Отображение DVMH-программ на кластеры с ускорителями

ИПМ им. М.В.Келдыша РАН
Притула М.Н.

Содержание

- Направление развития архитектуры кластеров
- Модель программирования DVMH
- Схема выполнения DVMH-программы
- Режимы выполнения и динамическое планирование
- Возможности сравнительной отладки

Кластерные архитектуры

- 62 из Top500 имеют ускорители:
  - 50 – NVIDIA
  - 7 – Intel
  - 3 – AMD/ATI
  - 2 – IBM
- Растущее количество ядер центральных процессоров на общей памяти
- Существенная разнородность внутри узла

Способы программирования

- MPI/SHMEM + OpenMP/Pthreads + CUDA/OpenCL
- MPI/SHMEM + OpenACC/HMPP/PGI Accelerator
- MPI/SHMEM + OpenMP 4.0
- DVMH

Модель DVMH

- Расширение DVM-модели
- Параллелизм по данным и параллелизм задач
- Директивы
- Fortran & C
  - !DVM$ <directive> [<clause> {, <clause>}]
  - #pragma dvm <directive> [<clause> {, <clause>}]
- Ориентация на кластеры с разнородными ускорителями

Программа Якоби на Fortran

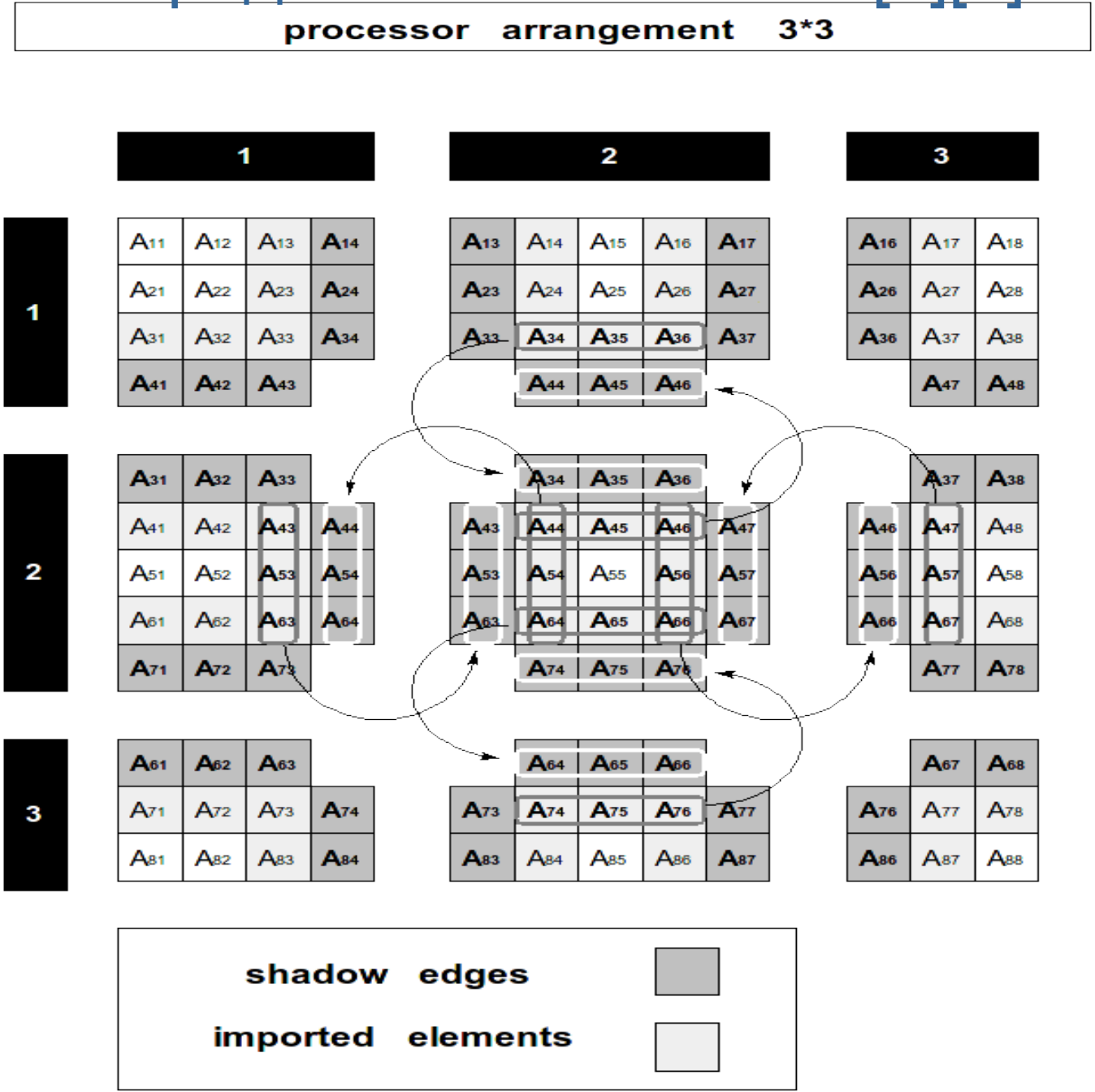
```
      PROGRAM    JAC_F77
      PARAMETER    (L=8,  ITMAX=20)
      REAL     A(L,L), B(L,L)

      PRINT *,  '**********  TEST_JACOBI   **********'

      DO IT  =  1,  ITMAX
        DO  J  =  2, L-1
           DO  I  =  2, L-1
                A(I, J)  =  B(I, J)
           ENDDO
        ENDDO
        DO  J = 2,  L-1
           DO  I = 2,  L-1
                   B(I, J) =  (A(I-1, J) + A(I, J-1) +
     *                         A(I+1, J) + A(I, J+1)) / 4
           ENDDO
        ENDDO
      ENDDO
      END
```

```
      PROGRAM   JAC_DVM
      PARAMETER   (L=8,  ITMAX=20)
      REAL    A(L,L), B(L,L)
CDVM$ DISTRIBUTE   ( BLOCK,  BLOCK)  ::  A
CDVM$ ALIGN  B(I,J)  WITH  A(I,J)
C     arrays A and B  with block distribution
      PRINT *,  '**********  TEST_JACOBI  **********'
      DO IT  =  1,  ITMAX
CDVM$   PARALLEL  (J,  I)  ON  A(I,  J)
        DO  J  =  2, L-1
          DO  I  =  2, L-1
            A(I, J)  =  B(I, J)
          ENDDO
        ENDDO
CDVM$   PARALLEL  (J,  I) ON B(I,  J),SHADOW_RENEW (A)
C       Copying shadow elements of array A from
C       neighboring processors before loop execution
        DO  J = 2,  L-1
          DO  I = 2,  L-1
            B(I, J) =  (A( I-1, J ) + A( I, J-1 ) +
     *                                A( I+1, J ) + A( I, J+1 )) / 4
          ENDDO
        ENDDO
      ENDDO
      END
```

Распределение массива A[8][8]



```
      PROGRAM   JAC_DVMH
      PARAMETER   (L=4096,  ITMAX=20)
      REAL    A(L,L), B(L,L)
CDVM$  DISTRIBUTE    ( BLOCK,   BLOCK)   ::   A
CDVM$  ALIGN  B(I,J)  WITH  A(I,J)
C      arrays A and B  with block distribution
      PRINT *,  '**********  TEST_JACOBI   **********'
      DO IT  =  1,  ITMAX
CDVM$   REGION
CDVM$     PARALLEL  (J,  I)   ON  A(I,  J)
          DO  J  =  2, L-1
              DO  I  =  2, L-1
                 A(I, J)  =  B(I, J)
              ENDDO
          ENDDO
CDVM$     PARALLEL  (J,  I) ON B(I,  J),SHADOW_RENEW (A)
          DO  J = 2,  L-1
              DO  I = 2,  L-1
                  B(I, J) =  (A( I-1, J ) + A( I, J-1 ) +
     *                        A( I+1, J ) + A( I, J+1 )) / 4
              ENDDO
          ENDDO
CDVM$   END REGION
      ENDDO
CDVM$  GET_ACTUAL(B)
      PRINT *,B
      END
```

Модель DVMH

- Уровни параллелизма
  - Распределение данных и вычислений по MPI процессам
  - Распределение данных и вычислений по вычислительным устройствам
  - Параллельная обработка на вычислительном устройстве

  Позволяют органично отображать программы на кластер с многоядерными процессорами и ускорителями в узлах

Схема выполнения DVMH-программ

- Начальное распределение данных
- При входе в вычислительный регион – дополнительное распределение по устройствам, обеспечение актуальных данных на устройствах
- При параллельной обработке цикла на устройстве – выбор обработчика, подбор значений оптимизационных параметров

Режимы выполнения DVMH-программ

- Простой статический режим
  - Задается вектором производительностей
  - Распределение одинаково для каждого региона
  - Минимальные перемещения при перераспределении
- Динамический режим с подбором схемы распределения
- Статический режим с использованием подобранной схемы распределения
  - Переход из 2го режима / использование файла
  - Проблемы при изменении программы

Динамический режим

- Сбор информации на разных уровнях:
  - Параллельный цикл, последовательный участок, хостсекция
  - Экземпляр выполнения региона
  - Регион + соответствие данных
  - Регион
  - Программа
- Динамическое построение графа зависимостей по данным с учетом возможности дублирования актуальных данных

…

Динамический режим

- Периодическое построение глобальной схемы распределения
- Возможность записи построенной схемы в файл с детализацией до “Регион + соответствие данных”
- Возможность перехода в третий режим “на лету”

Настройка обработчиков

- ЦПУ обработчик
  - Количество параллельных нитей
  - Метод планирования расписания
- CUDA обработчики
  - Несколько штук – разные методы оптимизации / опции компиляции
  - Размеры блока нитей
  - Способ обработки редукционных операций

Сравнительная отладка

- Специальный режим работы региона
- Дублирующее одновременное выполнение на ЦПУ и ускорителях
- По завершению региона – сравнение значений выходных переменных
- Коррекция найденных ошибок “на лету” и продолжение вычислений

Заключение

- Языки модели DVMH обеспечивают высокий уровень переносимости прикладного ПО
- DVMH-программы обладают способностью гибко подстраиваться под конкретную вычислительную аппаратуру, на которой эти программы запущены
- Для получения последней версии обращайтесь на адрес dvm@keldysh.ru