

0.1 Методика исследования

В данном параграфе описана методика исследования linux HPC-систем с помощью бенчмарка Lid driven cavity 3D OpenFOAM. В рамках работы над проектом необходимо осветить базовые элементы установки, сборки и компиляции вычислительного пакета, которое предшествует исследованию эффективности работы архитектур. После чего будут освещены особенности работы с конфигурациями эталонной задачей и подходы получения данных.

Прежде всего следует обратить внимание на то, что в работе рассматривается официальный вариант пакета OpenFOAM [1]. Конфигурация этого программного обеспечения не поддерживает работу с GPU и для вычислений используется только MPI формат распараллеливания. Если для исследователя необходимо проведение расчётов с помощью видеокарт и гибридного MPI-OpenMP распараллеливание, нетипичных решателей и предикторов, необходимо использовать сторонние надстройки для пакета.

На первом этапе рассматриваемой методики, конечно, необходимо необходимо установить вычислительный пакет OpenFOAM с официального сайта проекта. Для этого:

1) Установить необходимые пакеты для компиляции:

```
sudo apt-get install build-essential cmake git ca-certificates
```

2) Установить общие пакеты для OpenFOAM:

```
sudo apt-get install flex libfl-dev bison zlib1g-dev  
libboost-system-dev libboost-thread-dev libopenmpi-dev  
openmpi-bin gnuplot libreadline-dev libncurses-dev libxt-dev
```

3) Клонировать репозитории, заменив dev на желаемую версию (нп.10):

```
git clone https://github.com/OpenFOAM/OpenFOAM-dev.git  
git clone https://github.com/OpenFOAM/ThirdParty-dev.git
```

4) Прописать переменную окружения в ./bashrc файле:

```
source $HOME/OpenFOAM/OpenFOAM-dev/etc/bashrc
```

5) После в консоле написать команду:

```
source $HOME/.bashrc
```

6) Перейти в ThirdParty-dev для сборки файлов директории:

```
./Allwmake
```

7) Перейти в OpenFOAM-dev для сборки след. файлов директории:

```
./Allwmake
```

После установки пакета необходимо удостовериться, что все сделано правильно и запустить тестовую задачу:

1) Создать папку, в которой будут находиться задачи для запуска:

```
mkdir -p $FOAM_RUN
```

2) Скопировать конфигурации обучающей задачи и запустите:

```
cd $FOAM_RUN
```

```
cp -r $FOAM_TUTORIALS/incompressible/simpleFoam/pitzDaily .
```

```
cd pitzDaily
```

```
blockMesh
```

```
simpleFoam
```

```
paraFoam
```

Обратим внимание на то, что при выполнении моделирования тестовой задачи, необходимо перейти в папку с файлами конфигурации, где хранятся начальные и граничные условия, информация о решетке и прочему, вызвать предиктор `blockMesh`, а после решатель `simpleFoam`. При успешном выполнении задачи, считается, что установка и сборка OpenFOAM завершена. Теперь необходимо скачать конфигурации бенчмарка в созданную `run`-папку с сайта.

Выполнив необходимые манипуляции и получив нужную конфигурацию рассматриваемого бенчмарка, можно приступать к следующему шагу исследования производительности: тестирование системы с помощью дополнительного инструментария. Необходимо измерить скорости передачи данных между оперативной памятью компьютера и процессором с помощью синтетического бенчмарка `STREAM`. Если бенчмарк-задача ориентирована на работу с памятью, то полученная информация поможет наиболее эффективно интерпретировать результаты.

Также необходимо получить сведения о поддерживаемой многопоточности вычислительного устройства в рамках одного ядра с помощью команды `scontrol show node`. При выполнении этой команды на узлах суперкомпьютера ВШЭ Charizma, выяснилось, что на вычислительном кластере отключена многопоточность на уровне ядра процессора, поэтому полученные результаты в рамках одного ядра можно интерпретировать как однопоточные.

Следующим этапом методики исследования высокопроизводительных систем является непосредственно получение результатов работы выбранного бенчмарка. Запуск программ необходимо выполнять в пакетном режиме с помощью `bash`-скриптов и системы `SLURM`. На сегодняшний день, научным сообщество ценится данная возможность

и, как следствие, большинство вычислительных кластеров организованы подобным образом.

Перейдем к особенностям запуска. Для получения первичной информации о производительности вычислительного устройства с помощью бенчмарка Lid driven cavity 3D, необходимо провести серию запусков эталонных задач с использованием параллелизма на уровне ядер. В рамках рассмотрения суперкомпьютера Харизмы и постоянного ажиотажа вычислительных ресурсов, хорошим результатом является получение усредненных исследуемых характеристик процессорного (задействованы все ядра процессора), узлового(задействованы все ядра узла) и однопоточного (последовательный вариант бенчмарка) выполнения задачи.

После произведения расчетов, необходимо рассмотреть различные срезы полученных данных, а именно : нормировку на энергопотребление, нормировку на усредненную частоту, цену и пиковую производительность вычислительного устройства. Этот анализ будет очень полезными и может значительно помочь ответить на вопросы экономичности и эффективности производимых в дальнейшем вычислений.

В силу специфики и гибкости масштаба выбранного бенчмарка, исследователь, проводя работы с помощью Lid driven cavity 3D, может построить график эффективности масштабируемости вычислительных ресурсов в зависимости от размера задачи. Это исследование является необходимым для понимания оптимального количества запрашиваемых мощностей для эффективного межъядерного баланса и возможного ускорения.

При проведении исследования производительности суперкомпьютерных архитектур, необходимо помнить, что Lid driven cavity 3D является реальной задачей, которая нацелена на получение физического результата. Поэтому вместе с рассмотренным бенчмарком важно использовать сопутствующие приложения для профилирования, которые осуществляют мониторинг системы во время выполнения эталонной задачи. В рамках данной методики использовался профилировщик linux систем perf, основная информация о котором предоставлена в предыдущей главе.

Итак, рассмотрим типичный подход к запуску бенчмарка Lid driven cavity 3D. Скачанный файл с конфигурацией содержит папки с необходимыми масштабами, в которых находятся вариации бенчмарка, а именно папки *fixedIter* и *fixedTol*. Перед началом запуска задачи, необходимо проделать некоторую предварительную подготовку задачи:

1) Перейдем в папку, предназначенную непосредственно для бенчмаркинга:

```
cd $FOAM_RUN/cavity3D/8M/fixeIter
```

2) Откроем репозиторий с файлами конфигурации:

```
cd system
```

3) Любым редактором открываем файл `decomposeParDict` и задаем количество субдоменов для разбиения сетки, при этом необходимо учитывать, что это значение должно быть строго равно количеству `mpi` процессов, установленных для выполнения задачи:

```
numberOfSubdomains 4;
```

Проделав эту процедуру, мы передали параметры системной функции `decomposeParDict` для деления сетки на области, каждая из которых будет передана отдельному `mpi` процессу. Наконец, для запуска задачи на вычислительном узле необходимо написать следующий `bash`-скрипт, указанный на рисунке 0.1.

```
#!/bin/bash
#SBATCH --job-name=Cavity_parallel_8M
#SBATCH --error=Cavity_parallel_8M-%j.err
#SBATCH --output=Cavity_parallel_8M-%j.log
#SBATCH --constraint=<type_node>
#SBATCH --time=23:00:00
#SBATCH --cpus-per-task=1      # Количество ядер CPU, выделенных для одного процесса
#SBATCH --nodes=1             # Количество используемых узлов

#SBATCH --ntasks=4           # Количество MPI процессов
#SBATCH --ntasks-per-node=4   # Количество процессов на один узел
#SBATCH --ntasks-per-socket=22 # Количество процессов на один сокет

cd /home/user/OpenFOAM/run/cavity3D_parallel/8M/fixeIter
blockMesh
decomposePar
perf stat mpirun -np 4 isoFoam -parallel # параллельный запуск бенчмарка с 4 ядрами и
#стандартной командой профилирования perf
```

Рис. 0.1: Пример `bash` скрипта запуска

Важным аспектом представленного скрипта является то, что для выполнения вычислений `sbatch` - задача резервирует все ядра процессора и не дает другим задачам `SLURM` - очереди занимать кэш-память процессора и ОЗУ. При таком подходе исклю-

чается вмешательство сторонних программ в анализ производительности процессора (сокета). В случае, если необходимо запустить задачу на всех ядра узла, данный параметр указывать не нужно.

С помощью представленной методики, исследователь способен получать время выполнения задачи на запрошенных вычислительных ресурсах без подмешивания посторонних задач, информацию о средней частоте работы ядра во время работы бенчмарка, информацию о количестве промахов в ветвлениях, общее количество инструкций, выполненных процессором и некоторые другие выводы о запуске задачи, которые используются для оценки эффективности работы сри. Для более полной картины о запуске, рекомендуется использовать все ключи `perf`, указанные в предыдущем параграфе.

Итоговой структурой методики оценивания производительности высокопроизводительных систем является:

- Установка вычислительного пакета OpenFOAM.
- Интеграция выбранного бенчмарка Lid driven cavity 3D в пакет.
- Подготовка конфигурации задачи в зависимости от проводимого исследования.
- Проведение первичных исследований рассматриваемых НРС-систем с помощью сторонних приложений .
- Запуск бенчмарка с использованием инструментов профилирования.
- Анализ эффективности масштабируемости системы, анализ оптимального ускорения.
- Рассмотрение результатов в различных срезах: нормирование на стоимость вычислительного устройства, пиковую производительность, энергопотребление.

Список использованных источников

[1] OpenFOAM User Guide OpenFOAM. *The OpenFOAM Foundation*. 2014.