Запуск задач на кластере ПТЦ

Гайдучок В. Ю. (gvladimiru@gmail.com),
Ганкевич И. Г (gig.spb@gmail.com)

План

- Введение
- Виртуальная машина
- Ресурсы центра
- PBS: основы
- Типы очередей
- Работа с кластером ТП
- Команды PBS
- Примеры
- Тестирование
- Заключение

  + Ссылки

Введение

- **Кластер** — группа компьютеров, объединённых высокоскоростными каналами связи и представляющая с точки зрения пользователя единый аппаратный ресурс.
- На базе центра развернуто несколько кластеров.
- Один служит для запуска виртуальных машин, другие — для высокопроизводительных пользовательских вычислений.
- Получив **виртуальную машину**, пользователь получает доступ к вычислительным кластерам, где он сможет запускать свои задачи на кластере.
- В случае если задача хорошо распараллелена, это **существенно уменьшит** время ее выполнения

Виртуальная машина

Доступ к вычислительным ресурсам осуществляется через виртуальную машину пользователя.

Назначение ВМ:

- Проведение вычислений
- Разработка приложений
- Хранение данных

Для получения виртуальной машины необходимо заполнить заявку, которую следует отправить по адресу noc@ptc.spbu.ru. Запросы на доступ к SMP машинам и на изменение конфигурации ВМ также следует отправлять на этот адрес.

Виртуальная машина

По умолчанию заводится виртуальная машина со следующими характеристиками:

- ОС: CentOS 5.6
- Количество ядер CPU: 4
- ОЗУ: 8Гб
- Жёсткий диск:

  /home — 50 Гб,

  Системные разделы — 8 Гб.

  Управление разбиением — lvm.

Виртуальная машина

Замечания

- Пользователь получает доступ к ВМ по ssh:

  ssh <user>@<address> -X

  («-X» для запуска графических приложений)

- Он имеет root привилегии для своей ВМ и может устанавливать и удалять программы, настраивать систему и т. д.

- В домашней директории по умолчанию будут присутствовать несколько скриптов, назначение которых объясняется дальше.

- На виртуальную машину могут быть установлены и другие ОС — требуемую ОС нужно указать в заявке (Unix, Linux).

Виртуальная машина

Директории

/ - корневой каталог (по умолчанию 8 Гб).

~/ - домашняя директория (по умолчанию 50 Гб). Ее размер указывается в заявке. Она смонтирована на отдельное устройство, данная папка доступна пользователю с **любого узла кластера**. Запущенная на кластере задача будет иметь доступ к данным в директории (задача запускается от имени пользователя).

/usr/local/ - директория, смонтированная на отдельное устройство. Данный каталог содержит программы доступные для использования (те программы, для которых у ПТЦ закуплены лицензии). Эта директория также доступна с **любого узла кластера T-Platform**.

Виртуальная машина

- Получив ВМ, пользователь может запускать свои задачи на ней.
- Он сможет устанавливать требуемое ПО на ВМ.
- Также ему будут доступны программы, закупленные ПТЦ (в основном ПО для научных расчетов, например, Matlab, Molpro, Crystal)
- Для задач, требующих больших вычислительных мощностей, пользователь может использовать ресурсы кластеров.
- Доступ к ним осуществляется через ВМ.

Ресурсы центра

Кластер виртуальных машин предназначен для работы ВМ пользователей (здесь запущены все ВМ).

Для пользовательских расчетов, требующих значительных вычислительных мощностей, используются следующие кластеры:

- **Кластер Т-Платформ**
- **SMP кластер**
- **Виртуальный кластер** (*несколько узлов кластера ВМ*)
- **Гибридный кластер** (*планируется*)

Ресурсы центра

Кластер виртуальных машин (HP)

- Блейд-сервер HP BL460G7:
  Процессоры: 2 x Intel Xeon X5670
  ОЗУ: 96GB RAM
  Сеть: 2 x 10GbE, 2 x QDR IB

- Система хранения HP:
  StorageWorks P4500 G2
  240 TB сырой ёмкости (120 x 2TB SAS HDD)

- Два коммутатора ProCurve E6600-48G-4XG
  (48 x 1GbE и 4 x 10GbE)

Итого: 60 серверов, 120 процессоров, 720 ядер, 5.76 TB RAM, 40Gbps IB, пиковая производительность 8.6 TFlop

Ресурсы центра

Сравнение выч. кластеров

	Кластер Т-Платформ T-EDGE96 HPC-0011828-001	**SMP кластер, HP Proliant DL980**	**Гибридный кластер, HP SL390s G7**
CPU	2x Intel E5335 2,0ГГц	8x Intel X7560 2,2 Ггц	2x Intel X5650 2,67 Ггц
Коммутатор	Infiniband 20 Гбит/с		
Дисковая память (на узел)	160 Гб	2 Тб	120 Гб
GPU	-	-	3x (8x) NVIDIA Tesla M2050
ОЗУ (на узел)	16 Гб	0,5-2 Тб	96 Гб
Сумм. ОЗУ	768 ГБ	3 ТБ	2.3 ТБ
Всего	48 узлов, 384 ядра	3 узла, 192 ядра	24 узла, 288 ядер, 112 GPU
Пиковая производительность	**3,07 Тфлопс**	**1,7 Тфлопс**	**59,6 Тфлопс**

Ресурсы центра

ПТЦ закуплено множество лицензий на различное ПО.
Программные продукты доступны в папке **/usr/local/** .

- ANSYS
- WIEN2k
- Molpro
- Crystal
- Gromacs
- Gaussian
- Firefly
- Mathematica
- Matlab
- Maple
- Comsol
- Vorpal*
- HyperChem*
- ...

Компиляторы:

- NAG
- PGI
- Lahey*
- Intel
- Platform MPI
- ...

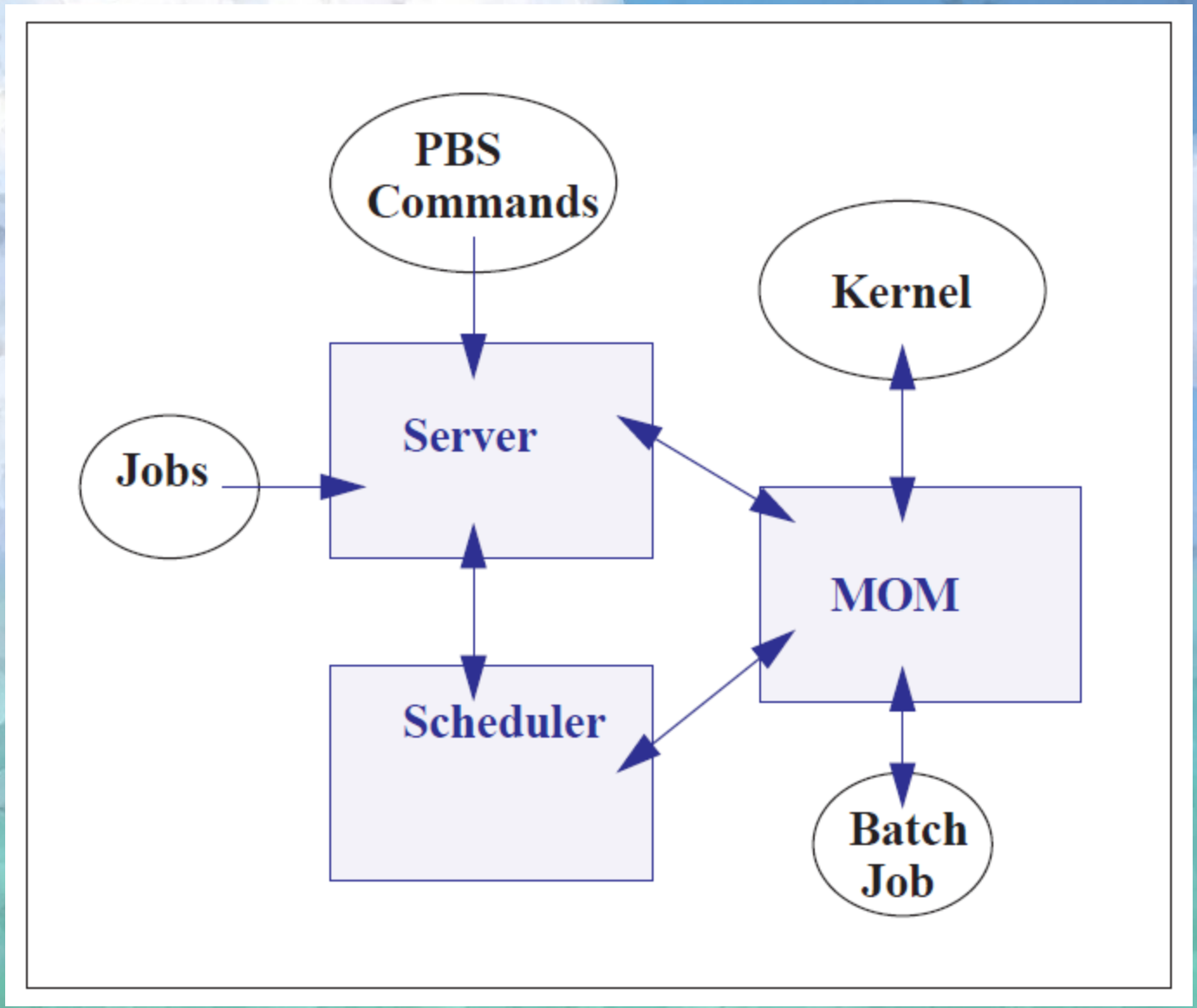
С результатами тестирования ПО можно ознакомиться на сайте.

PBS: основы

PBS - Portable Batch System — система управления распределенными вычислениями. Основная функция PBS — запуск вычислительных задач в вычислительной среде по расписанию. Наиболее часто используется для управления вычислительным процессом в кластерах.

На кластере Т-Платформ установлен TORQUE и Maui, а а на SMP — PBS Pro.

PBS: основы



Типы очередей

	short	long	infi	smp
приоритет	высокий	средний	низкий	-
процессорное время (макс.)	6ч	48ч	-	-
память на одном узле	2Гб			0,5-2 Тб
реальное время (макс.)	9ч	72ч	-	-
макс. число выполн. задач	3			
макс. задач в очереди	6			

short, long, infi — очереди на кластере Т-Платформ.
smp — очередь на SMP кластере.

Работа с кластером ТП

Для работы с кластером ТП создано несколько удобных скриптов.
По умолчанию они расположены в домашней директории пользователя.
Также там есть папка examples с тестовыми примерами.
Подробная документация - http://ptc.spbu.ru/hpc/

Основные скрипты для работы:

- ~/**submit-tp** — запуск задачи (постановка в очередь)
- ~/**status-tp** — просмотр состояния задач на кластере и дополнительной информации
- ~/**qdel** — удаление задачи из очереди (отмена поставленной задачи)

Работа с кластером ТП

submit-tp: примеры

Запуск скрипта helloworld.sh (на 1 ядре):

```
~/submit-tp -f ~/examples/helloworld.sh
```

Запуск my_script.sh на 32 ядрах:

```
~/submit-tp -n 32 -f ~/my_script.sh
```

Запуск my_script.sh с параметрами («abc») на 16 ядрах с отправкой письма при старте, окончании или отмене задачи:

```
~/submit-tp -n 16 -m abe -f "~/my_script abc"
```

Запуск my.sh на 64 ядрах с именем задачи «my» в очереди «short»:

```
~/submit-tp -n 64 -j my -q short -f "~/my.sh"
```

Работа с кластером ТП

submit-tp: примеры

После постановки задачи в очередь (выполнения ~/submit-tp) в консоль будет выведено сообщение вида 21790.pbs-tp.hpc.cc.spbu.ru — здесь указано ID задачи.

По окончании выполнения в папке, откуда был вызван submit-tp появятся 2 файла вида <имя задачи>.e<id> и <имя задачи>.o<id>, например:

```
job.1328549662.e21464
job.1328549662.o21464
```

Где файл <имя>.o<id> содержит стандартный вывод (stdout) выполненной задачи, а <имя>.o<id> содержит ошибки (все то, что программа записала в stderr)

Работа с кластером ТП

Пример задачи

Программа test использует MPI. Для запуска программы test на кластере необходимо написать скрипт (здесь скрипт назван ~/myscr.sh):

```
#!/bin/bash
NP=`cat $PBS_NODEFILE | wc -l`
mpirun -machinefile $PBS_NODEFILE \
                -np $NP ~/test "$1"
```

Запуск:

```
~/submit-tp -j MyJob -n 16 \
        -f "~/myscr.sh ~/new/myFile"
```

Задача myJob будет поставлена в очередь и выполнена на 16 ядрах. В результате будет создано 2 файла: MyJob.o21233 и MyJob.e21233 .

Работа с кластером ТП

test.cpp

```
int main(int argc, char* argv[]) {
    ……..
    MPI_Init(&argc, &argv);
    MPI_Comm_size(MPI_COMM_WORLD,
                  &prNum);
    MPI_Comm_rank(MPI_COMM_WORLD,
                  &rank);
    MPI_Get_processor_name(name, &length);
    printf("rank=%d node=%s\n", rank, name);
    if (rank == 0)
        if (argc > 1) {
            fileToRead = fopen(argv[1], "r");
            if (fileToRead != NULL) {
                printf("File is found!\n");
                fclose(fileToRead);
            }
        }
    MPI_Finalize();
    return(0);
}
```

Результат

(содержимое MyJob.o21233)

```
File is found!
rank=0 node=node-ib-1
rank=1 node=node-ib-1

……..

rank=14 node=node-ib-4
rank=15 node=node-ib-4
```

Работа с кластером ТП

submit-tp: примеры

Внимание! Программа, запущенная на выполнения должна иметь дело с данными, находящимися в **домашней директории пользователя**. Туда же должен быть записан и результат, так как **~/** - папка, доступная для всех узлов кластера (для чтения и записи).
Само приложение может находится либо в домашней директории, либо в **/usr/local** (доступна для чтения).

Следующие примеры выполнятся нормально:

```
/usr/local/app ~/tests/input
~/apps/myApp ~/new/a.txt -o ~/res/out
```

А этот пример вызовет ошибку (no such file)

```
~/apps/myApp /tmp/a.txt
```

Работа с кластером ТП

Замечания

• Запуск задачи аналогичен старту новой сессии — каждый раз при запуске задачи **на кластере** будет выполнен ~/.bashrc (в случае bash) для каждого процесса.

• Для многих программ в домашней директории уже существуют скрипты, делающие всю подготовительную работу. Например, ~/submit-wien.sh (будет рассмотрен дальше), ~/submit-crystal и т. д.

• Использование уже существующих скриптов существенно упрощает работу.

• Увеличение числа задействованных ядер **не всегда** приводит к существенному ускорению (**закон Амдаля**)

Работа с кластером ТП

status-tp: примеры

Следующая команда выводит информацию о задачах в очередях:

```
~/status-tp -a
```

Результат:

```
                                                                  Req'd    Req'd    Elap
Job ID          Username Queue    Jobname     SessID NDS TSK Memory Time  S Time
--------------- -------- -------- ----------- ------ ---- ---- -------- ------ - ------
21897           igankevich long     MPPcrystal 151834    1    4       --      --  R 119:3
   node1/1, node1/2, node1/3, node4/4
21898           igankevich long     test.sh         --   4    8       --      --  Q   --
   --
```

Как видно, в очереди стоят 2 задачи, запущенные пользователем igankevich, одна из которых имеет ID=21897, имя MPPCrystal и требует 4 ядра на одном узле. Она уже считается (статус R = Run). Здесь же указаны узлы, на которых она выполняется (node1/1 ...).

Вторая задача с ID=21898 не выполняется, она пока ждет ресурсов в очереди (статус Q = Queued).

Работа с кластером ТП

status-tp: примеры

Следующая команда сообщит общее число доступных ядер и число свободных ядер:

```
~/status-tp -p
```

Результат:

Total_TP= 344
Free_TP= 196

Как видно, из 344 доступных ядер свободно 196.

Команды PBS

Теперь будет дан небольшой обзор команд PBS.
Их довольно много, здесь будут указаны лишь важнейшие из них с описанием основных опций.
Пользователям рекомендуется работать со скриптами.

Основные команды:

- **qsub** - запуск задачи на выполнение
- **qdel** - удаление задачи из очереди (отмена поставленной задачи)
- **qstat** - просмотр состояния задач на кластере и дополнительной информации
- **qalter** - изменение некоторых параметров задачи
- **xpbs** - запуск графического интерфейса (GUI)

Команды PBS

qsub: примеры

Запуск задачи test.sh на SMP кластере:

```
/opt/pbs/default/bin/qsub \
    -q smp -l nodes=2:ppn=4 ~/smp_test.sh
```

Задача будет поставлена в очередь smp (на SMP), число узлов — 2, число ядер — 4 (итого — 8 ядер).

```
/usr/bin/qsub -q long -m abe -M a@b.com \
-l nodes=3:ppn=4,mem=100mb ~/smp_test.sh
```

Задача будет поставлена в очередь long (на T-Platform), задействовано 3 узла, 4 ядра на каждом (итого — 12 ядер), используя 100 Мб памяти с отсылкой письма.

Команды PBS

Использование команд

- Удаление задачи:
  `qdel 21897`

- Просмотр состояния задач:
  `qstat`

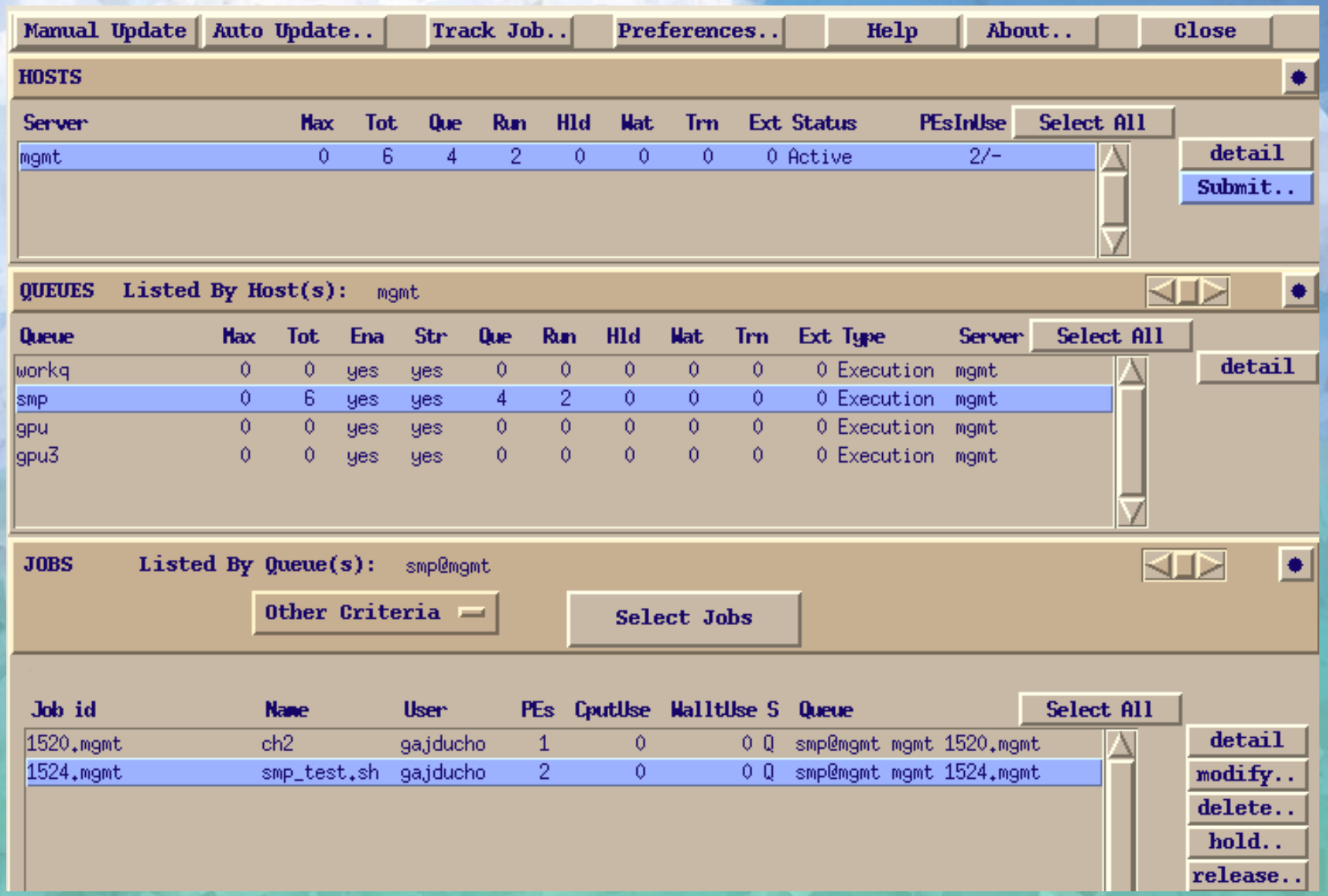
- Просмотр состояния задач с указанием узлов:
  `qstat -n`

- Просмотр состояния всех очередей:
  `qstat -Q`

- Изменение имени задачи:
  `qalter -N ch2 1520.mgmt`

Команды PBS

xpbs: графический интерфейс



Примеры

Запуск простой задачи

Запуск с использованием submit-tp (на T-Platforms):

```
~/submit-tp -n 16 -f ~/examples/helloworld.sh
```

Запуск с использованием qsub на T-Platforms:

```
/usr/bin/qsub -q long -l nodes=4:ppn=4 \
        ~/examples/helloworld.sh
```

Запуск с использованием qsub на SMP:

```
/opt/pbs/default/bin/qsub -q smp \
   -l nodes=4:ppn=4 ~/examples/helloworld.sh
```

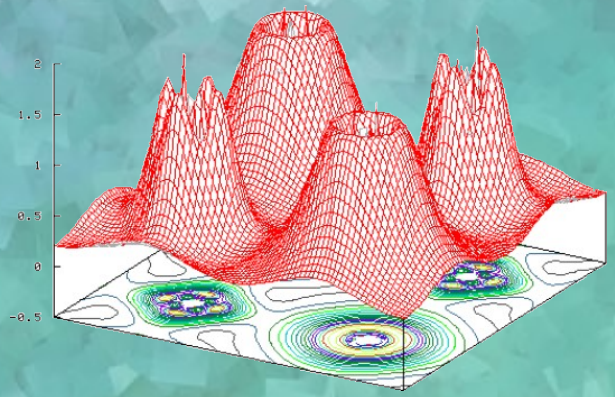
Стоит отметить, что для SMP кластера расположение некоторых программ отличается от расположения на кластере T-Platforms.

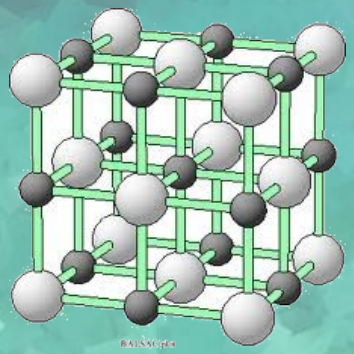
Примеры

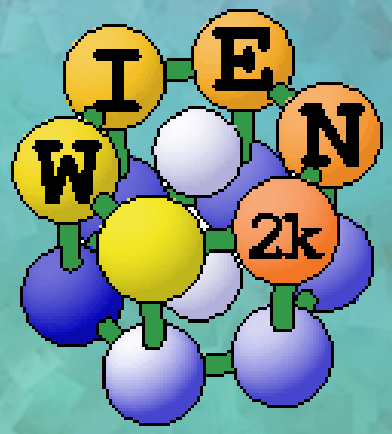
Запуск задачи WIEN2k

Для работы WIEN2k необходимо добавить в ~/.bashrc строки: **`source ~/.bashrc.wien2k`**

В процессе выполнения скрипта **`~/submit-wien2k.sh`** задаются требуемые переменные окружения, создаются необходимые директории, формируется файл .machines . Файл .machines содержит список узлов с указанием числа запускаемых на них процессов. Он формируется на основе переменной **PBS_NODEFILE** .







Примеры

Запуск задачи WIEN2k

```
~/submit-tp -n 16 -j WIEN_Fe
   -f "~/submit-wien2k.sh ~/wien/Fe"
```

-n 16 — указывает использовать 16 ядер;
-j WIEN_Fe — задает имя задачи (WIEN_Fe);
-f … — указывает скрипт для исполнения (задачу):
~/submit-wien2k.sh — скрипт, который будет запущен
~/wien/Fe — параметр для этого скрипта.

Здесь ~/wien/Fe - папка с исходными данными. В эту же папку будет записан результат. Данная папка «видна» всем узлам, участвующим в вычислении, так как находится в **домашней директории** пользователя, а задача будет запущена от имени **данного пользователя**.

Примеры

Запуск задачи WIEN2k

Результаты вычислений будут находится в папке **`~/wien/Fe`** , сообщения stdout (стандартный вывод программы) в файле WIEN_Fe.o21789, сообщения stderr в файле WIEN_Fe.e21789 (полагаем, что ID задачи=21789).

Примерное содержимое WIEN_Fe.o21789:

```
LAPW0 END
LAPW1 END
LAPW2 - FERMI; weighs written
LAPW2 END
SUMPARA END
CORE  END
MIXER END

real    951m33.184s
user    707m17.641s
sys     109m46.225s
```

Примеры

Запуск задачи Crystal

```
mkdir -p ~/crystal/inputs
cp ~/examples/crystal/tio2pr.d12 \
                ~/crystal/inputs
~/submit-tp -q short -f ~/submit-crystal.sh \
     -n 24 -j cr09_tio2pr-n4 -i tio2pr
```

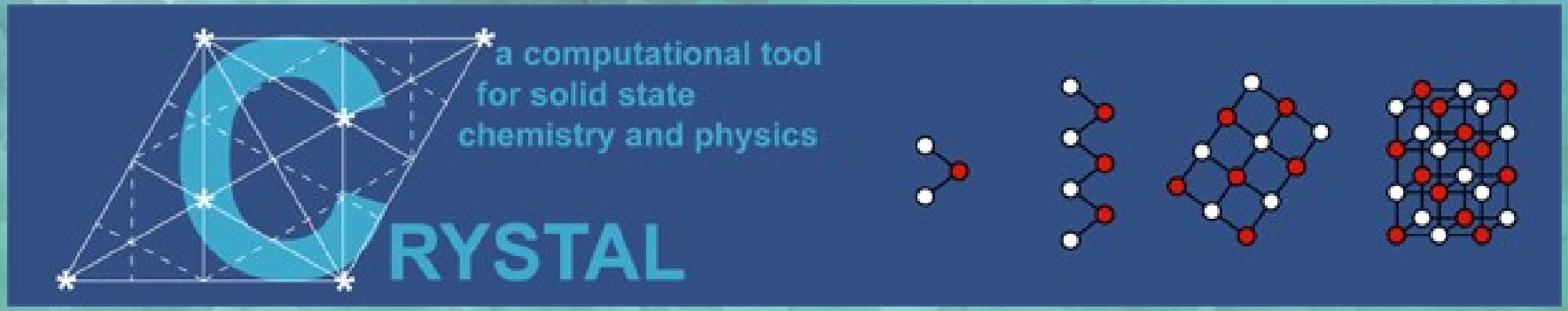
Здесь создается папка **~/crystal/inputs** , в нее копируется требуемый входной файл.
Затем вызывается скрипт ~/submit-tp: задача ставится в очередь «short», запрашивается 24 ядра, задается имя задачи («cr09_tio2pr-n4»), а опция «-i» указывает входной файл. Результат будет располагаться в ~/crystal/tio2pr.***** , сообщения stdout в cr09_tio2pr-n4.o<id>, сообщения stderr в файле cr09_tio2pr-n4.e<id>

Тестирование

Далее приведены результаты тестирования двух вариантов программы **Crystal** (Pcrystal и MPPcrystal). На графиках показано ускорение вычислений.

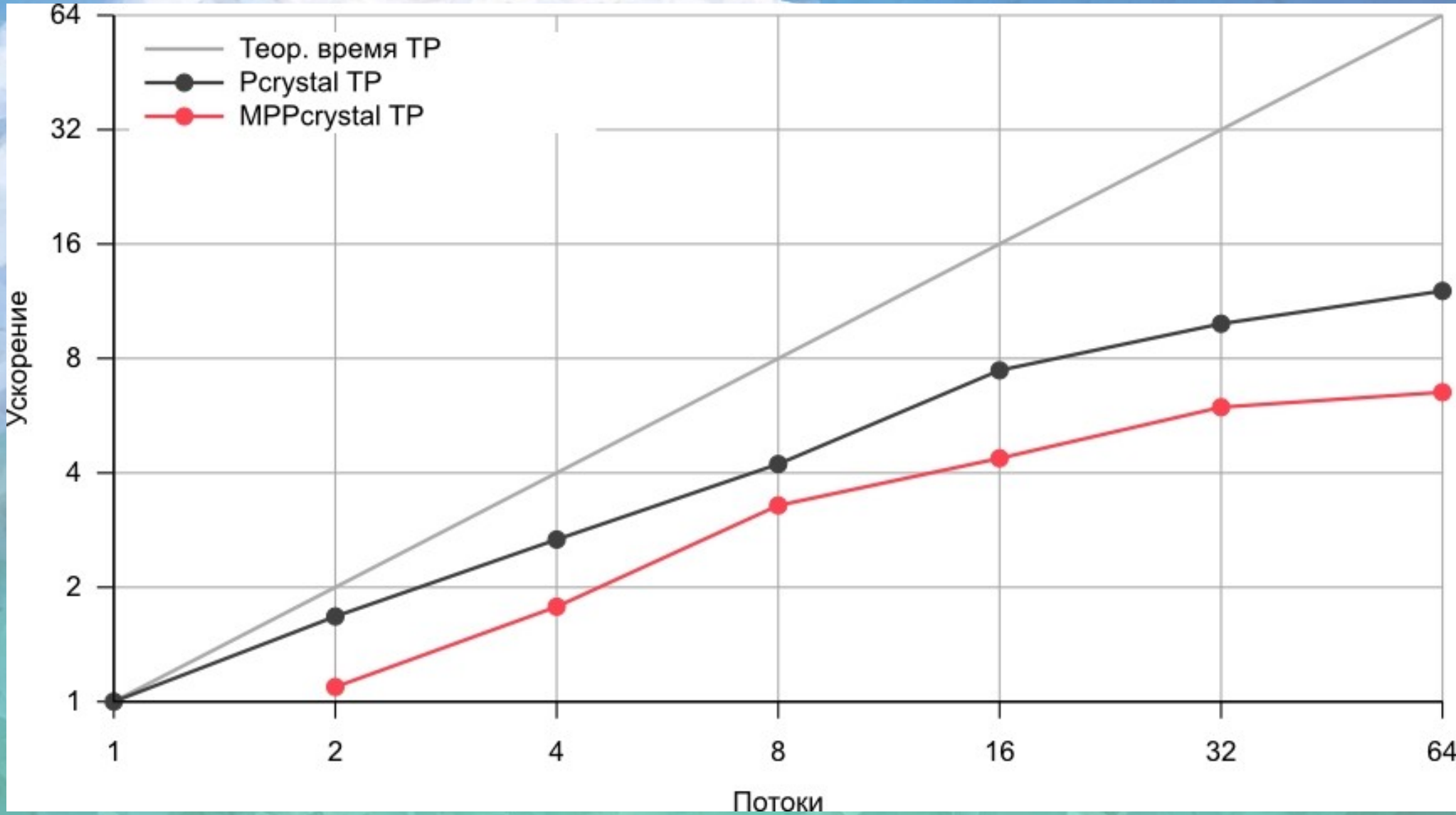
Тестирование проводилось как на кластере Т-Платформ, так и на SMP кластере.

Были проведены тесты на 1, 2, 4, 8, 32 и 64 ядрах (плюс 128 и 192 для SMP).



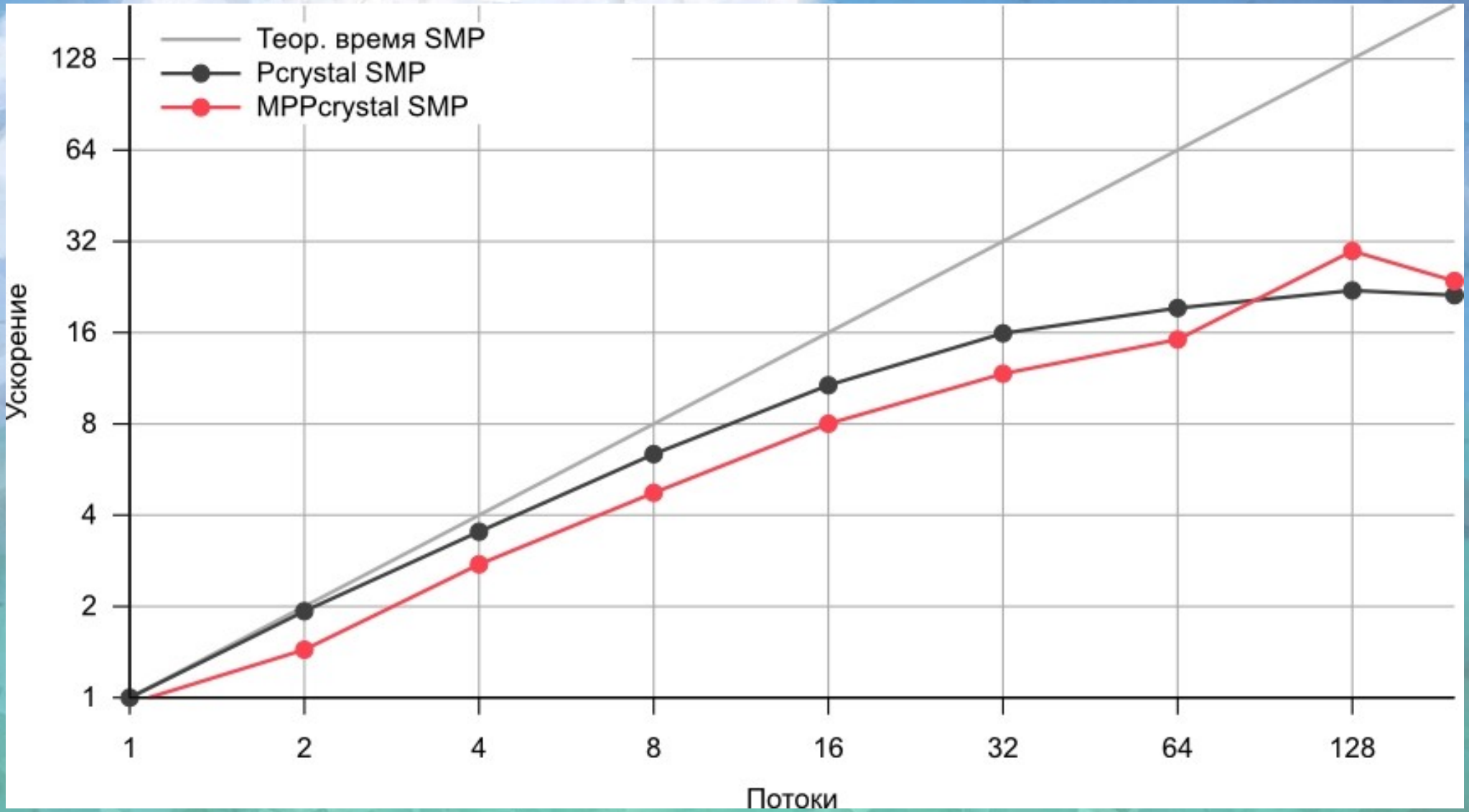
Тестирование

Кластер Т-Платформ



Тестирование

SMP кластер



Заключение

- ПТЦ предоставляет пользователям мощности вычислительных кластеров.

- На данный момент пользователям доступны кластер Т-Платформ, SMP кластер и виртуальный кластер. В ближайшее время будет введен в эксплуатацию гибридный кластер.

- Работа с вычислительными кластерами осуществляется через виртуальные машины.

- Для работы с кластером существует множество удобных скриптов, облегчающих запуск и мониторинг задач.

Ссылки

- http://ptc.spbu.ru/ - официальный сайт ПТЦ.
- http://ptc.spbu.ru/hpc/ - описание работы с кластером, множество примеров, запуск различных задач на кластере, информация об установленном ПО и об аппаратном обеспечении ...
- http://www.ptc.spbu.ru/zayavlenie_vm.doc - заявка на получение виртуальной машины.
- http://v105.ptc.spbu.ru/arch/user/ - скрипты для работы с кластером (по умолчанию уже добавлены в домашнюю директорию пользователя).
- http://www.pbsworks.com/SupportDocuments.aspx , TORQUE - документация (англ.) по PBS и TORQUE.

Вопросы

Спасибо