

Пакеты прикладных программ на OpenPOWER кластере

Бойков Е.А.

ВЦ ДВО РАН

2017-09-14

Информационные ресурсы для работы с кластером

- ▶ Документация пользователя кластера
 - ▶ <http://projects.ccfеbras.ru/projects/kb/wiki/PowerUserDoc>
 - ▶ Открытые ресурсы Redmine
 - ▶ Нет обратной связи
- ▶ Работа с заявками и ответы на вопросы
 - ▶ <http://ckp.ccfеbras.ru>
 - ▶ Простая система вопросов/ответов, не подразумевающая выразительного оформления и функциональности helpdesk и/или QA
- ▶ Внутренняя система управления задачами Redmine
 - ▶ <http://projects.ccfеbras.ru/projects/grid/issues>
 - ▶ Не для всех пользователей
- ▶ База знаний для всех пользователей
 - ▶ Либо добавить ресурс Redmine для всех пользователей с обеспечением нужной функциональности Redmine
 - ▶ Либо предложить отдельное решение

Пакеты прикладных программ по заявкам

- ▶ Quantum ESPRESSO 5.4, 6.0
 - ▶ PLUMED, QE-GIPAW, WEST
- ▶ ABINIT 8.2
- ▶ Заявки ЦКП <http://ckp.ccfеbras.ru>
 - ▶ Уже установленные
 - ▶ 70. C++, OpenACC, CUDA
 - ▶ 229. ESSL
 - ▶ 134. fortran, python 2.7
 - ▶ 233. Современные компиляторы языка C++ (стандарт C++ 14) и расширений (CUDA) для поддержки вычислений на GPU
 - ▶ MS Windows
 - ▶ 78. OpenCL (>0.9), Mono, .Net Framework 4.5
 - ▶ 79. MS Visual Studio (13, 15) SDK for Open CL (6)

Пакеты прикладных программ по заявкам

- ▶ Контейнеры
 - ▶ 74. numpy, tensorflow-gpu, CUDA® Toolkit 8.0 (with drivers), cuDNN v5.1, libcupti-dev, keras
 - ▶ 195. Операционная система ubuntu, Docker \geq 1.9, NVIDIA drivers \geq 340.29, nvidia-docker-plugin (см. <https://github.com/NVIDIA/nvidia>), docker TensorFlow 1.2 with GPU support, numpy virtualenv tmux
 - ▶ 232. Python 2.7 64-bit, CUDA 8.0, TensorFlow, cuDNN 6.1
 - ▶ 228. OC Debian, ansible, container
- ▶ Незавершенные и новые
 - ▶ 219. PyCUDA, GRASS, snap-engine, gdal, pygrass, proj4, geos
 - ▶ задача вычисляется на кластере x86
 - ▶ 215. GAMESS-US
 - ▶ Ошибка при сборке программы, связанная с архитектурой ppc64le
 - ▶ 230. VASP
 - ▶ 249. Torch7, TensorFlow, IBM ESSL, Lammps

Особенности работы с прикладными программами

- ▶ 2 плохо совместимых режима работы
 - ▶ Пакетный
 - ▶ Классические кластеры HPC
 - ▶ Используется планировщик и привычный программный стек
 - ▶ Интерактивный
 - ▶ Востребован в некоторых задачах
 - ▶ Нет готового решения: нужно выделять узлы в отдельных кластер, работающий иначе
- ▶ С системным ПО нет проблем, подобных тем, которые возникают с прикладными программами
 - ▶ Настройка прикладной программы часто требует участия и взаимодействия нескольких специалистов
 - ▶ Непривычная архитектура ррс64le создает больше трудностей в доводке прикладных программ
 - ▶ Некоторые программы требуют использования контейнеров

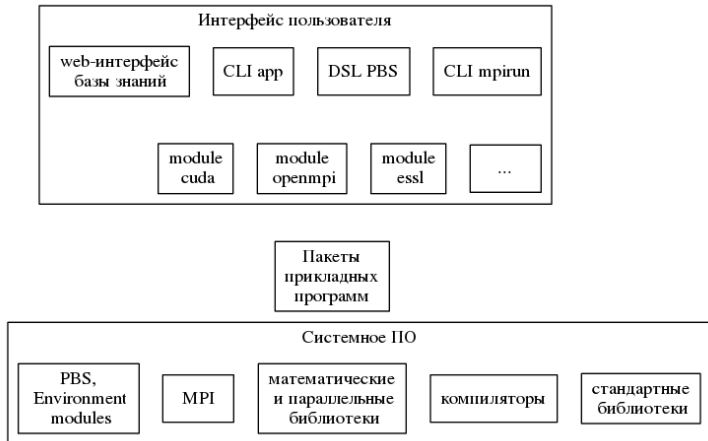
Системное ПО

- ▶ Стандартные библиотеки
 - ▶ glibc.ppc64le - 2.17
 - ▶ glib2.ppc64le - 2.46.2
 - ▶ libstdc++.ppc64le - 4.8.5
- ▶ MPI
 - ▶ IBM Spectrum MPI - 10.1.0.3
 - ▶ OpenMPI - 1.10
- ▶ Математические и параллельные библиотеки
 - ▶ IBM ESSL - 5.5
 - ▶ NVIDIA cuDNN - 5.1

▶ Компиляторы

- ▶ IBM XL C/C++ for Linux - 13.1.5
- ▶ IBM XL Fortran for Linux - 15.1.5
- ▶ NVIDIA Cuda compiler - 8.0.61
- ▶ PGI Compilers - 16.10
- ▶ PGI Compilers - 17.4
- ▶ gcc.ppc64le - 4.8.5
- ▶ gcc-c++.ppc64le - 4.8.5
- ▶ gcc-gfortran.ppc64le - 4.8.5
- ▶ libgcc.ppc64le - 4.8.5

Программный стек кластера и пользовательский интерфейс



Виды интерфейсов

- ▶ Интерфейс - это способ взаимодействия
- ▶ Не только GUI и CLI: много видов интерфейсов
- ▶ Каждый вид интерфейса решает определенные задачи
 - ▶ GUI
 - ▶ Интуитивно понятные элементы управления, низкий порог вхождения
 - ▶ Web-интерфейс (Redmine, Ganga)
 - ▶ GUI, доступный везде
 - ▶ Выразительное представление информации (в т.ч. документации, рецептов)
 - ▶ Облегчает взаимодействие специалистов
 - ▶ Языки разметки (HTML, Textile)
 - ▶ Связь между Web-интерфейсами и языками программирования
 - ▶ Языки программирования
 - ▶ Сами языки, функции, программы, форматы данных являются интерфейсами

Виды интерфейсов

- ▶ Формат Plain Text
 - ▶ Формат простого текста - универсальный интерфейс
- ▶ Система контроля версий (git)
 - ▶ Интерфейс между простым текстом и отслеживанием изменений
- ▶ CLI программы
 - ▶ Взаимодействие через аргументы, конвейеры команд, склеивание программ
- ▶ Пакетный режим
 - ▶ Взаимодействие между CLI программами и планировщиком
- ▶ DSL - язык, отражающий специфику предметной области
 - ▶ CLI, директивы PBS, HTML, Textile - это DSL
- ▶ Контейнер
 - ▶ Интерфейс между ядром Linux и программным стеком

Environment Modules

- ▶ Большую сложность можно скрыть за очень простым интерфейсом

```
$ module avail
----- /etc/modulefiles -----
cuda                essl                openmpi/pgi/1.10.2/2017  pgi/17.4
cudnn/5.1           mpi/mvapich2-ppc64le  openmpi/xl/1.10.6        pgi/2016
espresso/cpu/5.4    openmpi/gcc/1.10.6/4.8.5  openmpi/xl/2.0.2a1      pgi/2017
espresso/gpu/5.4    openmpi/gcc/2.0.2a1/4.8.5  openmpi/xl/2.1.0        singularity
espresso/gpu/5.4-esslcuda  openmpi/gcc/2.1.0/4.8.5  pbs                      spectrum_mpi
espresso/gpu/6.0    openmpi/pgi/1.10.2/2016  pgi/16.10(default)
```

```
$ module list
1) pbs                2) cuda              3) essl              4) spectrum_mpi
```

Пример описания пакетной задачи

- ▶ Пользовательский интерфейс кластера состоит из Web-интерфейса базы знаний и интерфейса описания задачи умеренной сложности

```
#!/bin/sh
#PBS -N qe_job_name
#PBS -q workq
#PBS -j oe
#PBS -l select=2:ncpus=160:ngpus=2:mpiprocs=20

# Загружаем необходимые модули
module add spectrum_mpi essl espresso/gpu/5.4

# Переходим в рабочую директорию
cd /home/user/qe_working_dir

# Запускаем исполняемый файл Quantum ESPRESSO "pw-gpu.x"
mpirun -np 40 -npernode 20 --hostfile $PBS_NODEFILE --bind-to core $QE_HOME/bin/pw-gpu.x < ausurf.in

exit 0
```

ППП на OpenPOWER кластере

- ▶ Специфика OpenPOWER кластера
 - ▶ ppc64le, GPU: порождает множество нерешенных задач
- ▶ Задача: установка, отладка, настройка ППП
 - ▶ Решение:
 - ▶ Отслеживание изменений, воспроизводимость
 - ▶ Передача задачи между специалистами
 - ▶ Получение рецепта
- ▶ Задача: создание документации и примеров запуска
 - ▶ Решение:
 - ▶ Представление рецептов в базе знаний
 - ▶ Устранение дублирования, устаревания
 - ▶ Обратная связь с пользователями в базе знаний
- ▶ Задача: определение параметров планировщика, MPI, environment modules, GPU, тонких настроек CPU и GPU
 - ▶ Решение:
 - ▶ Накопление рецептов в базе знаний
 - ▶ Расширение рецептов на другие случаи
 - ▶ Поддержка актуальности базы знаний в случае изменения конфигурации кластера

Спасибо за внимание!