Wykorzystanie klastra Wydziału Informatyki PB do prowadzenia własnych obliczeń

Wojciech Kwedlo
Wydział Informatyki PB
wkwedlo@ii.pb.bialystok.pl

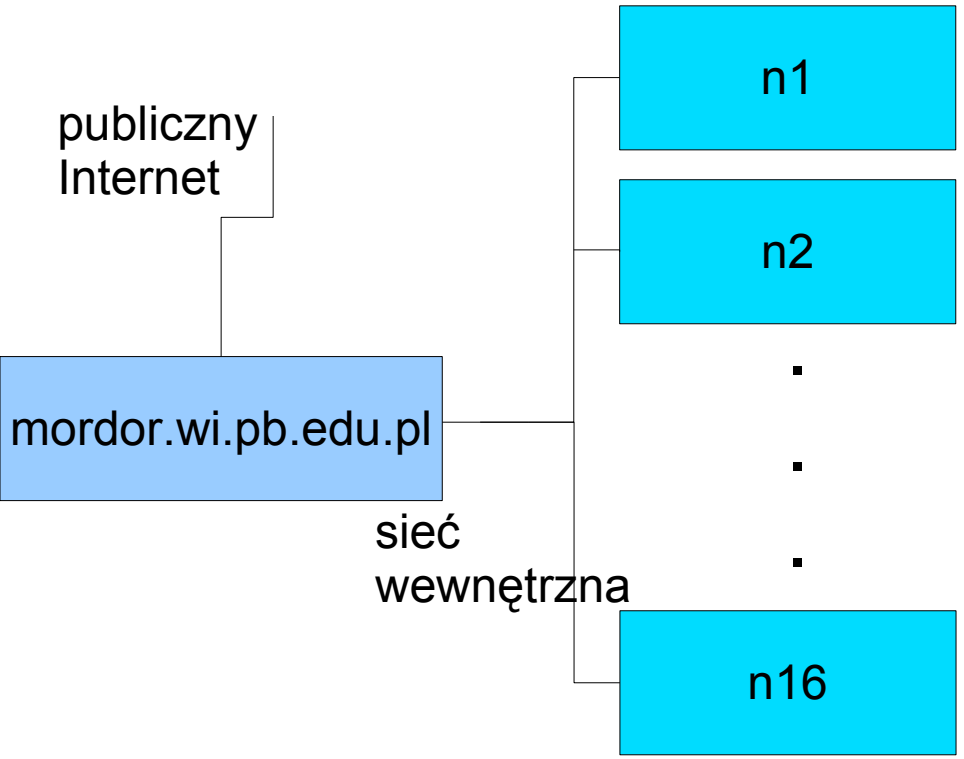
Cele prezentacji

- Zapoznanie potencjalnych użytkowników z architekturą i zasadami funkcjonowania klastra.
- Logowanie się
- Kompilacja programów
- Zlecanie zadań, i monitoring zadań w kolejce
- Zrównoleglanie aplikacji ***nie wchodzi*** w zakres tej prezentacji

Wymagania

- Znajomość podstawowych poleceń Uniksa.
- Posiadanie konta na klastrze (kontakt: wkwedlo@ii.pb.bialystok.pl
- Dostęp do internetu i programów wykorzystujących protokół ssh (Putty i Winscp pod Windows)
- Własna aplikacja, kompilowana z kodu źródłowego C, C++ albo Fortranu.
- Aplikacja o charakterze wsadowym (bez interfejsu graficznego)
  - Z interfejsem teoretycznie możliwe, ale bardzo obciąża sieć

Architektura klastra obliczeniowego Mordor



- Węzeł zarządzający (mordor)
  - 2xXeon 2.800 MHz, 2 GB RAM
  - macierz RAID 1
  - sieciowy system plików (NFS)
  - system kolejkowy
  - monitoring klastra (Ganglia)
  - ***logowanie użytkowników***
- Węzły obliczeniowe: (n1,n2, ..., n16)
  - 2xXeon 2.8GHz, 2 GB RAM
  - 64-bitowy Linux
  - ***nie można się na nie logować***
- Sieć wewnętrzna
  - Gigabit Ethernet (protokół TCP/IP)
  - Infiniband 4x (10 Gb/s, tylko dla aplikacji równoległych MPI)

Klaster - jednolite środowisko obliczeniowe

- Podręcznikowa definicja mówi o klastrze jako "Zbiorze komputerów, sprawiających dla użytkownika wrażenie jednej, silniejszej maszyny"
- Jest to osiągalne poprzez:
  - Identyczny operacyjny (Centos 4 oparty na jądrze 2.6).
  - Identyczny katalog domowy dla każdego z użytkowników na każdej z maszyn. Np użytkownik wkwedlo ma dostęp do identycznej zawartości katalogu /home/wkwedlo, niezależnie od tego czy jest zalogowany na jednym z węzłów n1-n16 czy na węźle zarządzającym
- Możliwa jest zatem sytuacja, że prowadzimy obliczenia na jednym z węzłów obliczeniowych ***i nie ma znaczenia na którym***.
- Problem przydziału węzłów i procesorów użytkownikom.

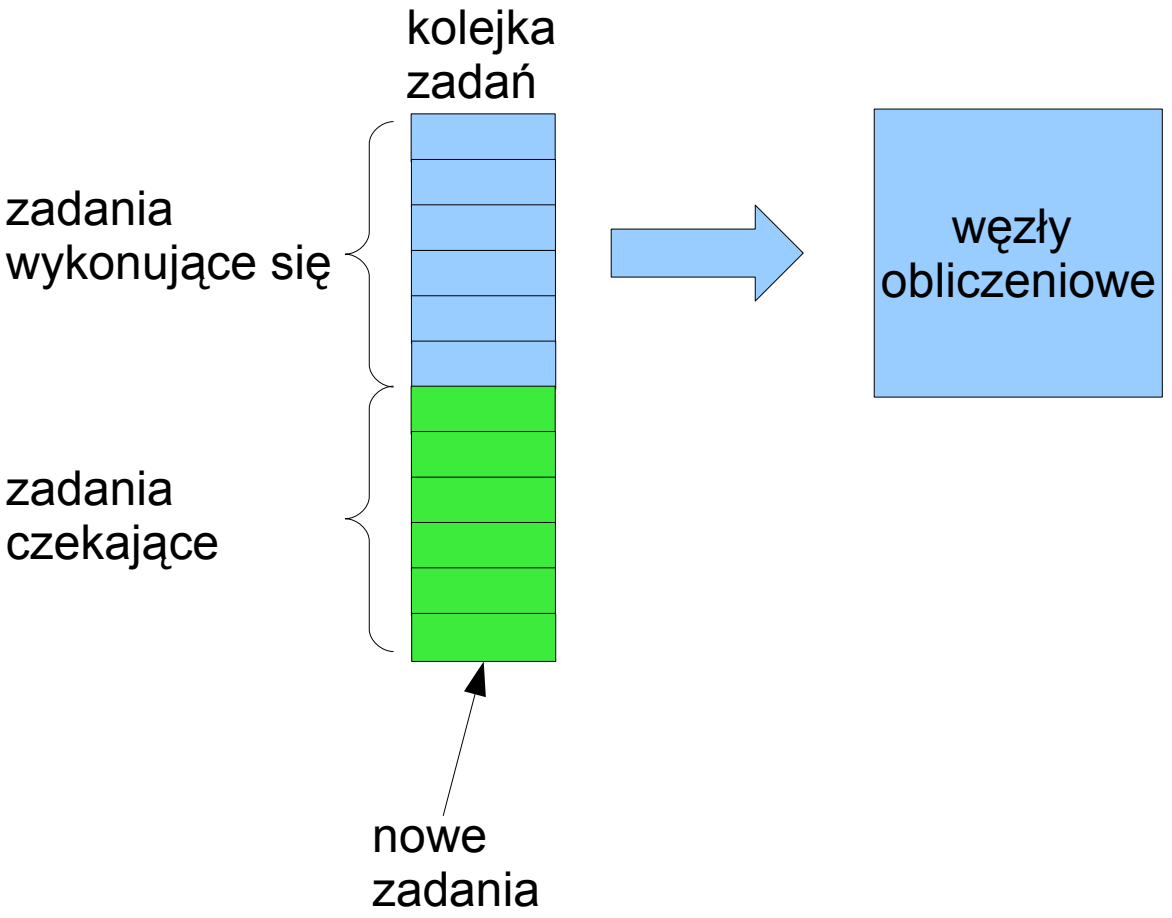
System kolejkowy PBS

- Bezpośrednie logowanie się użytkowników na węzły obliczeniowe prowadziłoby do problemów. Np. kilku użytkowników uruchomiłoby zadania na węźle n1, a węzeł n2 byłby pusty.
- Zamiast tego użytkownik zleca zadanie do kolejki. Jeżeli jakieś procesory są wolne planista klastra może uruchomić zadanie. Jeżeli nie, zadanie czeka w kolejce, aż zwolnią się potrzebne zasoby.
  - Zlecając zadania musimy wyspecyfikować potrzebne zasoby (np. liczbę procesorów).
  - Istnieje cała gama algorytmów planowania (szeregowania) od prostych (FIFO) do bardzo wyrafinowanych.
- Zlecanie do kolejki:

```
qsub skrypt
```

- Skrypt jest "wzbogaconym" skryptem Unixa.
  - definiowane są dodatkowe zmienne rozpoczynające się od PBS_
  - możemy zadawać dodatkowe parametry poprzez specjalny komentarz.

Architektura PBS



- Planista (maui) Wybór zadań z kolejki i przydzielanie im węzłów
  - Maksymalnie dwa zadania szeregowe w jednym węźle

Sprawdzenie stanu kolejki - polecenia qstat

```
wkwedlo@mordor ~/pbs $ qstat -n

mordor:
                                                                  Req'd  Req'd   Elap
Job ID                  Username Queue    Jobname    SessID NDS   TSK Memory Time  S Time
----------------------- -------- -------- ---------- ------ ----- --- ------ ----- - -----
23936.mordor            mkret    test     mc.sh        2255     1  --    --  168:0 R 25:46
   n15
23977.mordor            mkret    test     MRI_Liver.   2305     1  --    --  72:00 R 25:46
   n16
23988.mordor            mkret    test     OB_big.sh   11937     1  --    --  168:0 R 21:33
   n12
23989.mordor            mkret    test     MRI_Liver.  24628     1  --    --  72:00 R 21:32
   n11
23990.mordor            mkret    test     AP_big.sh   25024     1  --    --  168:0 R 21:26
   n11
23991.mordor            mkret    test     OB.sh       12904     1  --    --  168:0 R 21:22
   n10
23992.mordor            mkret    test     AP.sh       13128     1  --    --  168:0 R 21:20
   n10
```

stan zadania

przydzielony węzeł

- opcja -n polecenia qstat nakazuje pokazanie przydzielonych węzłów
- stany zadania
  - R - *running* (wykonujące się)
  - Q - *queued* (czekające w kolejce na wolne procesory)
  - E - *exiting* (kończące prace)

Przykładowy skrypt PBS - plik test

```
#!/bin/sh
#PBS-l nodes=1:ppn=1          jeden węzeł i jeden procesor
hostname                     wypisz nazwę hosta
pwd                          wypisz katalog
cd $PBS_O_WORKDIR            przejdź do katalogu z którego
                              wywołano qsub

pwd
echo Rozpoczynam obliczenia
sleep 30s                    czekaj 30s
echo Koncze obliczenia
```

w tej linii wywołanie programu wykonującego (wraz z wierszem poleceń) nasze skomplikowane obliczenia

```
wkwedlo@mordor ~/pbs $ qsub test

23995.mordor
```

- Zadanie trafiło do kolejki i uzyskało numer 23995
- Numer jest używany do identyfikacji zadania.
- W rzeczywistym zadaniu zamiast sleep 30, wstawić wywołania programu wykonującego obliczenia.

Anatomia skryptu PBS

- Dyrektywy PBS możemy zadawać na dwa sposoby:
- Jako specjalnie sformatowane komentarze w pliku:

```
#PBS -l nodes=1:ppn=1
```

- albo alternatywnie jako opcje polecenia `qsub`

```
qsub -l nodes=1:ppn=1 test.sh
```

Obydwa sposoby są równoważne. Niektóre inne dyrektywy PBS to:

`-o stdout`	Zapisz standardowe wyjście do pliku stdout
`-e stderr`	Standardowe wyjście diagnostyczne

Dalsze szczegóły: `man qsub`

Sprawdzenie stanu kolejki - polecenia qstat

```
 qstat -n

mordor:

                                                                  Req'd  Req'd   Elap
Job ID                  Username Queue    Jobname    SessID NDS   TSK Memory Time  S Time
----------------------- -------- -------- ---------- ------ ----- --- ------ ----- - -----
23936.mordor            mkret    test     mc.sh        2255     1  --    --  168:0 R 25:54
   n15
23977.mordor            mkret    test     MRI_Liver.   2305     1  --    --  72:00 R 25:53
   n16
23988.mordor            mkret    test     OB_big.sh   11937     1  --    --  168:0 R 21:40
   n12
23989.mordor            mkret    test     MRI_Liver.  24628     1  --    --  72:00 R 21:39
   n11
23990.mordor            mkret    test     AP_big.sh   25024     1  --    --  168:0 R 21:33
   n11
23991.mordor            mkret    test     OB.sh       12904     1  --    --  168:0 R 21:29
   n10
23992.mordor            mkret    test     AP.sh       13128     1  --    --  168:0 R 21:27
   n10
23995.mordor            wkwedlo  test     test        25491     1  --    --  12:00 R   --
   n16
```

nowe zadanie

domyślnie 12g.
na zadanie

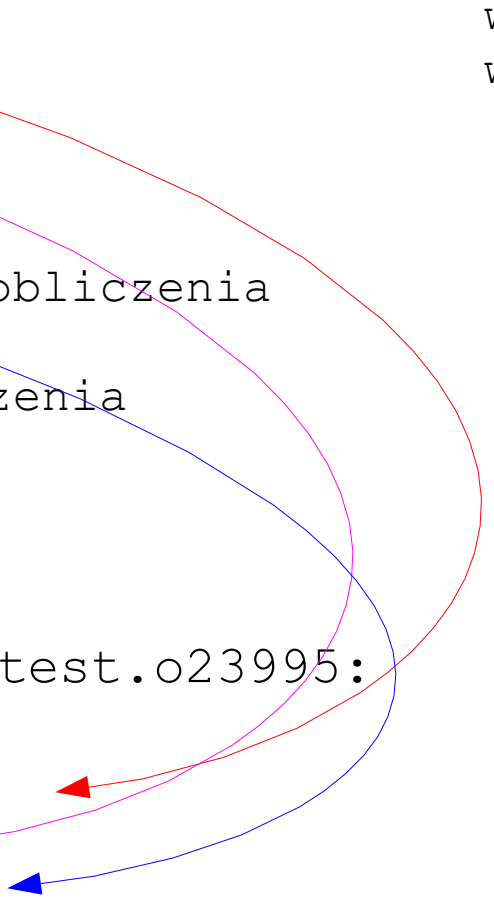
Po 30 sekundach

```
wkwedlo@mordor ~/pbs $ ls -l
total 8
-rwxr--r--  1 wkwedlo wkwedlo 127 Mar 13 12:45 test
-rw-------  1 wkwedlo wkwedlo   0 Mar 13 12:58 test.e23995
-rw-------  1 wkwedlo wkwedlo  90 Mar 13 12:58 test.o23995
```

- Plik test.o23995 standardowe wyjście zadania 23995
- Plik test.e23995 standardowe wyjście diagnostyczne (stderr) zadania 23995
- Nazwy plików można zmienić

Standardowe wyjście skryptu

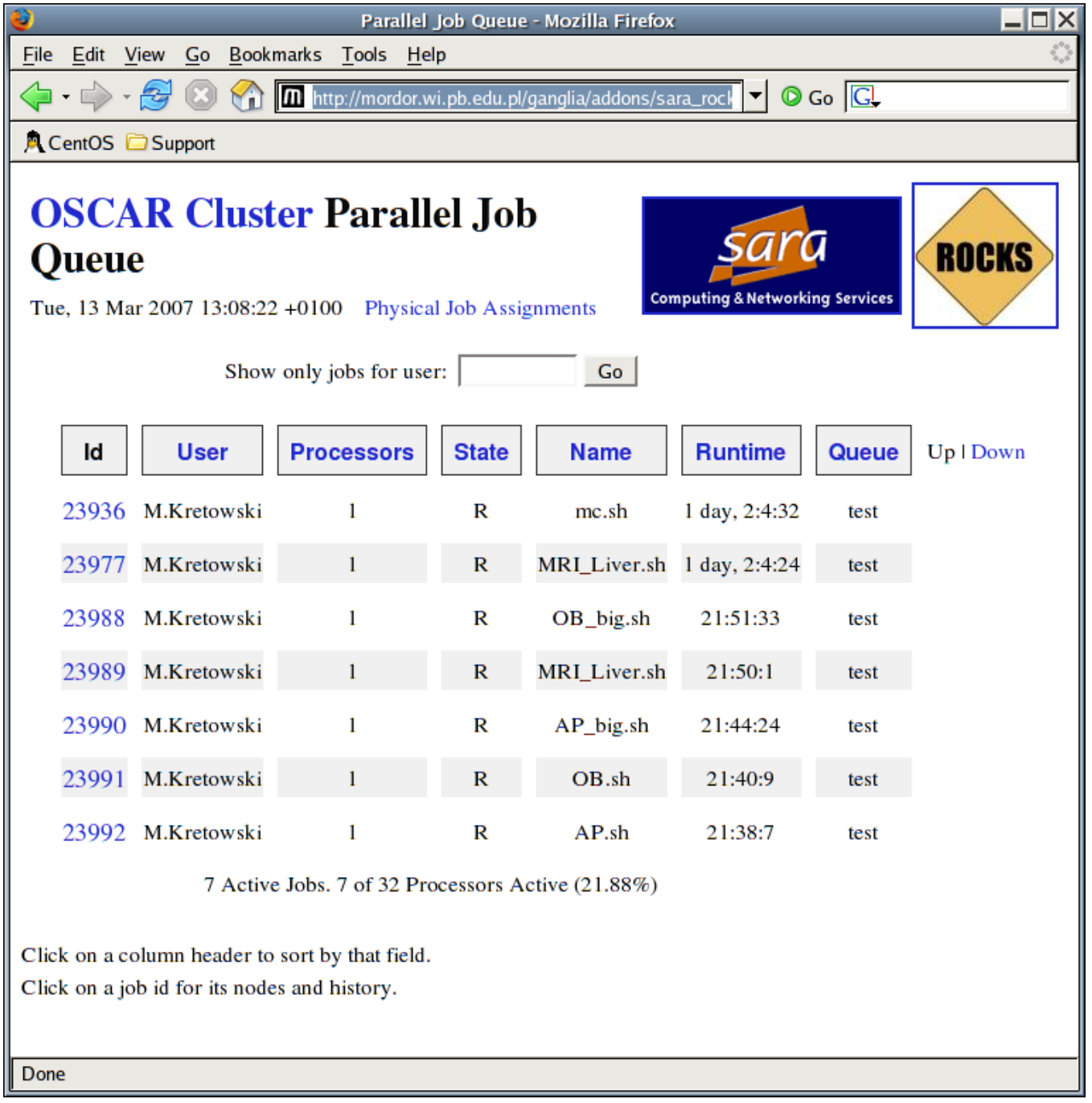
```
#!/bin/sh
#PBS-l nodes=1:ppn=1          jeden węzeł i jeden procesor
hostname                     wypisz nazwę hosta
pwd                          wypisz katalog
cd $PBS_O_WORKDIR             przejdź do katalogu z którego
                              wywołano qsub

pwd
echo Rozpoczynam obliczenia
sleep 30s                     czekaj 30s
echo Koncze obliczenia
```



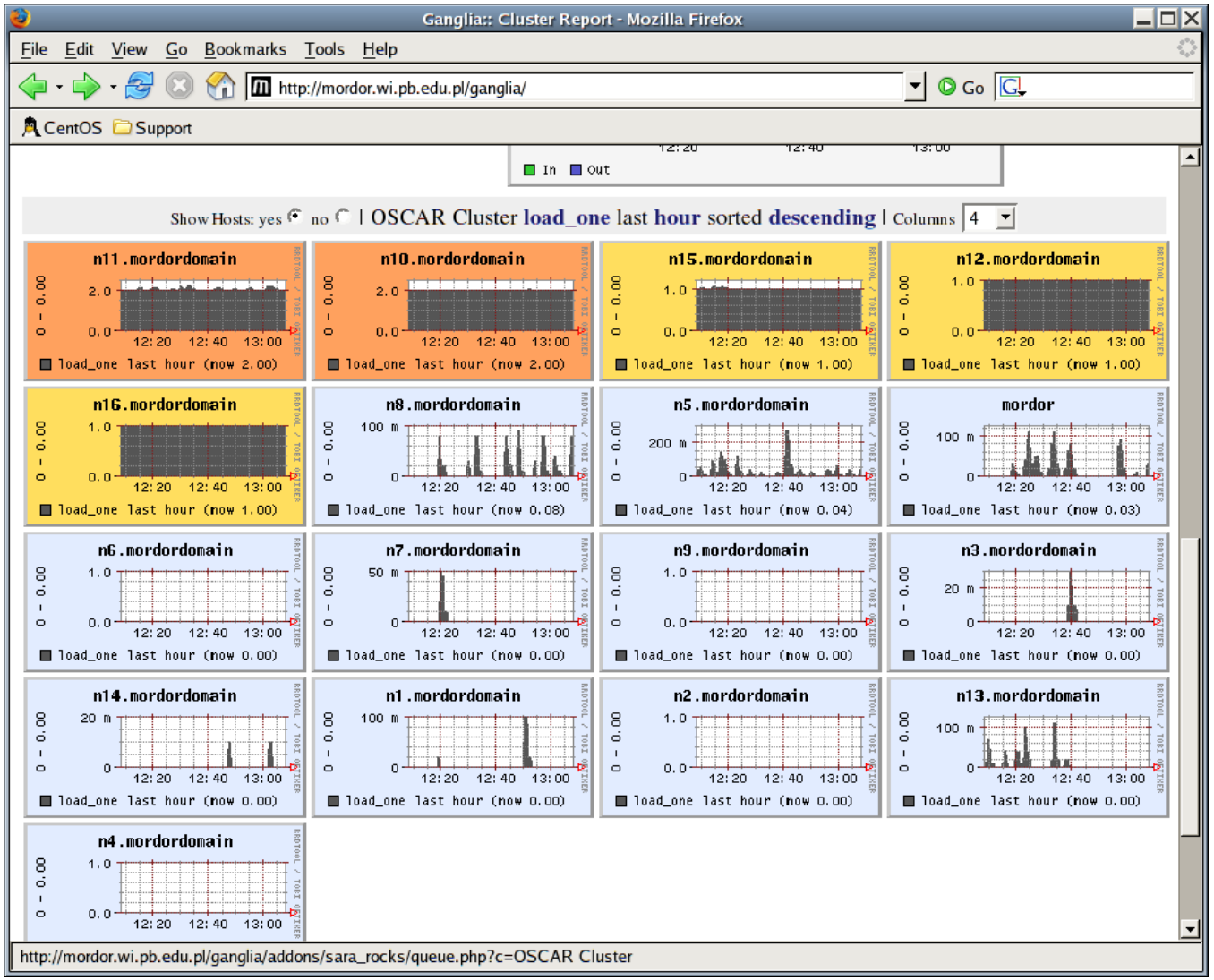
Zawartość pliku test.o23995:

```
n16.mordordomain
/home/wkwedlo
/home/wkwedlo/pbs
Rozpoczynam obliczenia
Koncze Obliczenia
```

Monitoring kolejki przez stronę www



Monitoring klastra przez www



- Różne metryki
  - obciążenie (procesora, dysku, sieci)
  - wolna pamięć
  - temepratura
  - wiele innych

Zlecamy wykonanie skryptu test 20 razy

```
wkwedlo@mordor ~/pbs $ qstat
Job id                    Name             User            Time Use S Queue
------------------------- ---------------- --------------- -------- - -----
23936.mordor              mc.sh            mkret           25:59:28 R test
23977.mordor              MRI_Liver.sh     mkret           25:58:38 R test
23988.mordor              OB_big.sh        mkret           21:46:56 R test
23989.mordor              MRI_Liver.sh     mkret           21:44:10 R test
23990.mordor              AP_big.sh        mkret           21:38:35 R test
23991.mordor              OB.sh            mkret           21:34:05 R test
23992.mordor              AP.sh            mkret           21:32:00 R test
23996.mordor              test             wkwedlo                0 R test
23997.mordor              test             wkwedlo                0 R test
23998.mordor              test             wkwedlo                0 R test
23999.mordor              test             wkwedlo                0 R test
24000.mordor              test             wkwedlo                0 R test
24001.mordor              test             wkwedlo                0 R test
24002.mordor              test             wkwedlo                0 R test
24003.mordor              test             wkwedlo                0 R test
24004.mordor              test             wkwedlo                0 R test
24005.mordor              test             wkwedlo                0 R test
24006.mordor              test             wkwedlo                0 R test
24007.mordor              test             wkwedlo                0 R test
24008.mordor              test             wkwedlo                0 R test
24009.mordor              test             wkwedlo                0 R test
24010.mordor              test             wkwedlo                0 R test
24011.mordor              test             wkwedlo                0 R test
24012.mordor              test             wkwedlo                0 Q test
24013.mordor              test             wkwedlo                0 Q test
24014.mordor              test             wkwedlo                0 Q test
24015.mordor              test             wkwedlo                0 Q test
```

Aktualnie obowiązujące ograniczenia

- Zadanie nie może mieć dłuższego czasu pracy (walltime) niż 7 dni (168 godzin). Jeżeli nie zadamy czasu pracy, to domyślnie przyjmowane jest 12 godzin. Po upłynięciu zadanego czasu pracy zadanie jest usuwane.

  ustawienie czasu na 30 godzin.

  ```
  #PBS -l walltime=30:00:00
  ```

- W danej chwili nie może wykonywać się więcej niż 16 zadań jednego użytkownika.
  - Jeden użytkownik nie zmonopolizuje całego klastra (32 procesory=32 zadania jednoprocesorowe)

Dalsze polecenia systemu PBS

`pbsnodes -a`

- Sprawdza stan węzłów

`qdel jobif`

- Usuwa z kolejki zadanie

`qsig -s signal jobid`

- Wysyła sygnał do zadania

`qalter jobid`

- zmiana parametrów zadania (po jego zgłoszeniu)

Scenariusz pracy

- Zalogować się na węzeł dostępowy: ssh na Uniksach, program putty pod Windows (host mordor.wi.pb.edu.pl). Poleceniami sftp lub programem WinSCP można przegrać niezbędne pliki (kod programu dane).
- Zalogować się na węzeł n1 i tam skompilować program.
- Stworzyć n skryptów PBS (np. w katalogu w podkatalogu domowym)
- Zlecić n razy zadania do kolejki (polecenie qsub). Po zleceniu zadania plik skryptu nie jest potrzebny i można go modyfikować.
- Można się zupełnie wylogować z klastra.
- Monitorować stan zadań: a) co jakiś czas logować się na klaster i poleceniem qstat sprawdzać, czy zadania się wykonały. b) przy pomocy strony www

Kompilatory Intela

- Zastosowanie kompilatorów Intela daje szybszy kod w porównaniu do darmowych kompilatorów GNU.

- Kompilatory zainstalowane są na węźle zarządzającym.
  - Jeżeli dwóch użytkowników zechce naraz kompilować program, zabraknie licencji.

- Kompilator C++ wywołujemy poleceniam `icpc` a fortranu 90 `ifort`.

- Polecane przeze mnie opcje kompilacji to: `-O3 -ipo`

- Obydwa kompilatory wspierają standard OpenMP (programowania wielowątkowego, dla maszyn SMP ze wspólną pamięcią)

Korzyści z klastra

- Praca na klastrze nie obciąża komputera na biurku.

- Nawet jeżeli nasz program nie jest zrównoleglony, możemy osiągnąć skrócenie czasu obliczeń, jeżeli wykonujemy wiele niezależnych eksperymentów.
  - Obliczenia dla poszczególnych eksperymentów (np. wywołania tego samego programu z różnymi parametrami i dla różnych danych) mogą być wykonywane jednocześnie na różnych procesorach klastra.

- Klaster pracuje 24 godziny na dobę 7 dni w tygodniu.

- Zlecamy zadania do kolejki i możemy się wylogować. Zadania we właściwym czasie zostaną wykonane.

Dalsze informacje

- mordor.wi.pb.edu.pl – strona domowa klastra, w tym monitoring kolejki i klastra.
- wkwedlo@ii.pb.białystok.pl – prośby o założenie konta
- klaster-users@kwi.pb.edu.pl - lista dyskusyjna użytkowników klastra
- http://www.clusterresources.com/pages/products/torque-resource-manager.php opis torque (wersji systemu PBS na klastrze)
- http://www.clusterresources.com/pages/products/maui-cluster-scheduler.php opis planisty MAUI

Przyszłość

- Nowy klaster (dotacja z Ministerstwa Nauki i Szkolnictwa Wyższego)
  - Infiniband DDR (20Gb/s)
  - 16 serwerów, każdy z dwoma procesorami dwurdzeniowymi
  - 2-3 krotnie większa moc obliczeniowa