Использование гибридной технологии CUDA-MPI в методе молекулярной динамики.

Уткин А.В., Фомин В.М.
Институт Теоретической и Прикладной Механики СО РАН им. С.А. Христиановича

Ожгибесов М.С.
Nanyang Technological University, Singapore

2. Философия гибридного кода и немного статистики.

Каждый GPU вычисляет независимую задачу.

Неэффективная загрузка системы!

Проводя аналогию – расчет серийного кода на параллельной станции.

Каждый GPU вычисляет часть всей задачи.

Затем используем MPI

для сбора частей в единую задачу.

MPI может собирать информацию как внутри одного узла , так и с разных узлов!

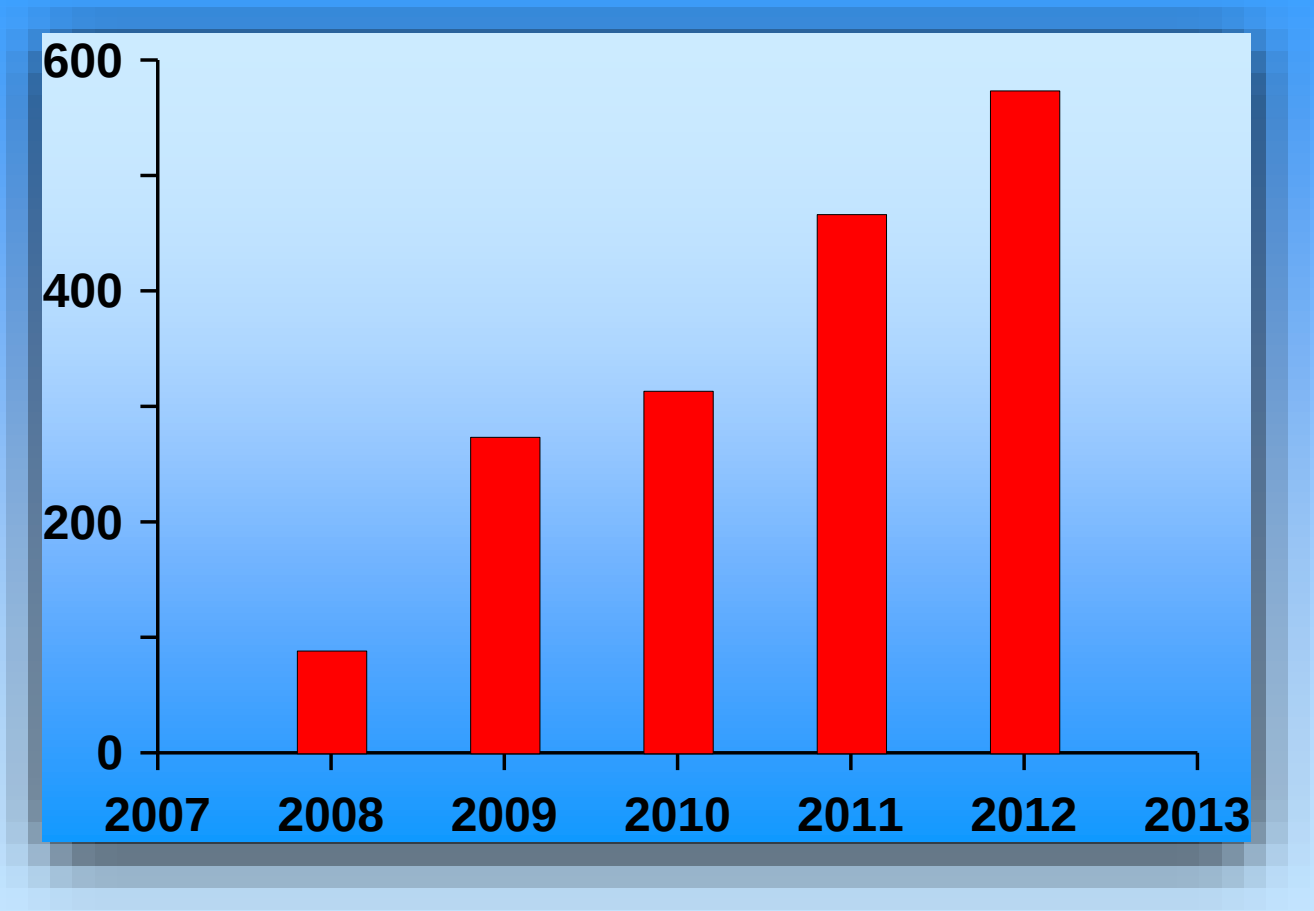
Или используем OpenMP

для сбора частей в единую задачу.

OpenMP может собирать информацию только внутри одного узла!

или испльзуем MPI для обмена между узлами, а OpenMP для обменов внутри узла!

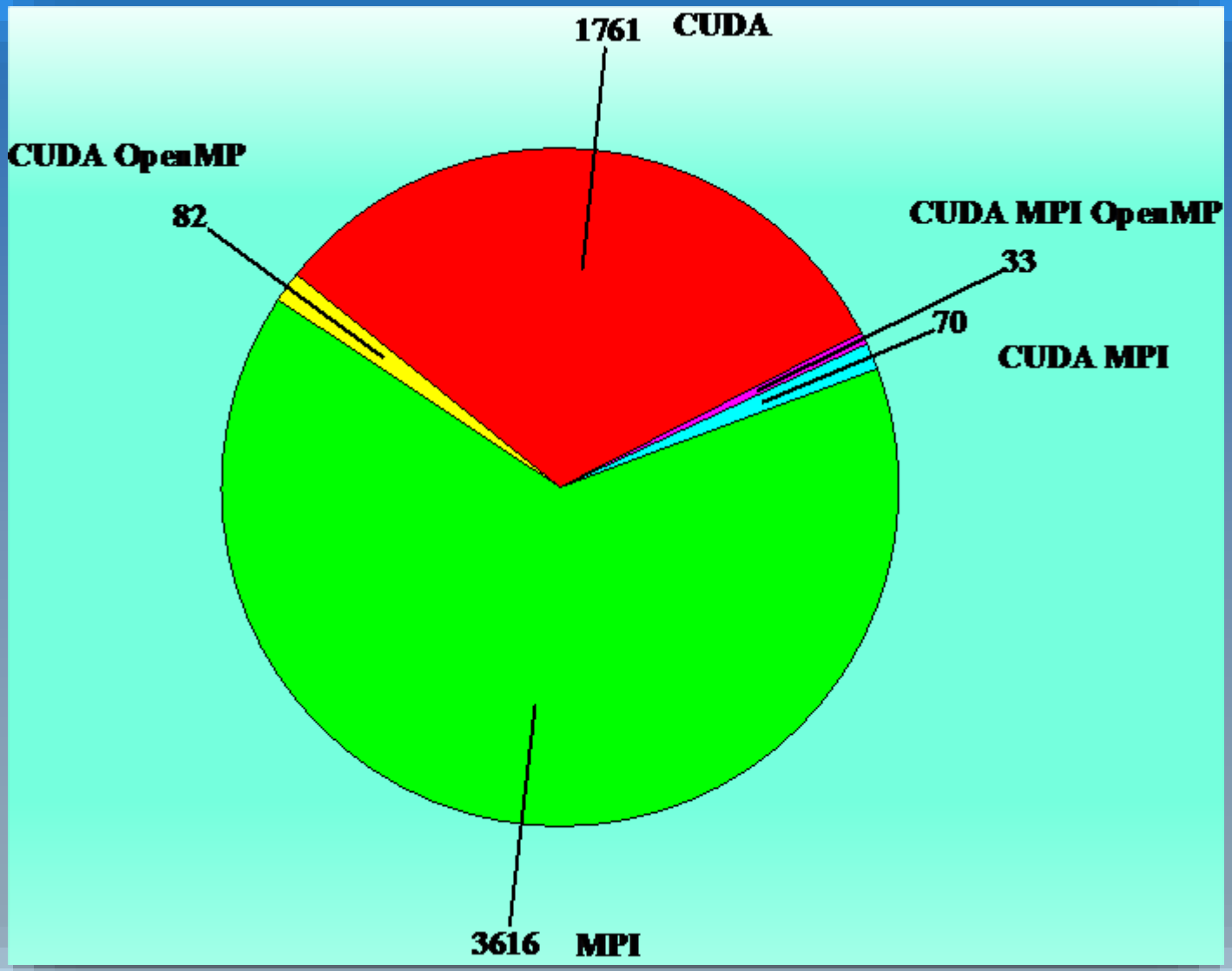
Число публикаций содержащих слово “CUDA” в аннотации*



Годы публикаций: 2008~2012

*Web of Science® search results.

Число публикаций связанных с параллельными расчетами*

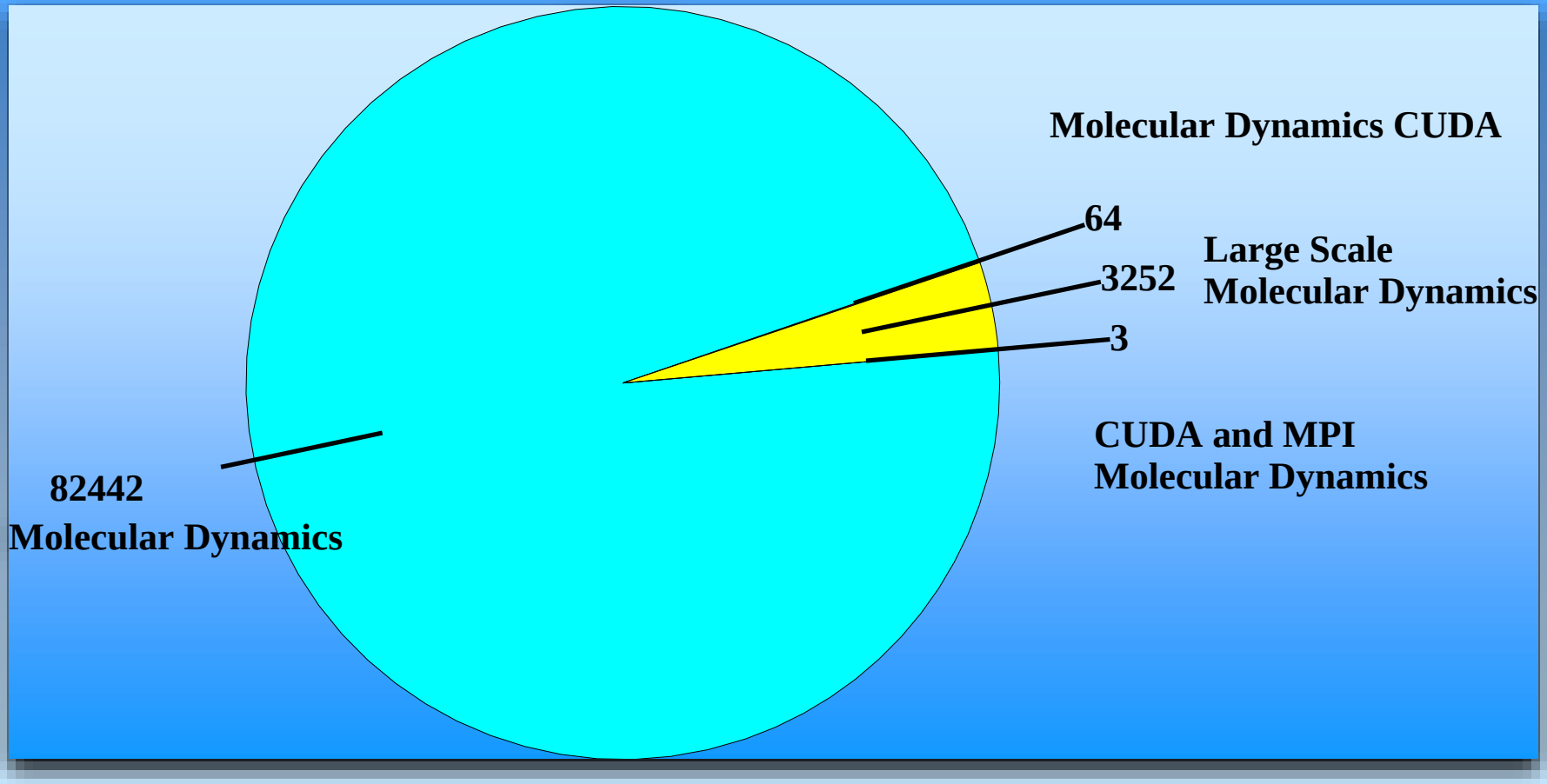


Годы публикаций: 2008~2013

*Web of Science® search results.

Число публикаций связанных с методом молекулярной динамики (MD simulations)*

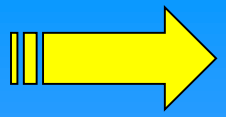
Годы публикаций: 2008~2013



*Web of Science® search results.

SERIAL CPU EXAMPLE

```
real(8):: A(n),B(n),C(n)
do i=1,n
C(i)=A(i)+B(i)
end do
end
```



CUDA GPU EXAMPLE

```
real(8):: A(n),B(n),C(n)
real(8), device, allocatable:: AD(:),BD(:),CD(:)
istat=cudaMalloc(AD,n)
istat=cudaMalloc(BD,n)
istat=cudaMalloc(CD,n)
istat=cudaMemcpy(AD,A,n,cudaMemcpyHostToDevice)
istat=cudaMemcpy(BD,B,n,cudaMemcpyHostToDevice)
istat=cudaMemset(CD,0,n)
   call Addition<<<n/64,64>>>(n,AD,BD,CD)
istat=cudaMemcpy(C,CD,n,cudaMemcpyDeviceToHost)
istat=cudaFree(AD)
istat=cudaFree(BD)
istat=cudaFree(CD)
end
c==============Kernel Subroutine============
attributes(global) subroutine Addition(n,A,B,C)
real(8), device :: A(n),B(n),C(n)
integer, value :: n
integer :: i
i = (blockidx%x-1)*64+ threadidx%x
if (i.le.n) then
C(i)=A(i)+B(i)
end if
end subroutine Addition
```

Выделяем память на GPU.

Копируем данные из памяти CPU в выделенную память на GPU.

Запуск ядра.

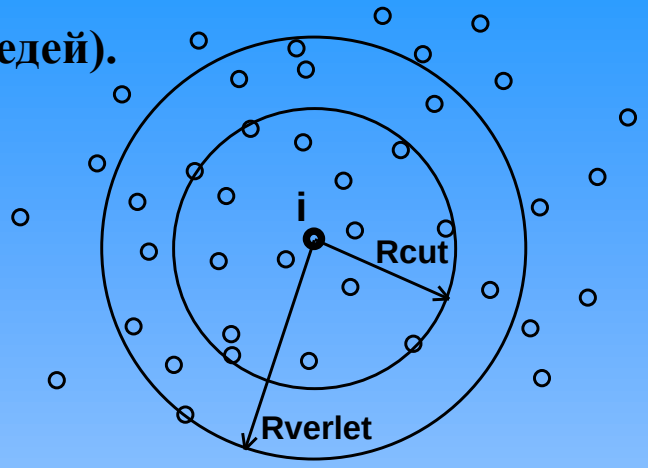
Копируем результаты расчетов обратно в память CPU.

Освобождаем выделенную память GPU.

Для каждого допустимого индекса массива запускается отдельная нить для выполнения нужных вычислений.

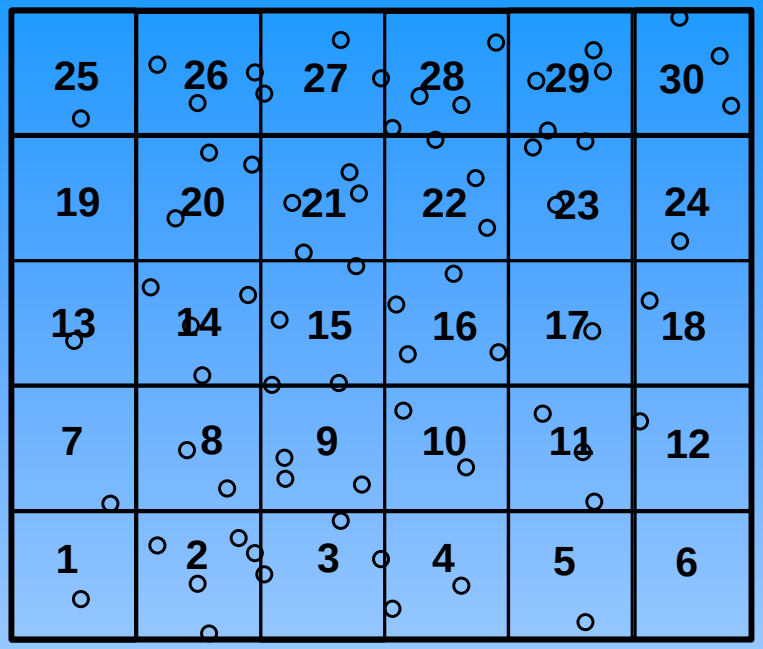
3. Общие методики оптимизации расчетов в методе молекулярной динамики

1. Список Верле (список соседей).



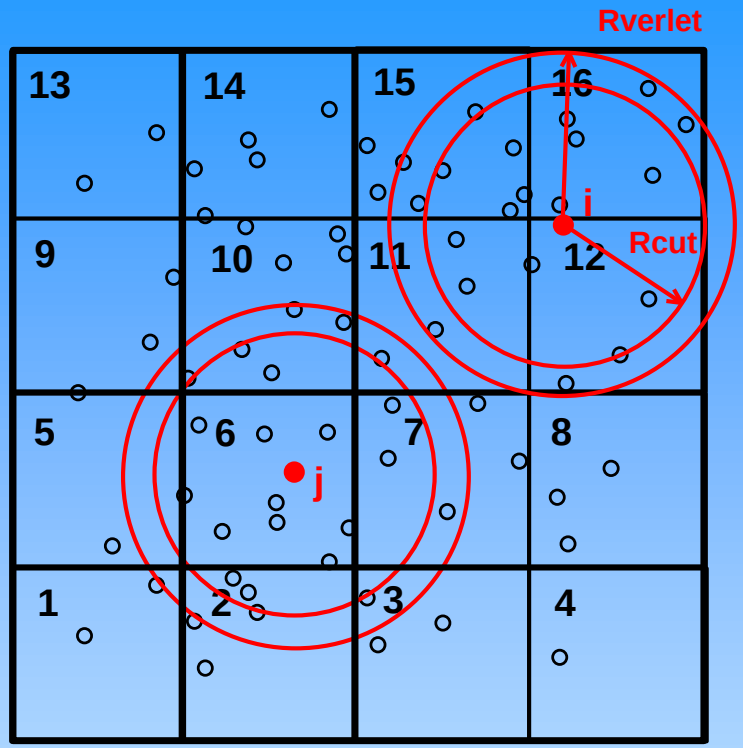
- **Вводится дополнительный радиус обрезания Rverlet > Rcut**
- **Для каждого атома i, создается свой список соседних атомов, попавших в сферу с радиусом Rverlet.**
- **При расчете сил, действующих на атом, i учитываются только соседние атомы из этой сферы.**
- **Интервал между созданием нового списка соседей обычно составляет 10-20 временных шагов или определяется автоматически (если максимальное смещение хотя бы одного атома системы больше чем разница (Rverlet – Rcut) - необходимо обновить список).**
- **Метод достаточно эффективен только при относительно небольшом числе атомов в системе.**
- **Основная проблема - необходимостью перебора всего набора атомов системы при построении списка Верле.**

2. Метод связанных списков (Cell linked list method).



- Трехмерная расчетная область делится на пронумерованные ячейки, чей размер немного больше или равен радиусу обрезания потенциала межатомного взаимодействия
- В начале вычислений создается массив, содержащий список номеров соседей каждой ячейки. Каждый атом взаимодействует только с атомами своей ячейки или с атомами соседних ячеек (26 ячеек).
- Сортировка атомов по ячейкам является быстрой операцией и может выполняться на каждом временном шаге.
- Можно использовать третий закон Ньютона при расчете сил Fij= -Fji. Это позволяет избежать двойного расчета силы в паре атомов i-j, и уменьшает число соседних ячеек с 26 до 13, что приводит к существенному уменьшению времени счета.

3. Различные гибридные комбинации метода связанных списков и списка Верле (гибридный список Верле).



➢**Самой очевидной гибридной комбинацией является метод, где список Верле для атома i создается не на основе перебора всех атомов системы, а на основе анализа расстояния до атомов в ячейке, которой принадлежит атом i и расстояния до атомов в соседних 26 ячейках.**

4. Параллельные алгоритмы метода молекулярной динамики в среде MPI

1. Разбиение системы по частицам; атомы системы равномерно делятся между расчетными процессорами.

Достоинства:
Легкость программирования;
Равномерная загрузка процессора на каждом временном шаге.

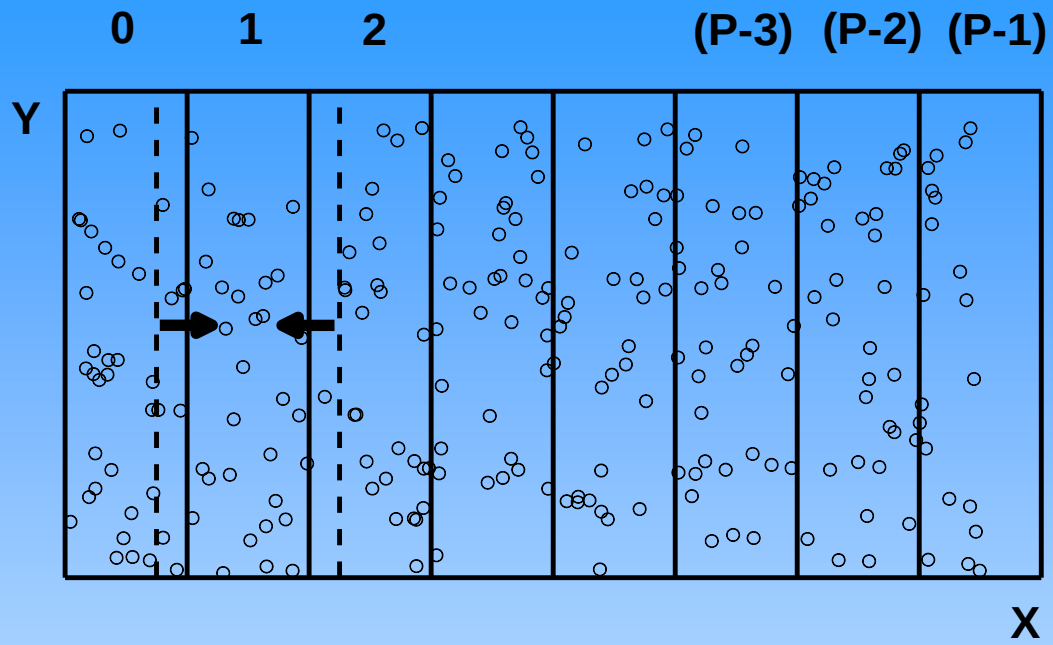
Недостатки.
Атомы, «приписанные» к одному и тому же процессору, могут не взаимодействовать друг с другом.

2. Разбиение системы по пространству; для всей системы атомов расчетная область разбивается на подобласти одинакового размера по числу процессоров.

Достоинства:
Частицы «приписанные» к данному процессору взаимодействуют друг с другом, а обмены между процессорами обусловлены лишь необходимостью пересылки информации из геометрически соседних граничных областей.

Недостатки.
Сложность программирования. Необходимость создания дополнительного алгоритма динамической балансировки подобластей для исследования пространственно-неоднородных систем .

Разбиение системы по пространству. Реализация масштабируемого алгоритма, основанного на одномерной параллелизации с динамической балансировкой.



Схематическое изображение трехмерной расчетной области в плоскости XY.

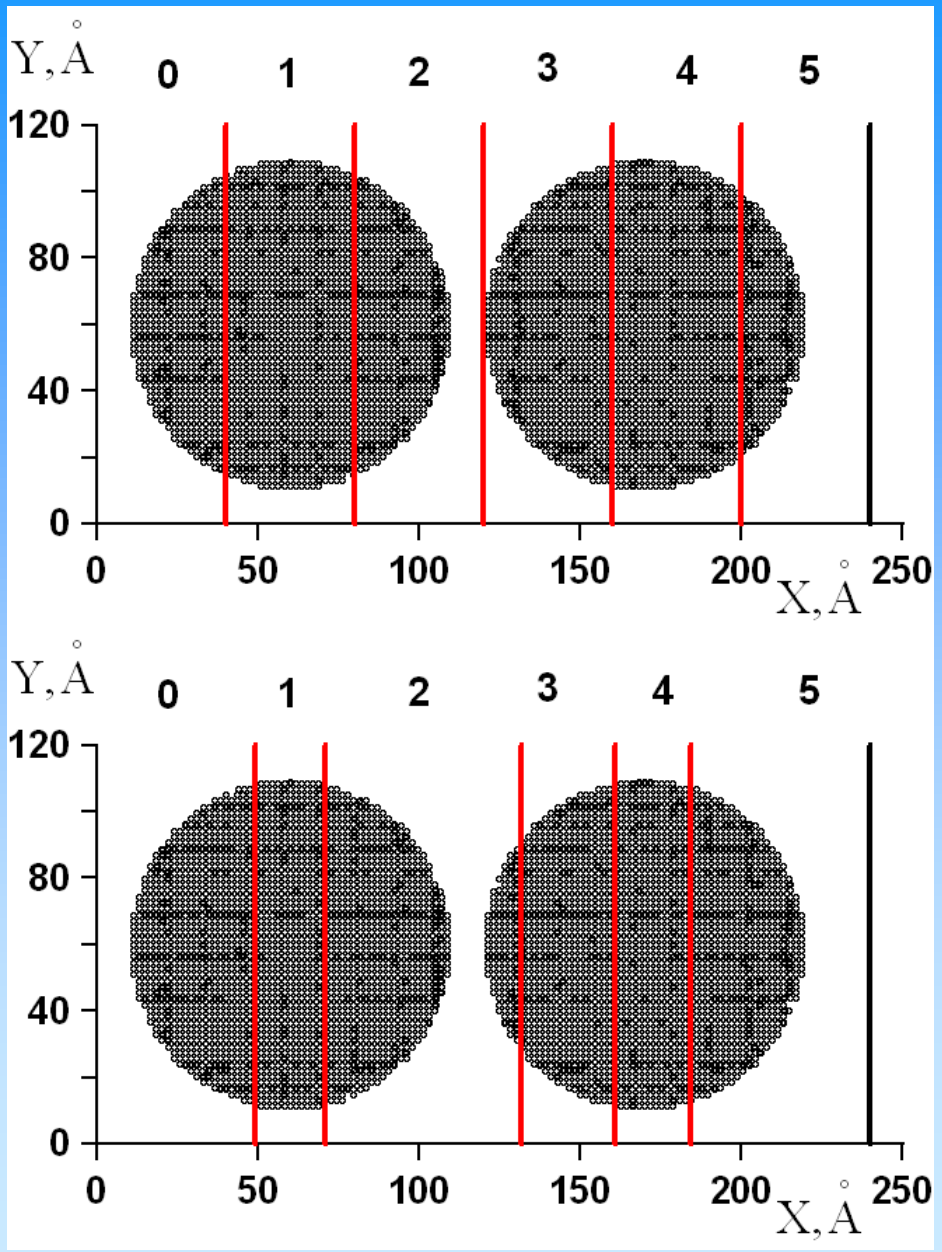
$N_{average} = N/P$ - среднее число атомов, которое должно приходиться на каждый процессор.

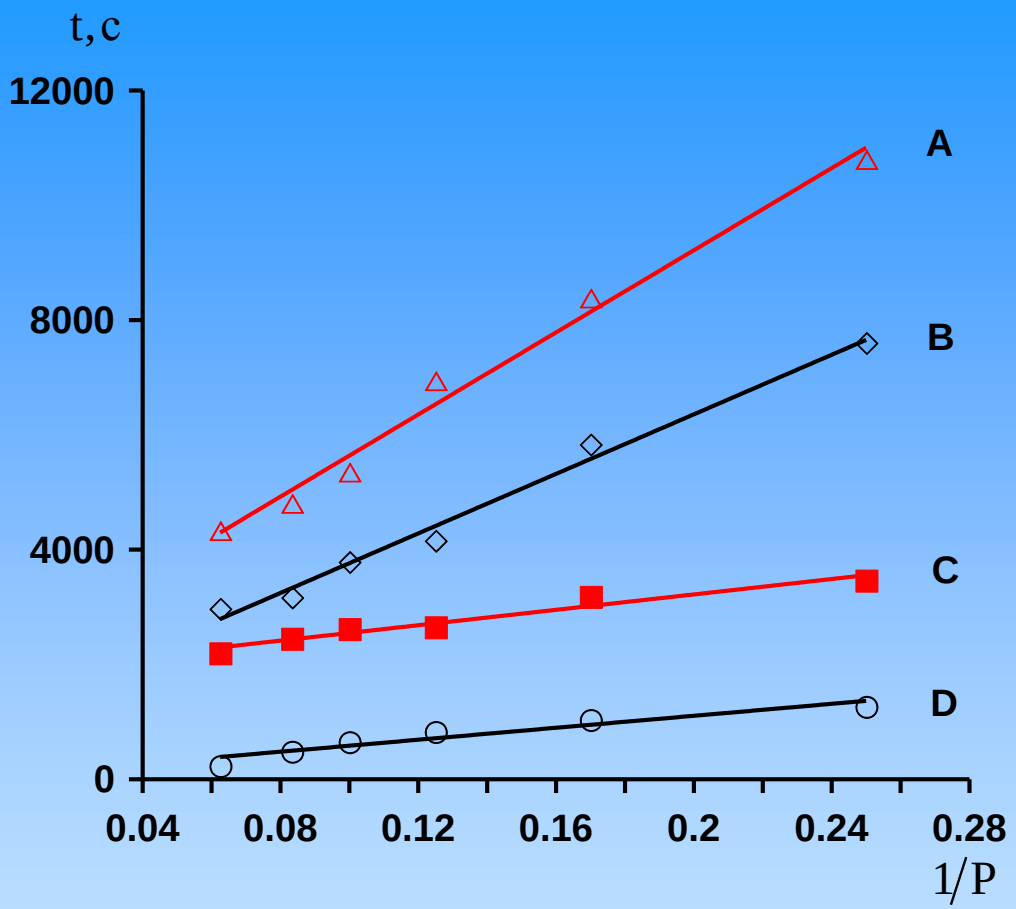
N_P - реальное число атомов в каждой подобласти.

$\left|\left(N_P - N_{average}\right)/N_{average}\right| \geq D$ -условие запуска алгоритма глобальной перестройки подобластей.

D - мера дисбаланса.

Схематическое изображение перестройки областей





Зависимость полного времени счета задачи (A, B) и времени межпроцессорных обменов (C, D) от величины, обратной числу узлов расчетной станции Т-Платформы КВК НРС-0012111-001 (ИТПМ СОРАН). A, C – программа без динамической балансировки, B, D – программа с динамической балансировкой. Система состоит из 709214 атомов, диаметр кластеров 200 A, 10000 шагов $\tau = 10^{-16}$ c

Потенциал EAM [Johnson R.A. Alloy models with the embedded-atom method // Phys.Rev.B. 1989,V. 39, P. 12554]

5. Программная реализация параллельного алгоритма для графических процессоров основанная на технологии CUDA NVIDIA

Достоинства:

Отсутствие явного влияния геометрии расчетной области (неявно геометрия задачи учитывается при быстрой сортировке в методе связанных списков).

Недостатки:

Низкая скорость передачи данных из главной памяти компьютера в память GPU

Таким образом, основная задача состоит в минимизации обмена данными между оперативной памятью компьютера и памятью GPU.

Было создано два варианта кода для GPU.

Вариант 1.

Основан на методе связанных списков (Cell linked list method).

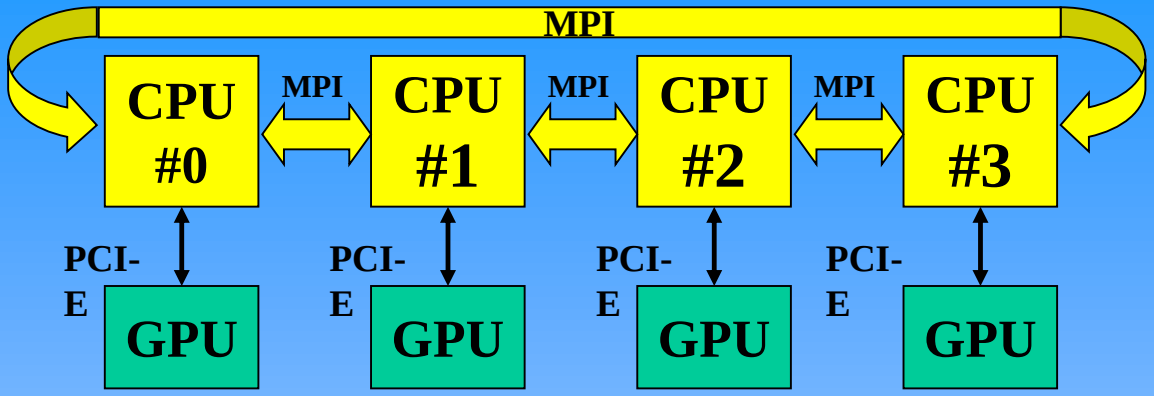
Основной недостаток: не удалось использовать закон Ньютона при расчете сил и сократить число соседних ячеек с 26 до 13

Причина: race condition

Вариант 2.

Основан на гибридной комбинации метода связанных списков и списка Верле (гибридный список Верле).

6. Программная реализация с использованием гибридной технологии MPI и CUDA NVIDIA



Идеология обменов информацией

Особенности программы:

Основа программы – базовый MPI код с динамической балансировкой.

Расчет силовых взаимодействий является самой трудоемкой частью вычислений, именно эта часть переносилась с CPU на GPU.

Код использует метод связанных списков (Cell linked list method).

Операция копирования данных с CPU на GPU отнимает очень много времени, поэтому метод связных списков создавался непосредственно на GPU.

Основной недостаток:

Необходимость копирования большого количества информации между CPU и GPU на каждом расчетном шаге, с целью организации обмена данными между GPU о перемещении атомов внутри расчетной области.

Меняющееся на каждом шаге число атомов на каждом вычислительном ядре CPU делает бессмысленным использование списков Верле.

Дополнительный недостаток:

Не удалось использовать закон Ньютона при расчете сил и сократить число соседних ячеек с 26 до 13 (race condition)

7. Сравнительный анализ различных подходов

Физическая система

В прямоугольный резервуар 2500*2500*3000 А помещалась система состоящая из 503000 атомов. Взаимодействие описывалось потенциалом Леннарда-Джонса. Общее время расчета составляло 0.05 нс.- 100000 шагов 5 10^{-16} с.

Система / Время	MPI (9CPU[1])	MPI+CUDA (9CPU+9GPU[1])	CUDA (1GPU[1])	CUDA (1 GPU[1])	CUDA (1 GPU[2])
Полный расчет	11600	6840	4379* 4853**	839* 5096**	1602* 11477**
Расчет сил	2100	548	3996* 3996**	137* 138**	199* 202**
Обмены между CPU и GPU		507			
Метод сортировки	CLLM	CLLM (26 соседних ячеек)	CLLM	CLLM+Verlet list	

Время в секундах

1 – Гибридный кластер ССКЦ СОРАН - НКС-30Т (40 узлов, на каждом два Xeon X5670 (2.93 GHz) CPUs и 3 NVIDIA Tesla M 2090 GPUs.

2 – ПК с GPU Tesla C1060 и CPU i7-920

*Обновление списка - 20 временных шагов.
** Обновление списка – каждый шаг.

Зависимость времени (секунды) расчета (MPI+CUDA) программы от числа GPU.

В прямоугольный резервуар 2500*2500*12550 А помещалась система Леннарда-Джонса состоящая из 2516100 атомов. Общее время расчета составляло 0.05 нс.- 100000 шагов 5 10^{-16} с.

GPU / Время	45	36	30	9
Полный расчет	6423	10208	10564	21426
Расчет сил	470	761	1069	2916
Обмены между CPU и GPU	636	584	489	957

Запуск CUDA 1 GPU Гибридный список Верле

Sl039 0: cudaMalloc: 7880410512 bytes requested;

not enough memory: 2(out of memory) rank 0 in job 1 sl039_54127 caused collective abort of all ranks

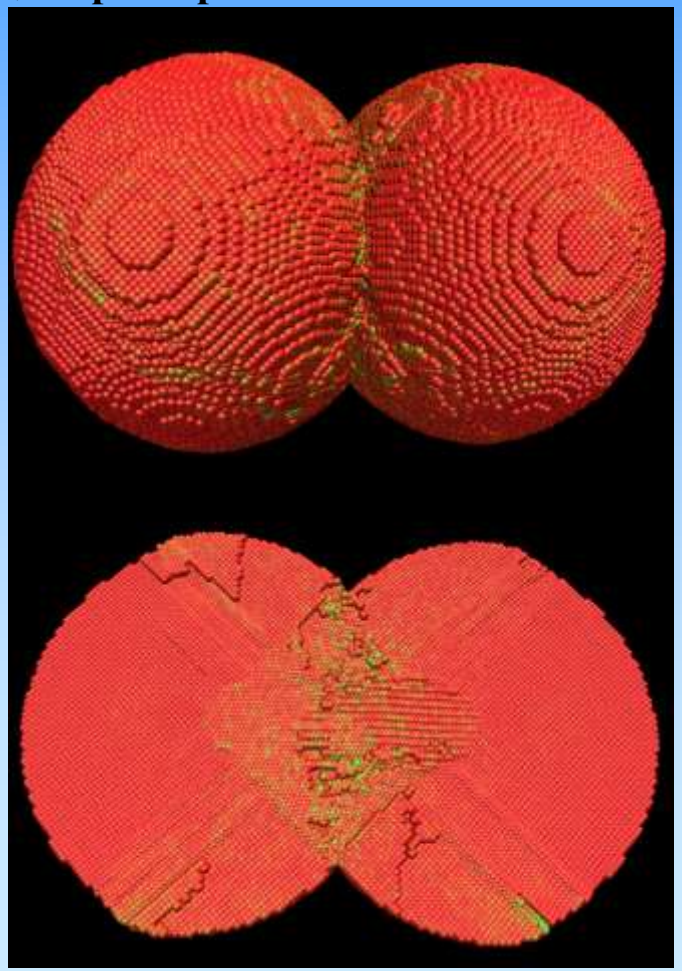
exit status of rank 0: killed by signal 9

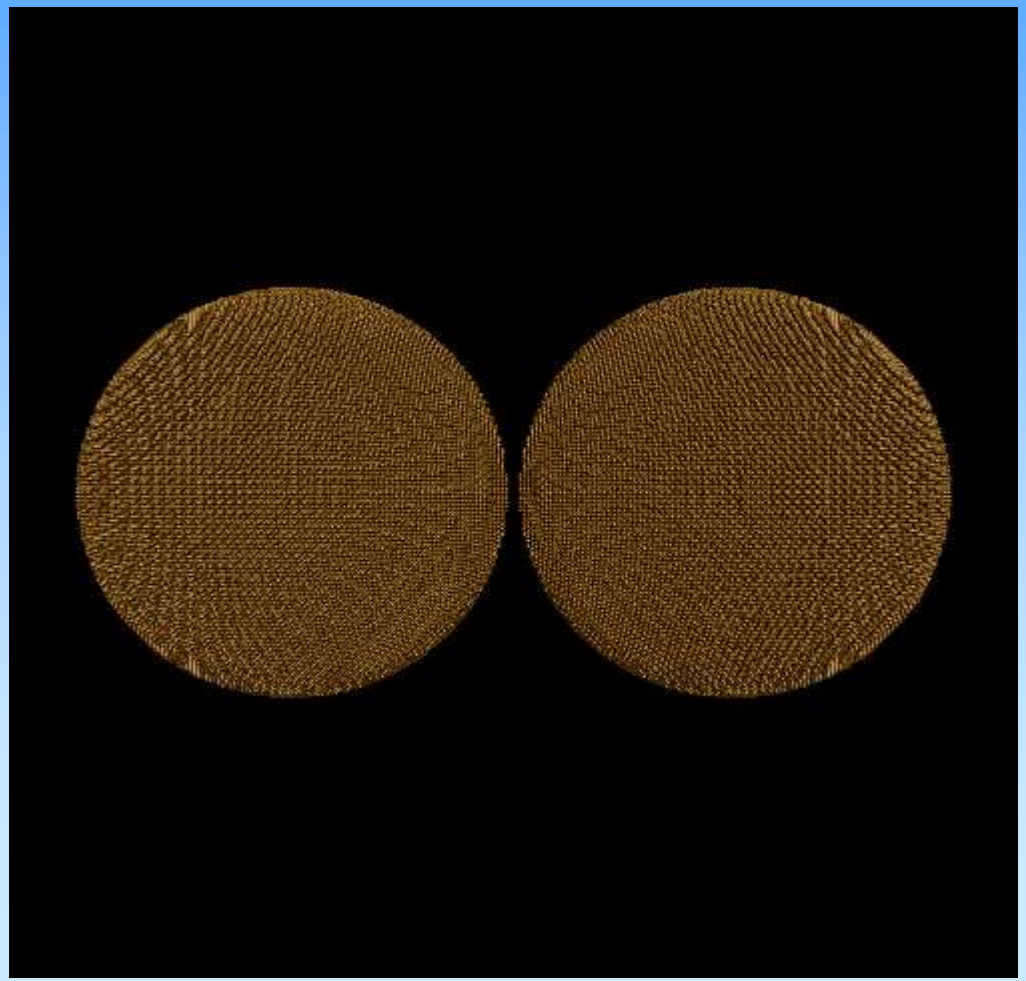
Физическая система.

Процесс столкновения сферических кластеров меди. Число атомов системы составляло от 88230 до 709214 атомов.

Межатомное взаимодействие описывалось методом внедренного атома. [Voter A.F. Embedded Atom Method Potentials for Seven FCC Metals: Ni, Pd, Pt, Cu, Ag, Au, and Al //Los Alamos Unclassified Technical Report LA-UR-93-3901, 1993.]

Общее время расчета составляло 0.005 нс.- 50000 шагов 10^{-16} с.





Зависимость времени расчета от различных факторов

Система / Число атомов	MPI (9CPU[1])	MPI+CUDA (9CPU+ 9GPU[1])	CUDA (1 GPU[1])	CUDA (1 GPU[2])
243644	16175	8067	5269	10238
709214	32156	14529	16970	30429
Метод сортировки	CLLM	CLLM (26 соседних ячеек)	CLLM+Verlet list*	

Время в секундах

1 – Гибридный кластер ССКЦ СОРАН - НКС-30Т (40 узлов, на каждом два Xeon X5670 (2.93 GHz) CPUs и 3 NVIDIA Tesla M 2090 GPUs.

*Обновление списка - 20 временных шагов.

2 – ПК с GPU Tesla C1060 и CPU i7-920

Спасибо за внимание!