Летняя школа-конференция по параллельному программированию 2017

Разработка библиотеки для параллельного программирования на основе распределенного N-мерного массива

Выполнил: Ижицкий Р. Л.
Руководитель: Городничев М. А.
14.07.2017

Научная проблема

- Повторное использование системного кода
  - ```
    for (int i = 0; i < size; ++i)
    {
        sendCounts[i] = N / size;
        if (rem > 0)
        {
            sendCounts[i]++;
            rem--;
        }
        displs[i] = sumDispls;
        sumDispls = sumDispls + sendCounts[i];
    }
    ```
- Повышение сложности алгоритмов
- Высокий порог вхождения в область параллельного программирования
    ```

Альтернативные решения

- DVM
  - ```
    #pragma dvm array distribute[block][block][block][*]
     double ARRAY[Z][Y][X][5];

    #pragma dvm parallel([I][J][K] on ARRAY[I][J][K][*])
     for(int I = 1; i < Z; ++I)
       for(int J = 1; J < Y; ++J)
         for(int K = 1; K < X; ++K)
           ARRAY[K][J][I][4] = Func(ARRAY[K][J][I][2], ARRAY[K][J][I][5]);
    ```
- HOPMA
- KeLP
- MATLAB (Distributed Arrays)
    ```

Цель работы

- Разработка библиотеки для управления распределенным N-мерным массивом

Требования

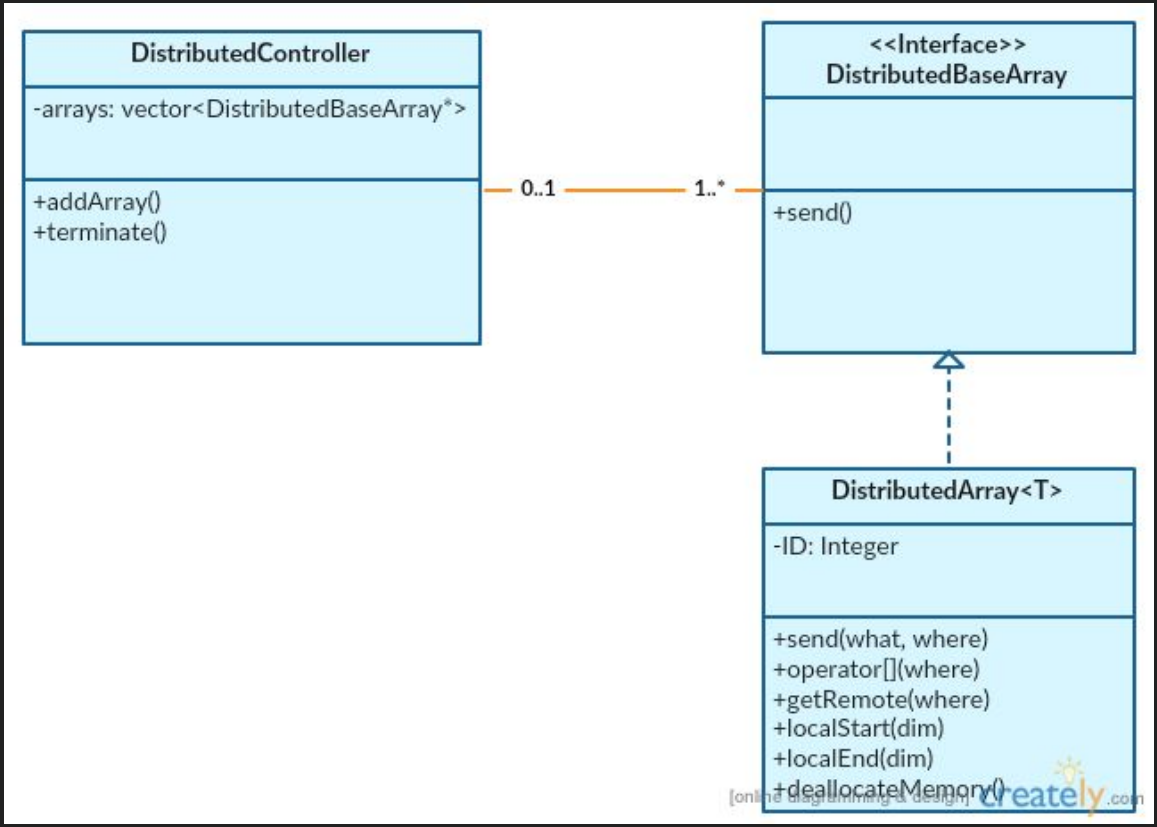
- Удобный и логичный для пользователя интерфейс
- Возможность получения данных с удаленных узлов

main()

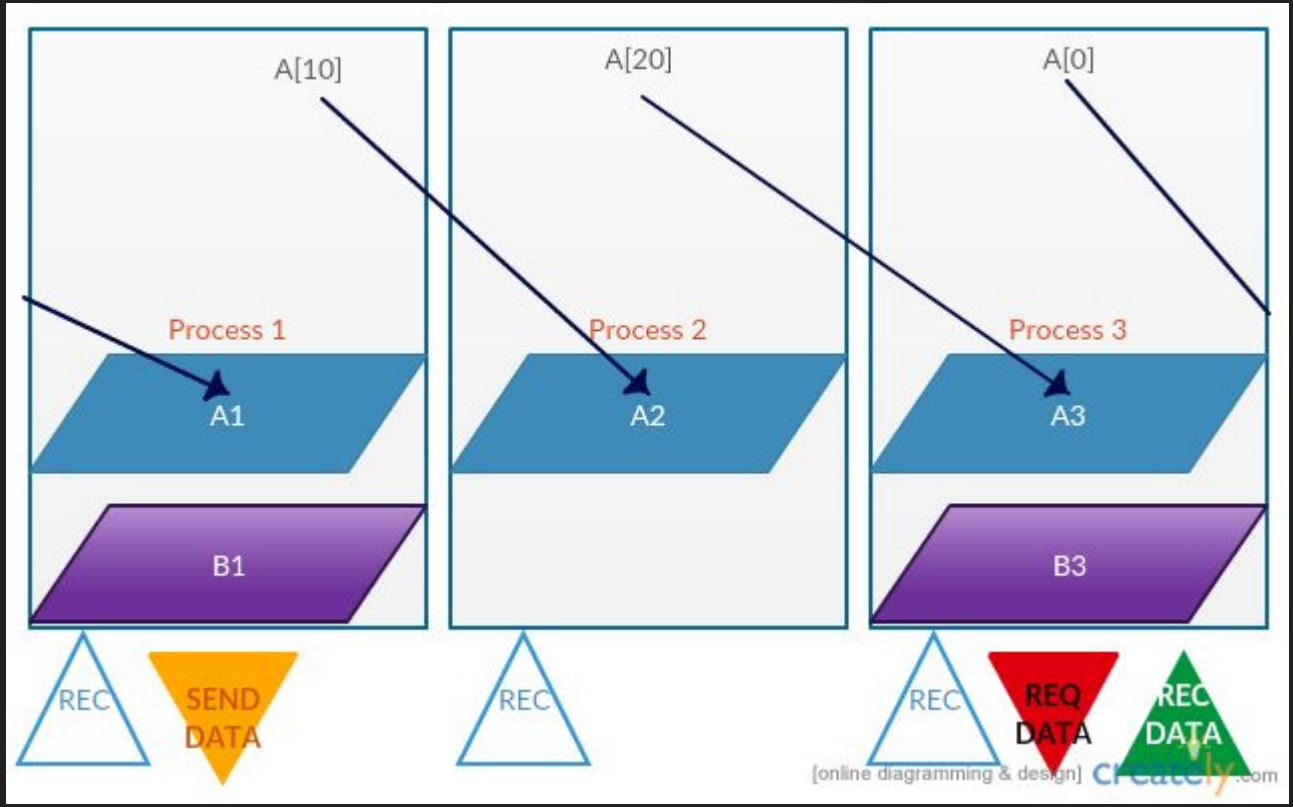
```
int rank, size;
MPI_Comm_rank(MPI_COMM_WORLD, &rank);
MPI_Comm_size(MPI_COMM_WORLD, &size);
DistributedController cntrl(rank);
DistributedArray<int> a = DistributedArray<int>(cntrl, {7, 2}, {4, 0}, {rank, 0});
for (unsigned int i = a.localStartInDim(0); i < a.localEndInDim(0); i++)
{
    a[{i}] = i;
    a[{i}] = a[{i}] * 2;
}

a.deallocate();
//DistributedArray<int> b = DistributedArray<int>(cntrl, {7, 5}, {2, 2}, {rank / 2, rank % 2});
cntrl.terminate();
MPI_Finalize();
```

Реализационные детали



Реализационные детали



Заключение

- Разработан прототип библиотеки:
  - Распределенный массив
  - Удобное обращение

- Протестировано:
  - Двумерные и одномерные массивы

Планы

- Дробление равномерного участка в памяти на фрагменты
  - ```
    while (fr = arrA.localFragments.getFragment())
        for (i = fr.localStart(); i < fr.localEnd(); i++)
            fr[i] = foo(i)
    ```
- Обмен фрагментов
- Сравнение с альтернативными решениями
- Тестирование
    ```

Спасибо за внимание