і іменем користувача. Це дозволить переглянути історію роботи користувача разом з виконанними роботами (Рис 5.3). Так як, черга alien, яку було вказано для роботи - перенаправляюча, то задача під номером 31 успішно перейшла на grid_rt. Також робота, для якокої нічого не було вказано, була за налаштуваннями черги default переведена на mono_short

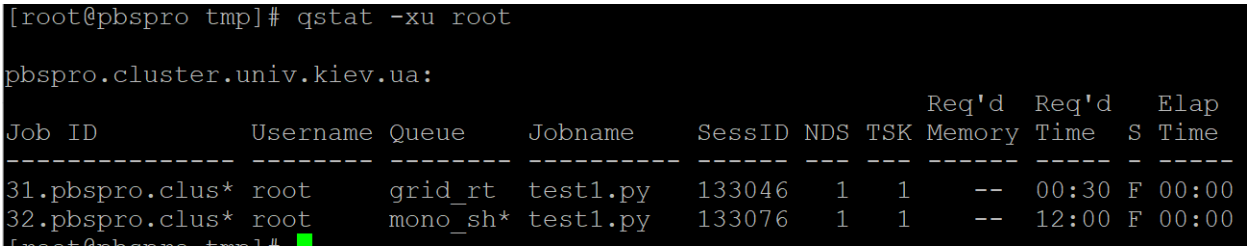

Рисунок 5.3. Список виконанних скриптів

5.2. Функція Primetime

Для перевірки функції Primetime була створена черга np_test, яка має параметри виконнаня роботи тільки в не робочий час. Скрипт на цій черзі запущений в робочий час (Рис 5.4)

```
[root@pbspro tmp]# qsub -q np_test test1.py
33.pbspro.cluster.univ.kiev.ua
```

Рисунок 5.4 Запуск скрипта для перевірки функції Primetime

При запуску qstat -f (атрибут -f побачити більше інформації) ми отримуємо вивід (Рис 5.5), на якому бачимо, що виконання не почалось. У пункті comment вказано, що робота почне виконуватись тільки в неробочий час.

```
euser = root
egroup = root
queue_rank = 1653904203267
queue_type = E
comment = Not Running: Job will run in nonprimetime only
etime = Mon May 30 12:50:03 2022
Submit_arguments = -q np_test test1.py
project = _pbs_project_default
Submit_Host = pbspro.cluster.univ.kiev.ua
```

Рисунок 5.5 Інформація про роботу в робочий час

5.3. Функція Preemptive Scheduling

Для перевірки функції Preemptive Scheduling необхідно запустити певну задачу, яка буде виконуватись деякий час (Рис5.6)

```
[root@pbspro tmp]# echo "sleep 60" | qsub
45.pbspro.cluster.univ.kiev.ua
```

Рисунок 5.6 Запуск довгої задачі

Після якої запускаємо швидкий скрипт на чергу з вищим пріоритетом (Рис 5.7)

```
[root@pbspro tmp]# qsub -q quick_serial test1.py
46.pbspro.cluster.univ.kiev.ua
```

Рисунок 5.7 Запуск скрипту на чергу з вищим пріоритетом

Виконуємо команду qstat для перевірки планувальника (Рис 5.8). Можна побачити, що скрипта під номером 46 немає. Завдання під номером 33 залишилось ще з минулого прикладу. Це сталося через те, що швидкий скрипт з більшим пріоритетом виконався раніше за допомогою функції Preemptive Scheduling.

```
[root@pbspro tmp]# qstat
Job id            Name             User             Time Use S Queue
----------------  ---------------- ----------------  -------- - -----
33.pbspro         test1.py         root                    0 Q np_test
45.pbspro         STDIN            root             00:00:00 R mono_short
```

Рисунок 5.8 Перевірка черг

5.4. Функція Standing reservation

Для тестування Standing reservation необхідно створити цей резерв (Рис5.9). Часові рамки з 15:00 до 17:00 для користувача lozhka. Цей резерв має назву test.

```
[lozhka@pbspro tmp]$ pbs_rsub -R 2000 -E 2100 -u lozhka -N test
R61.pbspro.cluster.univ.kiev.ua UNCONFIRMED
```

Рисунок 5.9 Створення Standing reservation

Для перевірки використовуємо pbs_rstat і серед інших, бачимо наш резерв(Рис 5.10). Окрім цього можна побачити, що резервувати можна потижнево. Окрім цього можна резервувати не увесь сервер, а окремі вузли.

```
[lozhka@pbspro tmp]$ pbs_rstat
Resv ID    Queue    User     State            Start / Duration / End

--------------------------------------------------------------------------
R53.pbspro R53      lozhka@p CO           Tue 15:00 / 7200 / Tue 17:00
R57.pbspro R57      lozhka@p CO           Tue 12:00 / 7200 / Tue 14:00
R61.pbspro R61      lozhka@p RN         Today 20:00 / 3600 / Today 21:00
```

Рисунок 5.10 Таблиця резервів

Тепер заходимо від імені рута і запускаємо скрипт(Рис 5.11)

```
[root@pbspro tmp]# qsub test1.py
62.pbspro.cluster.univ.kiev.ua
```

Рисунок 5.11 Запуск скрипта

При перевірці за допомогою команди qstat, ми бачимо, що завдання не починає виконуватись (Рис 5.12). Це відбувається через те, що на цей час сервер зарезервований для користувача lozhka.

```
[root@pbspro tmp]# qstat
Job id            Name             User             Time Use S Queue
----------------  ---------------- ----------------  -------- - -----
62.pbspro         test1.py         root                     0 Q mono_short
[root@pbspro tmp]#
```

Рисунок 5.12 Завдання знаходиться у черзі.

5.5. Функція FairShare

Для перевірки роботи FairShare можемо використати команду pbsfs -t (Рис 5.13). В результаті виконання цієї команди на екран виводиться дерево FairShare, яке було налаштовано вище.

```
[root@pbspro sched_priv]# pbsfs -t
 TREEROOT(0)
     usr1(52)
     grp2(51)
          usr5(56)
          usr3(54)
     grp1(50)
          usr4(55)
          lozhka(53)
     unknown(1)
```

Рисунок 5.13 Дерево FairShare

Ми бачимо, що від кореня розходяться гілки usr1, grp1, grp2 і unknown. В файлі, який налаштований як показувалось вище вказані пріоритети. При одночасній роботі користувачів з цих груп, пріоритети будуть розподілені як вказано у файлі resource_group.

6. Результат роботи

У результаті виконання даної роботи була протестована можливість встановлення системи керування ресурсами PBSPro (OpenPBS) на обчислювальний кластер Київського національного університету імені Тараса Шевченка. Була протестована можливість налаштування серверу згідно потреб університету. Були створені черги для роботи кластеру. Перевірена можливість налаштування планувальника. Було досліджено, що основні вимоги для переходу на нову систему, а саме існування таких функцій як Fair Share, Preemptive Scheduling та ін., виконані. Була проведена робота з тестування роботи серверу.

Результати тестування показали, що сервер є працездатним, він виконує завдання, додаткові функції – працюють (Таб 6.1).

Таблиця. 6.1 Порівнянн функції планувальника MAUI та планувальника OpenPBS

Функція	Стара версія	Нова версія	Показана
розширені пріоритети	В наявності	-	-
standing reservations	В наявності	В наявності.	Тест у розділі 5.4
fair share	В наявності	В наявності.	Тест у розділі 5.5

backfill	В наявності	-	Працює не так як у maui. Пояснення у розділі 4.4.3
Primetime	-	Є в наявності.	Тест у розділі 5.2
GRES	В наявності	Є в наявності, як resources.	Показана у розділі 4.4.4
QOS	В наявності	-	-
STARVING JOB	-	Є в наявності.	Показана у розділі 4.4.3
Preemptive Scheduling	В наявності	Є в наявності	Тест у розділі 5.3

Аналіз результатів тестування говорить про те, що OpenPBS виконує свої функції та може бути встановлений на кластер Київського національного університету імені Тараса Шевченка.

Висновки

1. Аналіз документації показав, що системи керування ресурсами maui і torque мають аналоги функціоналу у системах PBSPro та OpenPBS: fairshare, standing reservations, QOS, Priorities, Classes

2. Системи PBSPro та OpenPBS можуть бути виткористані на кластері Київського національного унівреситету імені Тараса Шевченка замість систем torque+maui

3. OpenPBS має відкритий вихідний код і є вільною для використання на відміну від PBSPro, що є перевагою для використання на Київського національного унівреситету імені Тараса Шевченка

4. Є можливість встановлення OpenPBS на сервер з системою Centos та налаштування згідно потреб Київського національного університету імені Тараса Шевченка

5. Результати тестування функціоналу показали працездатність серверу і можливість подальшого встановлення OpenPBS на кластері Київського національного університету імені Тараса Шевченка. Всі функції, окрім QoS, backfill та розширенні пріоритети, є в наявності для переходу на нову систему.

Перелік посилань

1. NASA. Portable Batch System (PBS): Overview [Електронний ресурс] – Режим доступу до ресурсу: https://www.nas.nasa.gov/hecc/support/kb/portable-batch-system-(pbs)-overview_126.html Online: дата 05.05.2022 рік
2. Scientific Programming Team. Научитесь использовать планировщик заданий PBS Pro[Електронний ресурс] – Режим доступу до ресурсу: https://learn.scientificprogramming.io/learn-to-use-pbs-pro-job-scheduler-ffd9c0ad680d Online: дата 05.05.2022 рік
3. Softpanorama. Установка відкритої версії PBSpro[Електронний ресурс]] – Режим доступу до ресурсу: https://softpanorama.org/HPC/PBS_and_derivatives/PBSpro/Installation/index.shtml Online: дата 05.05.2022 рік
4. Icybcluster. Загальні відомості про обчислювальні кластери[Електронний ресурс]– Режим доступу до ресурсу: http://icybcluster.org.ua/index.php?lang_id=2&menu_id=6/ Online: дата 07.05.2022 рік
5. Information and Computer Center. Про кластер[Електронний ресурс] – Режим доступу до ресурсу: http://cluster.univ.kiev.ua/ukr/ Online: дата 07.05.2022 рік
6. Openpbs. What is OpenPBS®?[Електронний ресурс] – Режим доступу до ресурсу https://www.openpbs.org/ Online: дата 07.05.2022 рік

7. Миколай Лазаревич. Система пакетной обработки заданий torque Руководство пользователя[Електронний ресурс] – Режим доступу до ресурсу

   http://docplayer.com/27232138-Sistema-paketnoy-obrabotki-zadaniy-torque-rukovodstvo-polzovatelya.html Online: дата 09.05.2022 рік

8. Kmbioc. OpenPBS Users Manual[Електронний ресурс] – Режим доступу до ресурсу

   http://kmbioc.kjtj.cas.cn/xwdt/tzgg/201303/P020130322635594695720.pdf Online: дата 09.05.2022 рік

9. Y. Joanna Wong. Job Scheduling with Maui and Torque[Електронний ресурс] – Режим доступу до ресурсу

   https://biostat.wustl.edu/sysadmin/UTB_1_MauiTorque.pdf Online: дата 09.05.2022 рік

10. Altair. Altair PBS Professional 2021.1.1 Administrator's Guide[Електронний ресурс] – Режим доступу до ресурсу

    https://help.altair.com/2021.1.1/PBSProfessional/PBSAdminGuide2021.1.1.pdf Online: дата 11.05.2022 рік

11. hirenvadalia. Open Source License for OpenPBS and PBS Professional[Електронний ресурс] – Режим доступу до ресурсу

    https://github.com/openpbs/openpbs/blob/master/LICENSE Online: дата 11.05.2022 рік

12. Altair. PBSPro User Guide[Електронний ресурс] – Режим доступу до ресурсу

    https://www3.physnet.uni-hamburg.de/physnet/PBSproUG.pdf Online: дата 11.05.2022 рік

13. Altair. PbSPro Administrator Guide[Електронний ресурс] – Режим доступу до ресурсу

https://secure.altair.com/docs/PBSproAG_Print.pdf Online: дата 15.05.2022 рік

14. NASA. PBS Job Queue Structure[Електронний ресурс] – Режим доступу до ресурсу

https://www.nas.nasa.gov/hecc/support/kb/pbs-job-queue-structure_187.html Online: дата 15.05.2022 рік

15. Altair Engineering. PBS Professional 12.2 User`s Guide[Електронний ресурс] – Режим доступу до ресурсу

https://euler.cemeai.icmc.usp.br/images/Manuais/PBSProUserGuide12.2.pdf Online: дата 15.05.2022 рік

16. Adaptivecomputing. Server Parameters[Електронний ресурс] – Режим доступу до ресурсу

https://docs.adaptivecomputing.com/torque/4-0-2/Content/topics/12-appendices/serverParameters.htm Online: дата 15.05.2022 рік

17. Adaptivecomputing. Queue attributes[Електронний ресурс] – Режим доступу до ресурсу

http://docs.adaptivecomputing.com/torque/4-0-2/Content/topics/4-serverPolicies/queueAttributes.htm#route_destinations Online: дата 15.05.2022 рік