

0.1 Методика

- 1).Изучение пакета и выбор необходимых компонентов в зависимости от конфигурации кластера
- 2).Скачать архив cp2k-X.Y.tar.bz2 с Git репозитория

Ссылка для скачивания: <https://github.com/cp2k/cp2k/tree/master>

- 3).Провести установку CP2K с необходимыми дополнениями(поддержка MPI,GNU,AVX512,LIBINT,Fortran,LIBXSMM,CUDA,LIBXC,spglib)
- 4).Компиляция.Необходимо указать местоположение компилятора и библиотек. Примеры для нескольких распространенных архитектур можно найти в папке arch. Условно существует шесть версий:
 - sdbg - OpenMP + debug settings,
 - sopt - OpenMP + OMP-NUM-THREADS=1,
 - ssmp - OpenMP,
 - pdbg - MPI + OpenMP + debug settings,
 - popt - MPI + OpenMP + OMP-NUM-THREADS=1,
 - psmp - MPI + OpenMP.

Нужно будет изменить один из этих файлов в соответствии с настройками системы.Теперь можно собрать CP2K, используя эти настройки (где -j N допускает параллельную сборку с использованием N процессов): `make -j N ARCH=архитектура VERSION=версия`. Например:

```
make -j 4 ARCH=Linux-x86-64-gfortran VERSION=psmp
```

- 5).Установка необходимых флагов при компиляции. Их можно посмотреть на github'е с установкой cp2k.
- 6).Скачать бенчмарк и запустить тестовую задачу

Ссылка для скачивания: <https://github.com/cp2k/cp2k/tree/master/benchmarks/QS/>

- 7).Запуск бенчмарка на различном числе MPI потоков, изменять их количество по степени 2,а также запуск с GPU.

Это можно сделать с помощью sbatch-скрипта (указав ntasks,cpus-per-task,gpus, и т.д)

- #SBATCH –job-name=task - Название задачи
 - #SBATCH –time=12:00:00 - Максимальное время выполнения
 - #SBATCH –ntasks=N - Количество MPI процессов
 - #SBATCH –nodes=N - Требуемое кол-во узлов
 - #SBATCH –gpus=N - Требуемое кол-во GPU
 - #SBATCH –cpus-per-task=N - Требуемое кол-во CPU
- 8). Анализ полученных данных, просмотр результатов профилирования nvprof или . Нормирование мощности.