

Документация по скриптам запуска программного обеспечения

1. Введение

Система скриптов запуска - это унифицированная система для выполнения расчетов на вычислительном кластере. Предоставляет стандартизированный способ настройки вычислительных ресурсов (CPU, GPU, память, время выполнения) и отправки задач в очередь менеджера задач SLURM.

Система поддерживает более 20 научных пакетов программ по направлениям физика, математика, химия и другим, включая Gaussian, GROMACS, ANSYS, COMSOL, CP2K, Crystal, WIEN2k и другие.

2. Работа с модулями

Lmod - система управления модулями, которая позволяет гибко настраивать программное окружение для работы с различным программным обеспечением. Самый простой способ подготовить окружение для конкретной версии программного обеспечения - использовать систему модулей. Это позволяет избежать конфликта между версиями ПО и упрощает настройку переменных окружения.

Основные команды Lmod:

- `module avail` - показать список доступных модулей
- `module load <имя_модуля>` - загрузить модуль
- `module list` - показать загруженные модули
- `module unload <имя_модуля>` - выгрузить модуль
- `module purge` - выгрузить все модули

Примеры загрузки модулей:

```
module load gaussian/submit
```

Рекомендации:

- Загрузка нужных модулей должна выполняться перед запуском задач
- Для автоматизации можно добавлять команды `module load` в конфигурационные файлы (например, `.bashrc`)
- Для очистки окружения от всех загруженных модулей используйте команду `module purge`

3. Запуск задач

Для запуска расчетов на вычислительном кластере используйте специализированные скрипты отправки задач. Эти скрипты обеспечивают унифицированный интерфейс для настройки вычислительных ресурсов и управления задачами в системе SLURM.

3.1. Общая структура запуска

`<скрипт_запуска> <входной_файл_или_директория> [опции...]`

- `<входной_файл_или_директория>` : Обязательный параметр. Путь к входному файлу задачи или к рабочей директории (зависит от программы).
- `[опции...]` : Необязательные (если не указано иное) параметры для настройки ресурсов и поведения задачи.

3.2. Общие опции

Эти опции доступны для большинства скриптов запуска программ.

- `-A <account>` : **Обязательный** параметр. Номер исследования (проекта) в системе управления ресурсами (например, RIMS).
- `-m <ресурс>` : *Опциональный* параметр. Универсальное имя вычислительного ресурса (например, `cluster` , `vm`). Значение по умолчанию `cluster` .
- `-q <очередь>` : *Опциональный* параметр. Раздел (очередь) кластера, на котором будет запущена задача (например, `basic` , `p100` , `ds`). Значение по умолчанию `basic` . Смотрите спецификацию ресурсов.
- `-c <значение>` : *Опциональный* параметр. Общее количество ядер CPU.

- `-n <значение>` : *Опциональный* параметр. Количество вычислительных узлов. Используется совместно с `-ppn`.
- `-ppn <значение>` : *Опциональный* параметр. Количество ядер на узел. Используется совместно с `-n`.
- `-g <значение>` : *Опциональный* параметр. Общее количество GPU.
- `-gp <значение>` : *Опциональный* параметр. Количество GPU на узел.
- `-L <время>` : *Опциональный* параметр. Время выполнения задачи. Формат: минуты, часы:минуты, дни-часы, дни-часы:минуты:секунды и пр. Значение по умолчанию может быть 48 часов.
- `-Q <qos>` : *Опциональный* параметр. Уровень качества обслуживания (Quality of Service).
- `-h` : *Опциональный* параметр. Показать справку по доступным опциям.

3.3. Специфичные опции для групп программ

3.3.1. Программы, принимающие один входной файл

К этой группе относятся: Gaussian, Quantum Espresso, ORCA, Abinit, GULP, Anaconda (Python).

- **Входной файл:** Указывается как первый аргумент после имени скрипта запуска.
 - **Gaussian:** Файлы с расширением `.gjf` или `.com`.
 - **ORCA:** Файлы с расширением `.inp`.
 - **COMSOL:** Файлы с расширением `.mph`.
- **Дополнительные опции:**
 - **Gaussian:**
 - `.rwf` : Файлы `.rwf` (рабочие файлы) будут скопированы в рабочую директорию.
 - `.chk` : Файлы `.chk` (checkpoint файлы) будут скопированы в рабочую директорию.
 - **Quantum Espresso:**
 - `-e <исполняемый_файл>` : **Обязательный** параметр. Имя исполняемого файла QE (например, `pw.x`, `ph.x`).
 - `-a <аргументы>` : *Опциональный* параметр. Дополнительные аргументы, передаваемые исполняемому файлу.
 - **ORCA:**
 - `-t` : *Опциональный* параметр. Использовать временную директорию для выполнения задачи.

- **Crystal:**

- `.d12` : Автоматически распознаваемое расширение для входного файла.
- `.d3` : Файлы `.d3` (данные для Dispersion).
- `.f9` : Файлы `.f9` (данные для Fock matrix).
- `-MPP` : Использовать MPP версию исполнителя.
- `-jobdir <директория>` : Указать рабочую директорию.
- `-P` : Запустить Properties после Crystal.
- `-nocrystal` : Не запускать Crystal, только Properties.

- **Anaconda (Python):**

- `-e <окружение>` : *Опциональный* параметр. Имя Conda окружения (по умолчанию `base`).
- `-u` : *Опциональный* параметр. Отключить буферизацию вывода (по умолчанию включена).

3.3.2. Программы, работающие в рабочей директории

К этой группе относятся: GROMACS, ANSYS.

- **Рабочая директория:** Указывается как первый аргумент после имени скрипта запуска.

- **Дополнительные опции:**

- **GROMACS:**

- `-a <аргументы>` : *Опциональный* параметр. Аргументы, передаваемые `mdrun_mpi`.
- `--threads <количество>` : *Опциональный* параметр. Количество потоков OpenMP на процесс MPI.
- `-n <узлы>` : *Опциональный* параметр. Количество узлов (альтернатива `-s`).

- **ANSYS:** Дополнительные опции отсутствуют.

3.3.3. Программы с особым синтаксисом запуска

К этой группе относятся: WIEN2K, COMSOL.

- **WIEN2K:**

- Первым аргументом указывается исполняемый файл или команда (например, `x lapw1`).
- **Дополнительные опции:**

- -kr <количество> : *Опциональный* параметр. Количество k-точек.
 - -mp <количество> : *Опциональный* параметр. Количество MPI процессов.
 - -M <память> : *Опциональный* параметр. Память на k-точку.
 - -s <директория> : *Опциональный* параметр. Директория для временных файлов.
 - -T : *Опциональный* параметр. Сохранить временные файлы.
- **COMSOL:**
 - Первым аргументом указывается файл с расширением .mph .
 - Дополнительные опции отсутствуют.

3.4. Примеры запуска

Пример 1: Gaussian

```
# Запуск Gaussian 16 на двух узлах раздела *extended2* с 56 ядрами, время 4
submit-gaussian-16 $HOME/test-gaussian/test.com -A 2302-005-081 -c 56 -L 4-
# или с указанием узлов и ядер на узел
submit-gaussian-16 $HOME/test-gaussian/test.com -A 2302-005-081 -n 2 -ppn 2
```

Пример 2: Quantum Espresso (QE)

```
# Запуск Quantum Espresso 6.8.0 (pw.x) на одном узле *basic* с 16 ядрами, в
submit-qe-6.8.0 input.in -e pw.x -A 1234-567-890 -c 16 -L 12:00:00 -m clust
```

Пример 3: ORCA

```
# Запуск ORCA 5.0.3 на одном узле *extended1* с 8 ядрами, используя временн
submit-orca-5.0.3 input.in -A 1234-567-890 -c 8 -L 24:00:00 -m cluster -q e
```

Пример 4: Crystal

```
# Запуск Crystal 14 на одном узле *extended1* с 8 ядрами, время 24 часа
submit-crystal-14.1.0.4 $HOME/test-crystal/test.d12 -A 1234-567-890 -c 8 -L
```

Пример 5: CP2K

```
# Запуск CP2K 8.2.0 на одном узле *basic* с 16 ядрами, время 12 часов
submit-cp2k-8.2.0 $HOME/test-cp2k/input.inp -A 1234-567-890 -c 16 -L 12:00:
```

Пример 6: GROMACS

```
# Запуск GROMACS 2019 в директории $HOME/test-gromacs в разделе *P100* с 4  
submit-gromacs-2019 $HOME/test-gromacs -A 1234-567-890 -n 4 -ppn 28 -gn 1
```

Пример 7: WIEN2K

```
# Запуск WIEN2K 12.1 с 2 узлами по 8 ядер, 4 k-точки и 2 MPI процесса, врем  
submit-wien2k-12 x lapw1 -A 1234-567-890 -n 2 -ppn 8 -kp 4 -mp 2 -L 24:00:0
```

Пример 8: Abinit

```
# Запуск Abinit 9.4.2 на одном узле раздела *basic* с 12 ядрами, время 12 ч  
submit-abinit-9.4.2 input.in -A 1234-567-890 -c 12 -L 12:00:00 -m cluster -
```

Пример 9: ANSYS

```
# Запуск ANSYS 19.1 на одном узле раздела *basic* с 8 ядрами, время 12 часо  
submit-ansys-19.1 input.dat -A 1234-567-890 -c 8 -L 12:00:00 -m cluster -q
```

Пример 10: COMSOL

```
# Запуск COMSOL 5.6 на одном узле раздела *basic* с 8 ядрами, время 12 часо  
submit-comsol-5.6 $HOME/test-comsol/model.mph -A 1234-567-890 -c 8 -L 12:00
```

**Пример 11: GULP

```
# Запуск GULP 5.1 на одном узле раздела *basic* с 4 ядрами, время 6 часов  
submit-gulp-5.1 input.gin -A 1234-567-890 -c 4 -L 6:00:00 -m cluster -q bas
```

Пример 12: Anaconda (Python)

```
# Запуск Python скрипта в окружении 'myenv' на одном узле раздела *basic* с  
submit-anaconda-2023.03 script.py -e myenv -A 1234-567-890 -c 2 -L 2:00:00
```

3.5. Спецификация ресурсов (разделы SLURM)

В вычислительном центре доступны следующие разделы (partitions) для запуска задач:

Основные разделы CPU:

Раздел	Кол-во узлов	Ядер на узел	Всего ядер	Память на ядро (МБ)	Время по умолчанию (часы)
basic	30	24	720	5300	48*
extended1	12	28	336	8900	48
extended2	18	56	1008	8900	48
smp	5	144, 72, 60	472	15000	48

Примечание: раздел **basic** является разделом по умолчанию и имеет максимальное время выполнения 14 дней.

Разделы с GPU:

Раздел	Кол-во узлов	Ядер на узел	Всего ядер	GPU	Память на ядро (МБ)	Время по умолчанию (часы)
P100	9	28	252	1×NVidia Tesla P100 16ГБ	8900	48
A6000	2	56	112	1×NVidia RTX A6000 48ГБ	2053	48
K40	10	24	240	2×NVidia Tesla K40M 12ГБ	5300	48

Особенности оборудования:

- **basic**: 2×Intel Xeon E5-2680 v3 2.50 ГГц (2×12 ядер), 128 ГБ RAM
- **extended1**: 2×Intel Xeon E5-2690 v4 2.60 ГГц (2×14 ядер), 256 ГБ RAM
- **extended2**: 2×Intel Xeon Gold 6258R 2.70 ГГц (2×28 ядер), 512 ГБ RAM
- **P100**: поддержка avx2, NVIDIA Tesla P100
- **A6000**: поддержка avx512, NVIDIA RTX A6000
- **K40**: поддержка avx2, 2×NVIDIA Tesla K40M

Ограничения QoS:

- Максимальное количество одновременно выполняемых задач на один аккаунт: **5**
- Максимальное общее количество задач (выполняемых + в очереди): **7**

Рекомендации:

- Используйте `sinfo` для проверки доступности узлов перед запуском задач
- Указывайте только необходимые features для избежания длинных очередей
- Для GPU-задач убедитесь, что ПО оптимизировано для выбранного графического процессора

4. Устранение неполадок

4.1. Типичные ошибки и пути их решения

Система использует единую систему сообщений об ошибках. Ниже приведены наиболее распространенные ошибки, сгруппированные по категориям:

Ошибки параметров и аргументов

- Ошибка! Было указано неправильное значение аргумента! - проверьте правильность указанных опций
- Ошибка! указан неизвестный параметр! - убедитесь, что имя параметра указано верно
- Ошибка! для данного параметра требуется значение! - укажите значение для параметра

- **Ошибка! Заданы несовместимые параметры!** - проверьте совместимость используемых опций

Ошибки файлов и директорий

- **Ошибка! Указанного файла не существует!** - проверьте путь и имя файла
- **Ошибка! Указанный файл невозможно прочитать!** - проверьте права доступа к файлу
- **Ошибка! Указанная директория не существует!** - проверьте путь к директории
- **Ошибка! Имя файла содержит пробелы!** - используйте имена без пробелов

Ошибки ресурсов и выделения

- **Ошибка! Запрошено слишком много ядер!** - уменьшите количество запрашиваемых ядер
- **Ошибка! Запрошено слишком много GPU!** - уменьшите количество запрашиваемых GPU
- **Ошибка! Не могу подобрать узлы!** - измените параметры ресурсов или попробуйте позже
- **Ошибка! Количество ядер не может быть равно нулю!** - укажите корректное количество ядер

Ошибки модулей и окружения

- **Ошибка! Неправильный модуль!** - проверьте имя загружаемого модуля
- **Ошибка! Неправильный модуль для данного ресурса!** - используйте совместимый модуль
- **Ошибка! Задана переменная, используемая в 'env' файле!** - не переопределяйте системные переменные

Ошибки типа данных и значений

- **Ошибка! Указанное значение не является целочисленным!** - укажите целое число
- **Ошибка! Указанное значение не является положительным целочисленным!** - укажите положительное число
- **Ошибка! Неправильное значение определенного времени или временного интервала!** - проверьте формат времени

Критические ошибки системы

- Ошибка! Не удастся запустить задачу! - общая ошибка отправки задачи
- Ошибка! Внутренняя ошибка! - обратитесь в службу поддержки
- Ошибка! Ошибка при работе с файлом описания библиотеки! - проблема с системными файлами

Предупреждения системы

- Внимание! Количество запрашиваемых ядер было уменьшено! - система автоматически скорректировала ресурсы
- Внимание! Был выбран неоптимальный для данного ресурса вариант! - используется резервный вариант запуска
- Внимание! Используется модуль! - информирование о загруженном модуле

Рекомендации по устранению ошибок:

1. **Проверьте синтаксис команды** - используйте `-h` для просмотра справки по параметрам
2. **Убедитесь в доступности файлов** - проверьте пути и права доступа
3. **Проверьте доступность ресурсов** - используйте `sinfo` для просмотра доступных узлов
4. **Следуйте формату значений** - особенно для времени и числовых параметров
5. **При повторяющихся ошибках** - используйте повышенный уровень детализации `-v 2` для диагностики

Примечание: Для получения детальной информации об ошибке используйте параметр `-v 2` (повышенная детализация). Если ошибка повторяется, сохраните полный вывод и обратитесь в службу поддержки.

4.2. Проверка статуса задач

- Для проверки статуса ваших задач можно использовать команды менеджера задач, установленного в системе (например, `queue` для SLURM).

5. Рекомендации

5.1. Лучшие практики

- **Использование модулей:** Рекомендуется использовать `module load <имя_модуля>` для настройки окружения.
- **Правильное указание опций:** Указывайте все необходимые опции, включая `-A` (номер исследования), `-m` (ресурс), `-q` (очередь), `-c` (количество ядер) и `-L` (время выполнения), в соответствии с требованиями вашей задачи.
- **Абсолютные и относительные пути:** Для указания входных файлов и директорий рекомендуется использовать абсолютные пути (`/home/username/path/to/file`) или относительные пути (`./file.com`, `../input/`), чтобы избежать путаницы.
- **Проверка перед запуском:** Перед отправкой задачи убедитесь, что все необходимые файлы находятся на месте и доступны.
- **Учет ресурсов:** Указывайте реалистичные значения для количества ядер, памяти и времени выполнения, чтобы не блокировать систему дольше, чем необходимо. Это повысит приоритет вашей задачи в очереди.
- **Мониторинг задач:** Периодически проверяйте статус своих задач, чтобы своевременно реагировать на сбои или отмену.

5.2. Примеры запуска (кратко)

- **Общая структура:** `<скрипт_запуска> <входной_файл_или_директория> [опции...]`
- **Gaussian:** `submit-gaussian-16 input.com -A 1234-567-890 -c 8 -L 24:00:00 -m cluster -q basic`
- **Crystal:** `submit-crystal-14.1.0.4 input.d12 -A 1234-567-890 -c 8 -L 24:00:00 -m cluster -q basic`
- **CP2K:** `submit-cp2k-8.2.0 input.inp -A 1234-567-890 -c 16 -L 12:00:00 -m cluster -q basic`
- **GROMACS:** `submit-gromacs-2019 $HOME/test/md -A 1234-567-890 -n 4 -ppn 16 -gn 1 -L 24:00:00 -m cluster -q p100`
- **Другие ПО:** Структура запуска аналогична для большинства пакетов. Ознакомьтесь с документацией конкретного ПО или используйте `<скрипт_запуска> -h` для получения справки.