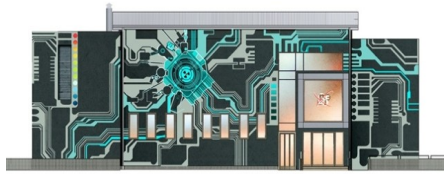
Obliczenia w CIŚ



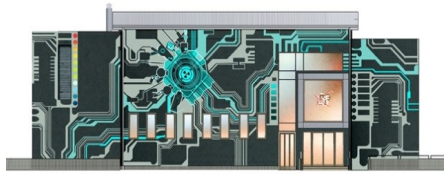
Jak korzystać z CIŚ i co można tam zrobić

Aplikacje

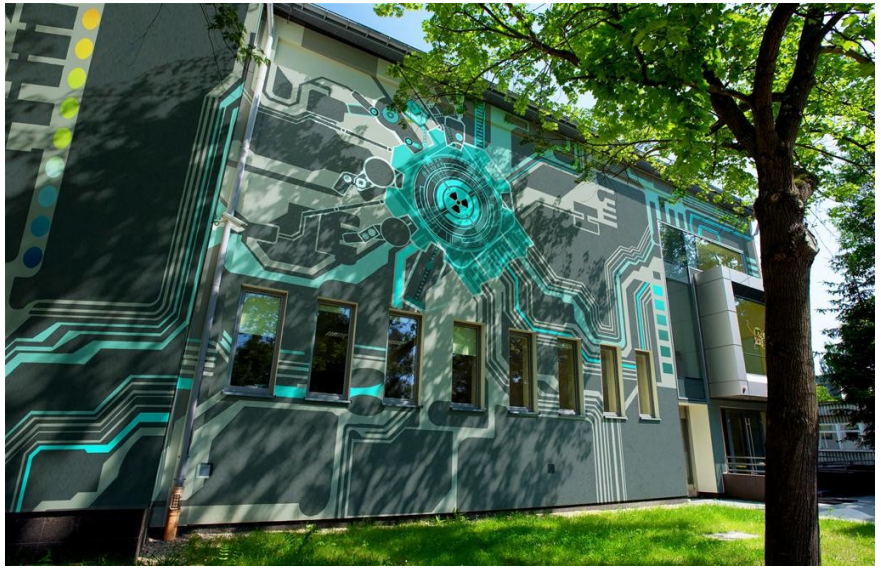
Krzysztof Nawrocki

16 grudnia 2014

Z czego składa się CIŚ

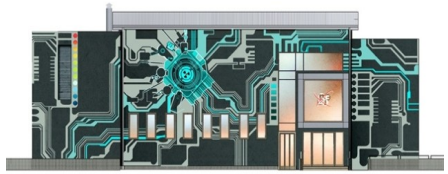


- Grupy i tematy naukowe
  - Zespół Analiz Reaktorowych (ZAR)
  - Centrum Doskonałości MANHAZ
  - Complex Systems Team
  - Obliczenia HEP i Astrofizycznych
  - Projektowanie i symulacje PET
  - Inne (dużo innych)



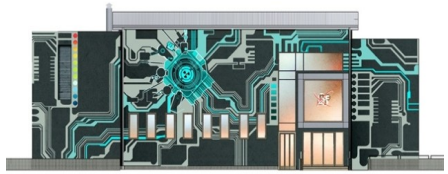
- Granty obliczeniowe
  - https://www.cis.gov.pl/pl/granty-obliczeniowe

Z czego składa się CIŚ (cd)



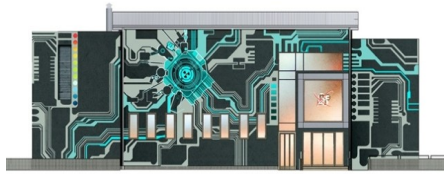
- Zasoby obliczeniowe
  - 285 TFLOPS
  - 13760 rdzeni obliczeniowych
  - 3PB przestrzeni dyskowej
  - 83456 GB RAM
- Klastry
  - HPC
  - GRID
  - Big Data

Źródła informacji o klastrze



- www.cis.gov.pl
- wiki: doc.cis.gov.pl/cis
  - https://doc.cis.gov.pl/cis/index.php/Oprogramowanie_naukowe
- seminaria CIŚ
- spotkania deweloperów CIŚ
  - https://doc.cis.gov.pl/cis/index.php/Rozwoj_oprogramowania

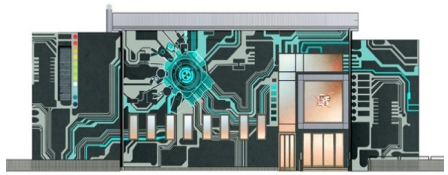
Dostępne oprogramowanie



- Oprogramowanie naukowe
  - np.: Ansys Fluent, OpenFOAM, Mathematica, MATLAB, Simulink, MDCS, CPLEX, Maxima, R, Geant4, GATE, CORSICA, Fluka, Flair, PFLOTRAN, MCNP, Trio_U, SCALE, GAMESS, ROOT, ...
- Oprogramowanie narzędziowe
  - np.: gcc, open64, Intel Cluster Studio, PGI, blas, lapack, boost, openmpi, mpich2, ...
- więcej: moduły środowiska
  - https://doc.cis.gov.pl/cis/index.php/Moduły_środowiska
  - np: module avail, module whatis, module help, module list

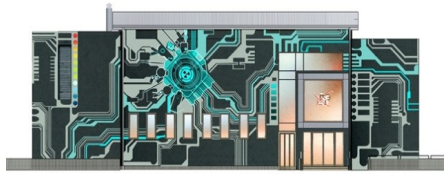
    module load <nazwa>, module unload <nazwa>

Narzędzia programistyczno - administracyjne

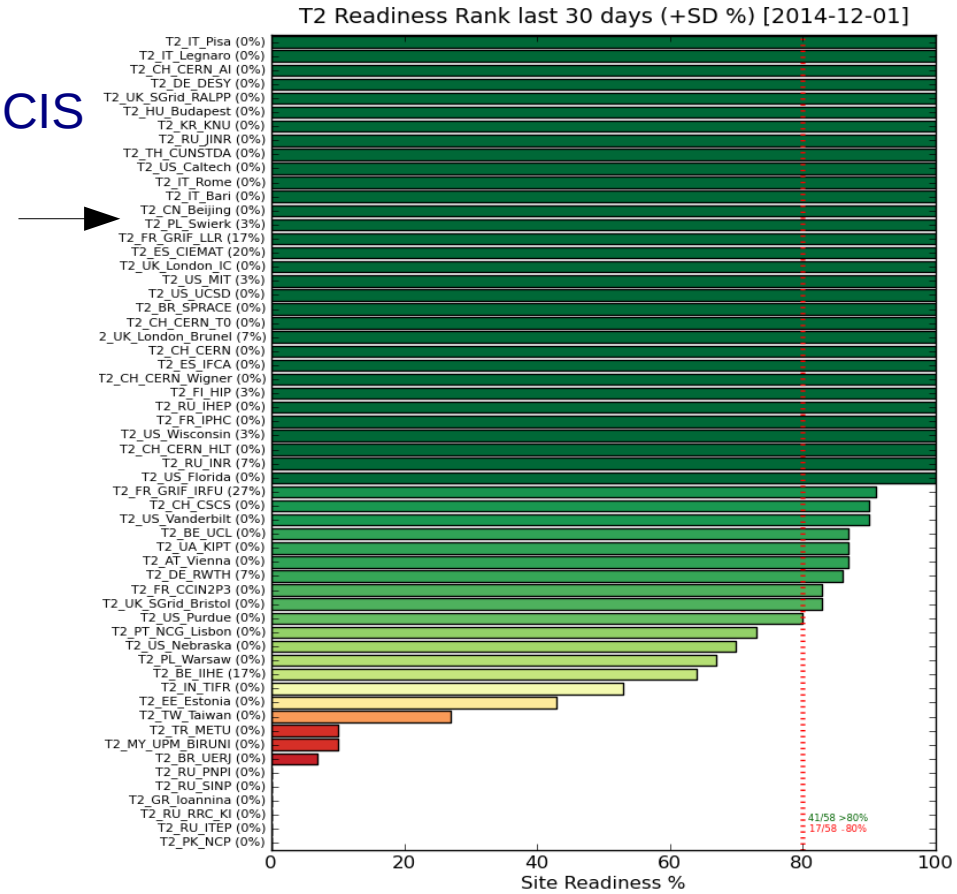


- Wiki
- Wsparcie: helpdesk.cis.gov.pl
- Repozytorium kodu: GIT / Gitolite
  - https://doc.cis.gov.pl/cis/index.php/Repozytorium_kodu_dla_uzytkownikow_klastra
- Monitoring użycia licencji
- Wewnętrzny „Dropbox“: seafile.cis.gov.pl

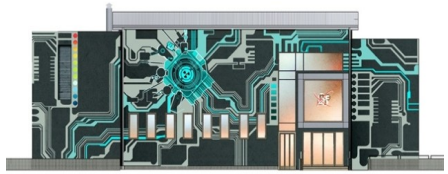
Obliczenia na Gridzie



- klaster NCBJ-CIS
  - http://gstat2.grid.sinica.edu.tw/gstat/site/NCBJ-CIS
- zasoby
  - CPU: 1100 rdzeni obliczeniowych
  - przestrzeń dyskowa: ok. 600 TB
  - łącze do świata: ~ 5 Gbps
- wspierane organizacje wirtualne (VO)
  - **LHCb** (T2D)
  - **CMS** (T2_PL_Swierk)
- maszyna dostępowa
  - gridui.cis.gov.pl

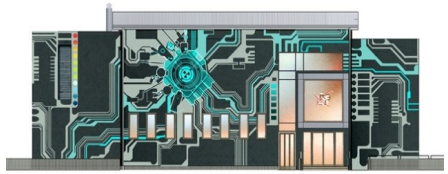


Klaster Big Data



- Big Data to analiza danych o wielkiej
  - objętości, różnorodności, zmienności
- Dedykowane narzędzia
  - **Apache Hadoop** – MapReduce (+ Pig, HBase, Solr, Mahout...)
  - **Apache Spark** – lazy evaluation (+ MLLib , GraphX, Spark Streaming)
- Instalacja testowa w CIŚ
  - ok. 200 rdzeni obliczeniowych, 10 TB przestrzeni dyskowej
  - oparta na dystrybucji CDH5 (Cloudera)

Centrum Informatyczne w Świerku



www.cis.gov.pl

Zapraszamy

JAK UZYSKAĆ DOSTĘP DO KLASTRA CIŚ

MICHAŁ WÓJCIK
NETWORK SECURITY TEAM

Charakterystyka zespołu

- Tworzenie i administracja polityką bezpieczeństwa
- Wewnętrzne testy penetracyjne
  - Dwóch członków zespołu z certyfikatami CEH
- Analizy bezpieczeństwa
- Wewnętrzne audyty bezpieczeństwa
- Wdrażanie mechanizmów bezpieczeństwa
- Administracja systemem IDS/IPS , SIEM
- Monitoring infrastruktury
- Administracja siecią
- Reagowanie na incydenty bezpieczeństwa
- Administracja platformą wirtualizacji

Kto może otrzymać dostęp do klastra CIŚ

- Pracownik Narodowego Centrum Badań Jądrowych
- Pracownik spoza NCBJ
  - *Po uzyskaniu grantu obliczeniowego, bądź dołączeniu do zespołu już realizującego obliczenia.*

Mechanizm dostępu do klastra

Dostęp można uzyskać na dwa sposoby:

- VPN
  - Konieczna dokładna weryfikacja tożsamości użytkownika (certyfikaty)
  - Wykorzystuje PKI oraz kryptografię asymetryczną
  - CIŚ CA wystawia oraz unieważnia certyfikaty
  - Pełna funkcjonalność (dostęp graficzny – zdalny pulpit lub konsola), wszystkie dostępne licencje
- SSH bez VPN
  - bezpośredni dostęp do serwera dostępowego o mocno ograniczonej funkcjonalności – jedynie terminal, brak licencji na niektóre programy

ui.cis.gov.pl port 22222

Procedura uzyskania dostępu do klastra CIŚ

- Zapoznanie się z regulaminem dostępnym na stronie http://www.cis.gov.pl/korzystanie-z-klastra, wydrukowanie go oraz podpisanie.
- Złożenie wniosku o certyfikat za pomocą aplikacji na stronie https://ca.cis.gov.pl/keys.php
- Spotkanie z administratorem CA.
  - W Świerku w budynku 88 pok. 103
  - W Warszawie na ul. Hożej 69
- Na spotkanie należy zabrać ze sobą:
  - dowód tożsamości
  - podpisany regulamin
  - przenośną pamięć flash

Na spotkaniu..

W trakcie spotkania:

- Weryfikowana jest tożsamość użytkownika.
- Zakładane jest konto do maszyn dostępowych klastra.
- Zostaje podpisany certyfikat użytkownika.
- Użytkownik dostaje instrukcję dostępu do klastra.

Gdy chcemy się połączyć...

Za pomocą tunelu VPN:

Niezbędne są wcześniej otrzymane trzy pliki:

- inazwisko.key – plik z kluczem prywatnym

  WAŻNE! Plik ten jest poufny. Nie powinien być nikomu przekazywany (nawet administratorowi).

- inazwisko.crt – plik z certyfikatem użytkownika

- ca.crt – publiczny certyfikat Centrum Autoryzacji

Konfiguracja połączenia VPN

- System Windows
  - Wymagany jest klient OpenVPN oraz plik *inazwizko.ovpn*
- Linux - Network Manager
- Mac OS X - program Tunnelblick
- https://ca.cis.gov.pl/vpn_howto.php

Przedłużenie ważności certyfikatu

- Każdy certyfikat użytkownika wydany zostaje z określoną datą ważności.
- Koniec daty ważności sygnalizowany jest automatycznie poprzez wiadomość e-mail.
- Instrukcja jak przedłużyć ważność certyfikatu znajduje się na stronie:

  https://ca.cis.gov.pl/extend.php

- Po wygenerowaniu wniosku o podpisanie nowego certyfikatu (przedłużenie ważności obecnego), należy „ręcznie” przesłać go do administratora CA:

  ca@cis.gov.pl

MICHAŁ WÓJCIK

NETWORK SECURITY TEAM

Zasoby obliczeniowe

Dostęp i uruchamianie zadań

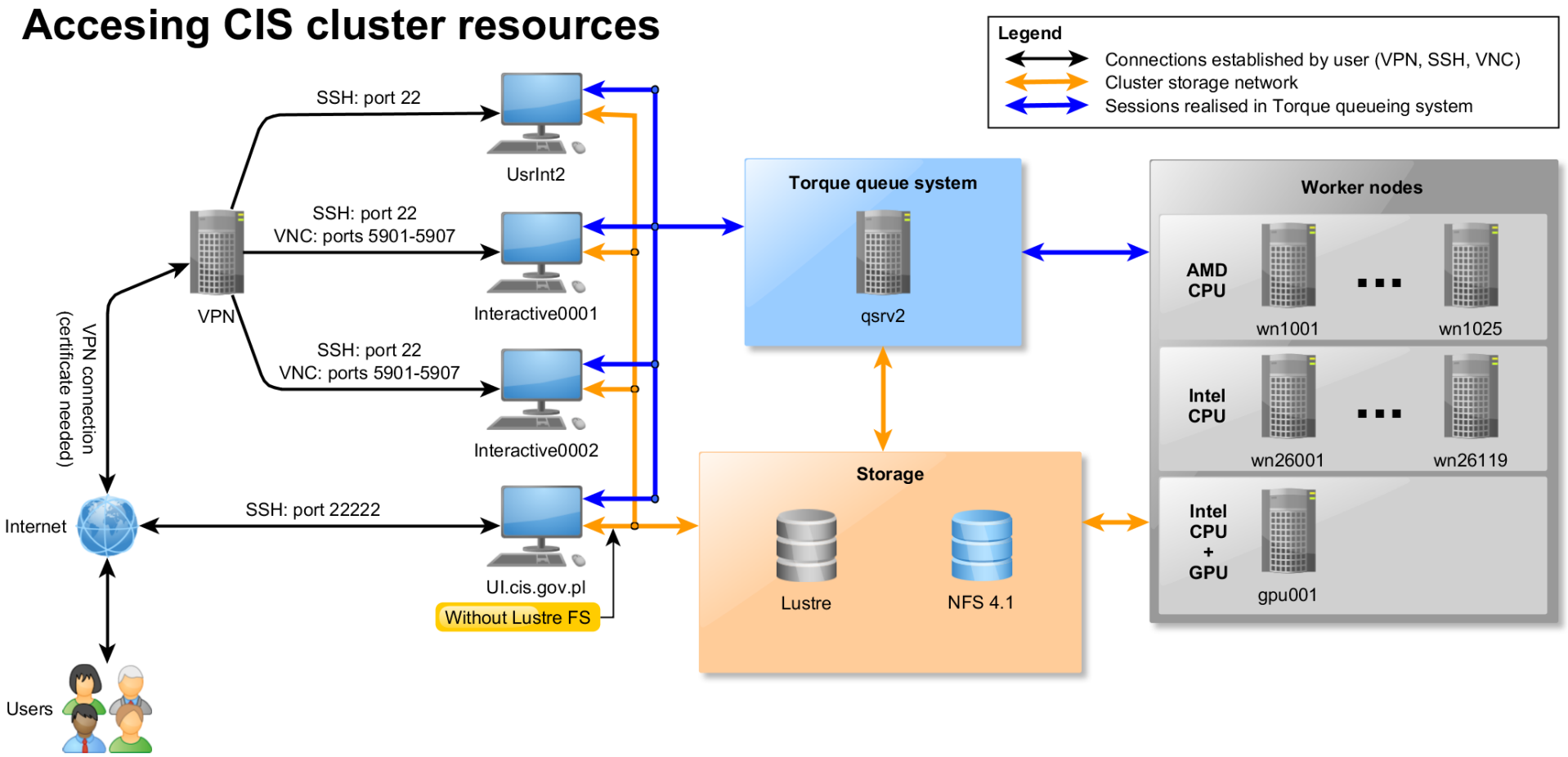
Piotr Wasiuk

Warszawa, 16 grudnia 2014

Spis treści

- Wprowadzenie;
- Struktura systemu klastra;
- Dostępne zasoby obliczeniowe;
- Uzyskiwanie dostępu do klastra obliczeniowego CIŚ;
- Uruchamianie procesów obliczeniowych;
- Dokumentacja oraz uzyskiwanie pomocy;

Struktura systemu klastra CIŚ



Dostępne zasoby obliczeniowe

- Serwery marki HP:
  - 24 serwery wyposażone w 4 procesory marki AMD (model: Opteron(TM) 6276@2.3GHz)
  - 64 rdzenie / serwer;
  - 256GB pamięci RAM;
  - 4 x Ethernet 10Gbit/s + 2 x Infiniband 40Gbit/s;
  - Wydajność serwera: 588.8 GFLOPS;
- Serwery marki SuperMicro:
  - 118 serwerów wyposażonych w 2 procesory marki Intel (model: Intel(R) Xeon(R) CPU E5-2680 v2@2.80GHz)
  - 20 rdzenie / serwer (40 rdzeni z uwzględnieniem HT)
  - 128 GB pamięci RAM;
  - 1 x Ethernet 10Gbit/s + 1 x Infiniband 40Gbit/s;
  - Wydajność serwera: 448 GFLOPS;

Dostępne zasoby obliczeniowe

- Ponadto, każdy z serwerów posiada zamontowane następujące zasoby dyskowe:
  - Lokalny system plików ***(scratch)*** oparty o dyski SSD zapewniający wydajność na poziomie ~500MB/s;
  - Współdzielony system plików ***(home)*** – macierz marki NetApp o wydajności do 1GB/s (interfejs Ethernet 10Gbit/s);
  - Współdzielony system plików ***(Lustre)*** – wysokowydajny system plików na tymczasowe dane obliczeń (ok 2.5GB/s, poprzez interfejs Infiniband 40Gbit/s);

Dostęp do klastra CIŚ - serwery

- Serwery dostępowe:
  - Interfejs dla uruchamiania zadań obliczeniowych;
  - Jedyne maszyny o bezpośrednim dostępie;
- Dostępne protokoły:
  - SSH:
    - port 22 z obsługą przekierowywania systemu X11;
    - port 22222 dla serwera UI.cis.gov.pl dostępnego spoza sieci VPN;
  - VNC:
    - porty 5901-5907 – wybierane w zależności od wymaganego trybu graficznego;

Dostęp do klastra CIŚ – serwery

Zestawienie serwerów dostępowych w systemie klastra CIŚ:

Nazwa serwera	Dostępność	Dostępne protokoły	
		SSH	VNC
UsrInt2.cis.gov.pl	Wewnątrz sieci VPN	Tak; port: 22	Nie
Interactive0001.cis.gov.pl	Wewnątrz sieci VPN	Tak; port: 22	Tak; porty 5901-5907
Interactive0002.cis.gov.pl	Wewnątrz sieci VPN	Tak; port: 22	Tak; porty 5901-5907
UI.cis.gov.pl	Publiczny	Tak; port: 22222	Nie

Dostęp do klastra CIŚ - protokół SSH

- Systemy Linux, UNIX, iOS, Android, etc. – wbudowane polecenie *ssh*:

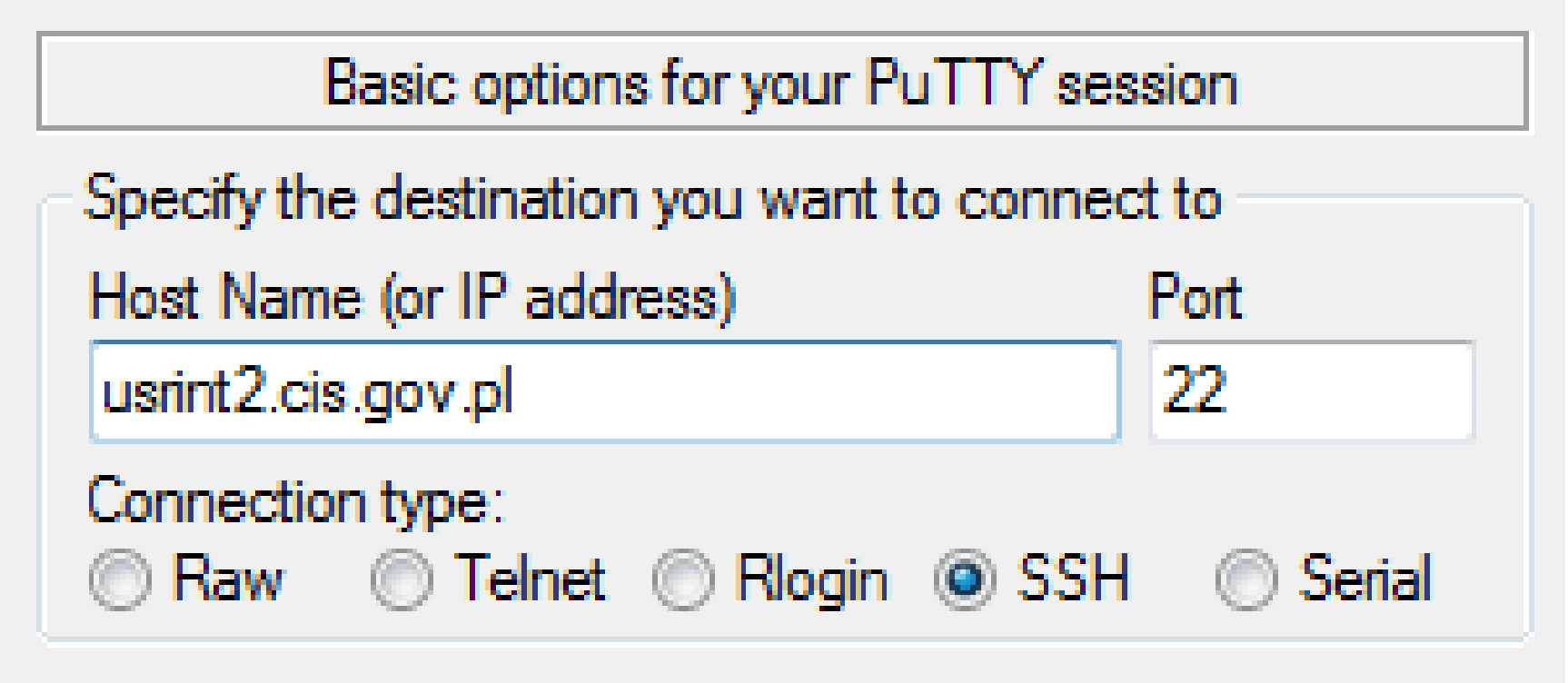
```
[user@host]$ ssh usrint2.cis.gov.pl
```

- Przekierowywanie systemu X11 w systemach Linuksowych:
  - Umożliwia pracę zdalną z oprogramowaniem wyposażonym w graficzny interfejs użytkownika na serwerach obliczeniowych klastra.
  - Obsługiwane przez system kolejkowy;
  - Zestawienie połączenia:

```
[user@host]$ ssh usrint2.cis.gov.pl -Y
```

Dostęp do klastra CIŚ - protokół SSH

- Protokół SSH – konfiguracja klienta dla systemu Windows (PuTTY).

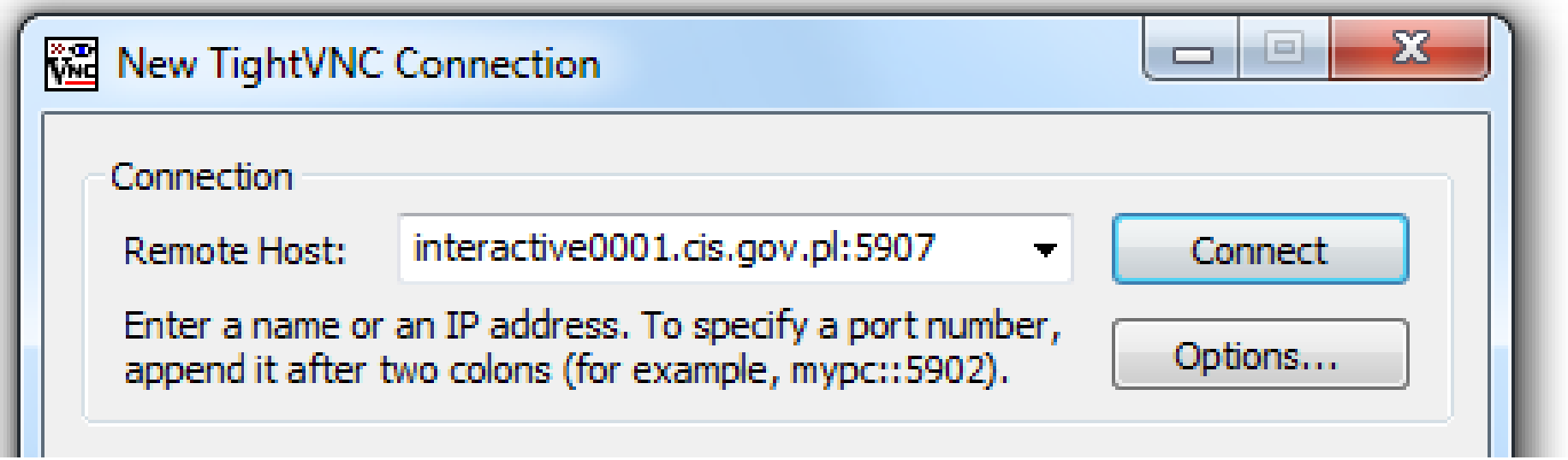


Dostęp do klastra CIŚ – protokół VNC

- Protokół VNC:
  - Umożliwia bezpośrednie połączenie z pulpitem serwera dostępowego;
  - Tryb pracy (rozdzielczość oraz głębia kolorów) określany na podstawie wybranego portu;
  - Udostępniane środowisko: pulpit GNOME w wersji 2.x.
  - Zastosowanie: aplikacje silnie nastawione na wykorzystanie GUI (np. typu CAD, symulatory), tworzenie dokumentów, testy, etc;

Dostęp do klastra CIŚ – protokół VNC

- Przykładowa konfiguracja klienta dla systemu Windows (TightVNC):



- Po poprawnym podłączeniu logowanie przy pomocy loginu i hasła Użytkownika;

System kolejkowy

- Idea stosowania systemu kolejkowego:
  - Wspólny dla wszystkich Użytkowników system nadzoru nad zadaniami obliczeniowymi;
  - Współpraca z wieloma aplikacjami oraz możliwość automatyzacji procesów obliczeniowych;
  - Możliwość definiowania wymaganych zasobów oraz stosowania polityk fairshare oferujących równy podział dostępnych zasobów dla wszystkich Użytkowników;
- Struktura logiczna systemu:
  - System kolejkowy -> Kolejki -> Zadania obliczeniowe;

System kolejkowy - kolejki

- Kolejki:
  - Wybór kolejki określa docelową architekturę oraz maksymalny dopuszczalny czas dla wykonania zadania obliczeniowego;
  - Polityka „fairshare”: priorytetyzowanie zadań na podstawie historii wykorzystania klastra, wymagań na zasoby obliczeniowe oraz wielkości zadania;

System kolejkowy - kolejki

- Zestawienie zdefiniowanych kolejek w systemie:

Procesory docelowe	Nazwa kolejki	Maksymalny czas wykonania - Walltime [h]
AMD 64 rdzenie/węzeł 256GB RAM/węzeł	a12h	12:00:00
	a3d	72:00:00
	a7d	168:00:00
	a14d	336:00:00
Intel 40 rdzeni/węzeł 128GB RAM/węzeł	i12h	12:00:00
	i3d	72:00:00
	i7d	168:00:00
	i14d	336:00:00
Intel+GPU	gpu	48:00:00

System kolejkowy – zadania obliczeniowe

- Rodzaje zadań obliczeniowych:
  - Wsadowe - skrypt uruchamiający procesy obliczeniowe przekazywany jest do systemu kolejkowego. Po zakończonych obliczeniach wyniki zwracane są Użytkownikowi w postaci plików z wyjściem STDOUT oraz STDERR (wykorzystanie: produkcyjne);
  - Interaktywne - Użytkownik uzyskuje bezpośredni dostęp do konsoli tekstowej węzła obliczeniowego (wykorzystanie: testy, edycja dokumentów, kompilacja oprogramowania, etc.)

System kolejkowy – uruchamianie zadań

- *qsub* – polecenie uruchamiające zadanie. Pomyślnie wykonanie zwraca ID nowo utworzonego zadania;
- Często wykorzystywane przełączniki:
  - „-q” : określa kolejkę (architektura CPU oraz czas walltime);
  - „-l” : pozwala na określenie ilości wymaganych zasobów (rdzenie CPU, węzły obliczeniowe, pamięć RAM, etc.);
  - „-I” : uruchamia zadanie w trybie interaktywnym;

System kolejkowy – polecenia Torque

- *qstat* – listuje wszystkie aktualnie uruchomione oraz zakolejkowanie w systemie zadania;
- *qdel <ID>* – usuwa podane zadanie;
- *qnodes* – szczegółowa lista węzłów obliczeniowych wraz z ich aktualnym stanem;
- *qstatx, qnodesx* – nakładki na ww. polecenia ułatwiające pracę z systemem kolejkowym;

System kolejkowy - przykłady

- Uruchomienie zadania interaktywnego:
  - `qsub –I`
- Deklaracja wymaganych zasobów:
  - `qsub –l nodes=1:ppn=20 –I`
- Deklaracja zasobów oraz kolejki:
  - `qsub –l nodes=2:ppn=40 –q i7d skrypt.sh`
- Uruchomienie zadania z aktywnym przekierowaniem systemu X11:
  - `qsub -v DISPLAY=$DISPLAY -l nodes=1:ppn=64 -IX`

Uzyskiwanie pomocy

- System Wiki:
  - https://doc.cis.gov.pl/cis
  - Dostępna tylko wewnątrz sieci VPN;
  - Zawiera m.in. szczegółowe informacje na temat dostępnych zasobów sprzętowych, aplikacji, uruchamiania zadań;
  - Stanowi jeden z kanałów informowania o zmianach/nowościach wprowadzanych w systemie Klastra CIŚ;
- Helpdesk:
  - https://helpdesk.cis.gov.pl/otrs/customer.pl

PROTOKÓŁ MPI

Tobiasz Jarosiewicz

Warszawa
16 grudnia 2014

MPI

Message Passing Interface (**MPI**)

(z ang. *Interfejs Transmisji Wiadomości*) –protokół komunikacyjny będący standardem przesyłania komunikatów pomiędzy procesami programów równoległych działających na jednym lub więcej komputerach.

IMPLEMENTACJE PROTOKOŁU

- OpenMPI
- MVAPICH2
- Intel MPI

KOMPILACJA KODU

1. Ładowanie modułów

```
module load gcc
module load openmpi
```

2. Sprawdzenie zmiennych środowiskowych

```
echo $PATH
echo $LD_LIBRARY_PATH
```

3. Linkowanie biblioteki

```
gcc -I/usr/mpi/gcc/openmpi-1.4.3/include
```

Intel-MPI

1. Ładowanie modułów

```
module load intel-mpi
module load intel-composer
```

2. Linkowanie bibliotek

```
mpiicc -I/mnt/opt/tools/intel/(...)/include
```

MVAPICH

1. Ładowanie modułów

```
module load mvapich2
```

2. Linkowanie bibliotek

```
mpicc -
I/mnt/opt/tools/slc6/mvapich2/1.9a/include
```

URUCHAMIANIE

W obrębie jednego węzła obliczeniowego:

```
mpirun -np 40 <nazwa programu>
```

Zadania wykorzystujące wiele węzłów:

```
mpirun -f $PBS_NODEFILE -np <N> <nazwa programu>
```

Intel-MPI

```
mpirun -np <N> -f $PBS_NODEFILE <nazwa programu>
```

- Wybór warstwy fizycznej sieci do komunikacji między wątkami
  - Ethernet: `-genv I_MPI_FABRICS tcp`
  - InfiniBand: `-genv I_MPI_FABRICS ofa`
  - Shared memory: `-genv I_MPI_FABRICS shm`
  - Wybór wielu sieci: `-genv I_MPI_FABRICS shm:ofa`
- Przekazywanie zmiennych środowiskowych i opcji
  - `mpirun -genvall`
  - `mpirun -genv <opcje>`

Intel-MPI

- Przekazywanie zmiennych środowiskowych
  - `mpirun -genvall`
  - `mpirun -genv <opcje>`
- Optymalizacja
  - `mpirun -genv I_MPI_DYNAMIC_CONNECTION=1`
  - `mpirun -genv I_MPI_DAPL_SCALABLE_PROGRESS=0`

OpenMPI

```
mpirun -np <N> -machinefile $PBS_NODEFILE <nazwa programu>
```

- Wybór warstwy fizycznej sieci do komunikacji między wątkami
  - Ethernet: `--mca btl self,tcp`
  - InfiniBand: `--mca btl self,openib`
  - Shared memory: `--mca btl self,sm`
  - Wybór wielu sieci: `--mca btl self,sm,tcp,openib`

OpenMPI

- Przekazywanie zmiennych środowiskowych i opcji
  - `mpirun -x <nazwa zmiennej>=<path>`
- Optymalizacja
  - `--mca btl_openib_cpc_include rdmacm`
  - `--mca btl_openib_receive_queues P,1024,256,128,32`
  - `--mca btl_openib_receive_queues S,1024,256,128,32`

MVAPICH2

```
mpirun -np <N> -f $PBS_NODEFILE <nazwa programu>
```

- Wybór warstwy fizycznej sieci do komunikacji między wątkami
  - Ethernet: `-iface eth0`
  - InfiniBand: `-iface ib0`
  - Shared memory: `-env MV2_USE_SHARED_MEM`
- Przekazywanie zmiennych środowiskowych i opcji
  - `mpirun -genv <$nazwa zmiennej>=<path>`
- Optymalizacja
  - `MV2_USE_RDMA_CM 1`
  - `MV2_CM_RECV_BUFFERS 8192`
  - `MV2_ENABLE_AFFINITY 0`

DOKUMENTACJA

- Intel-MPI:

  software.intel.com/en-us/mpi-ug-lin-5.0.2-html

- OpenMPI:

  www.open-mpi.org/faq

- MVAPICH2:

  mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.0-userguide.html

Dziękuję za uwagę