

# Фреймворк для распараллеливания модели масс-с-пружинками под различные архитектуры

Артур Гудиев  
344 группа

Научный руководитель  
Сергей Николаев

# Постановка задачи

- Провести профилирование имеющегося модуля масс-с-пружинками
- Реализовать фреймворк для распараллеливания модели масс-с-пружинками
- Измерить время вычислений на OpenMP, CUDA и OpenCL
- Сравнить результаты

# Актуальность

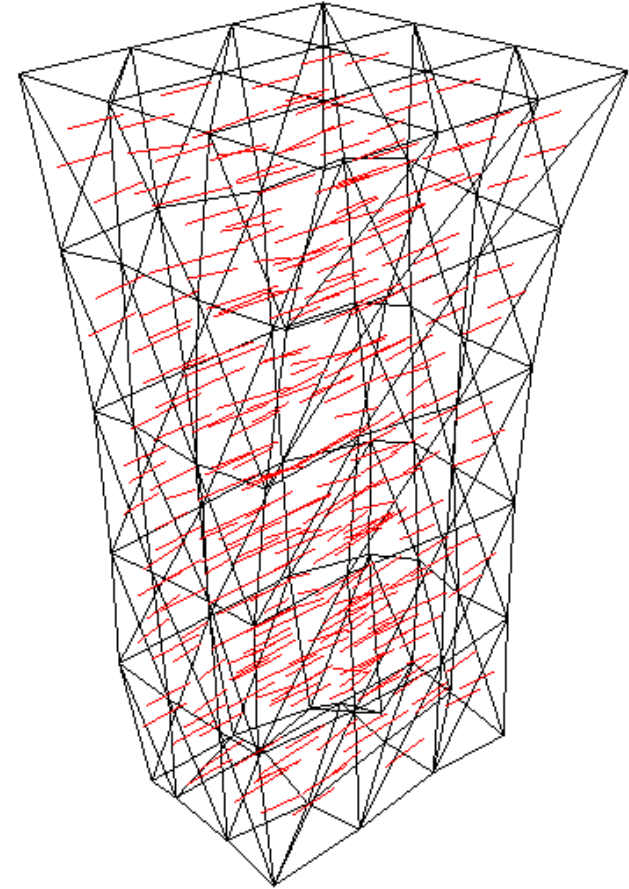
- Проект “Биомоделирование” использует для расчетов хирургических операций модель масс-с-пружинками
- Ресурсоемкость моделирования биоматериалов в механике  $\Rightarrow$  необходимость распараллеливания вычислений для экономии времени

# Обзор архитектур

- OpenMP — открытый стандарт для распараллеливания программ, написанных на языках C, C++ и Fortran
- OpenCL — фреймворк для написания программ, связанных с параллельными вычислениями на различных графических и центральных процессорах
- CUDA — программно-аппаратная архитектура параллельных вычислений, с использованием графических процессоров

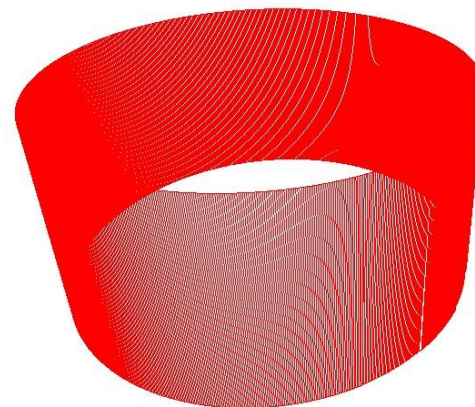
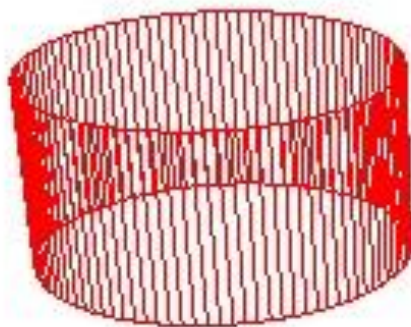
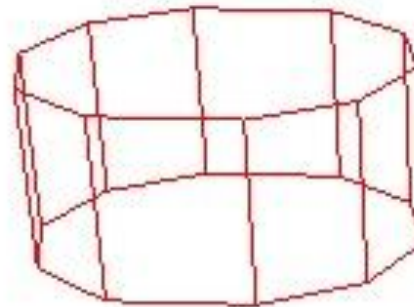
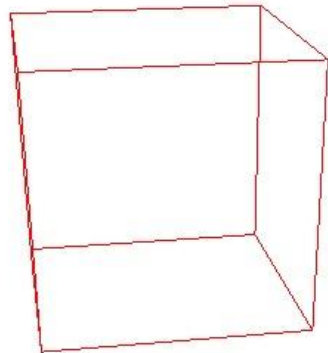
# Модель масс-с-пружинками

- При моделировании объект разделяется на вершины-массы, которые соединяются между собой пружинками. Каждая вершина имеет свои координаты, силу, которая действует на нее и ускорение. Каждая пружинка имеет начальную длину, упругость и вязкость .



# Рассмотренные объекты

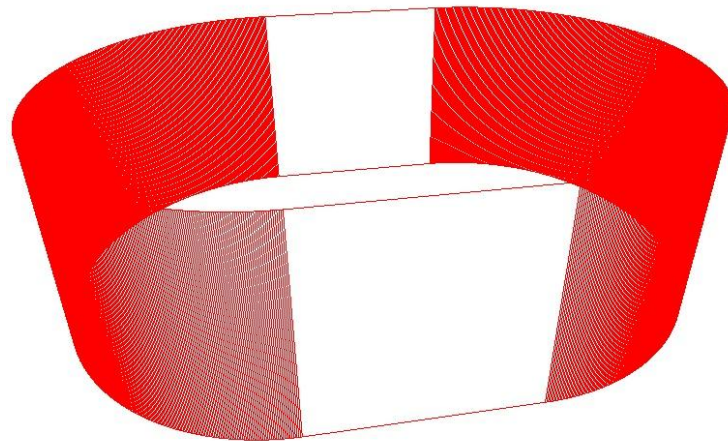
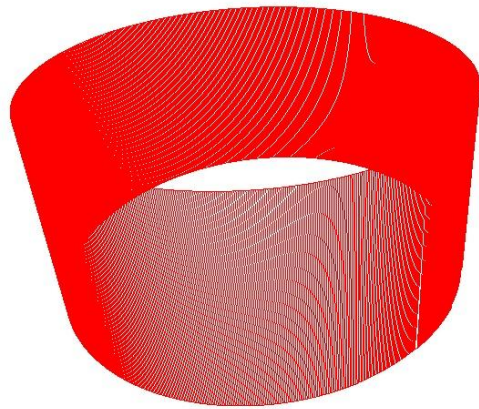
При реализации фреймворка были рассмотрены простейшие геометрические фигуры, такие как куб и различные призмы с 20, 200 и 2000 вершинами, половина из которых была зафиксирована.



# Рассмотренные операции

Применялась заранее заданная постоянная сила, получились простейшие операции:

- Растяжение
- Сжатие



# Структура модуля модели масс-с-пружинками

- Процедура `ConnectSprings()`
- Процедура `DrawSystem()`
- Процедура `DoCycleIteration()`



# Псевдокод функции DoCycleIteration()

---

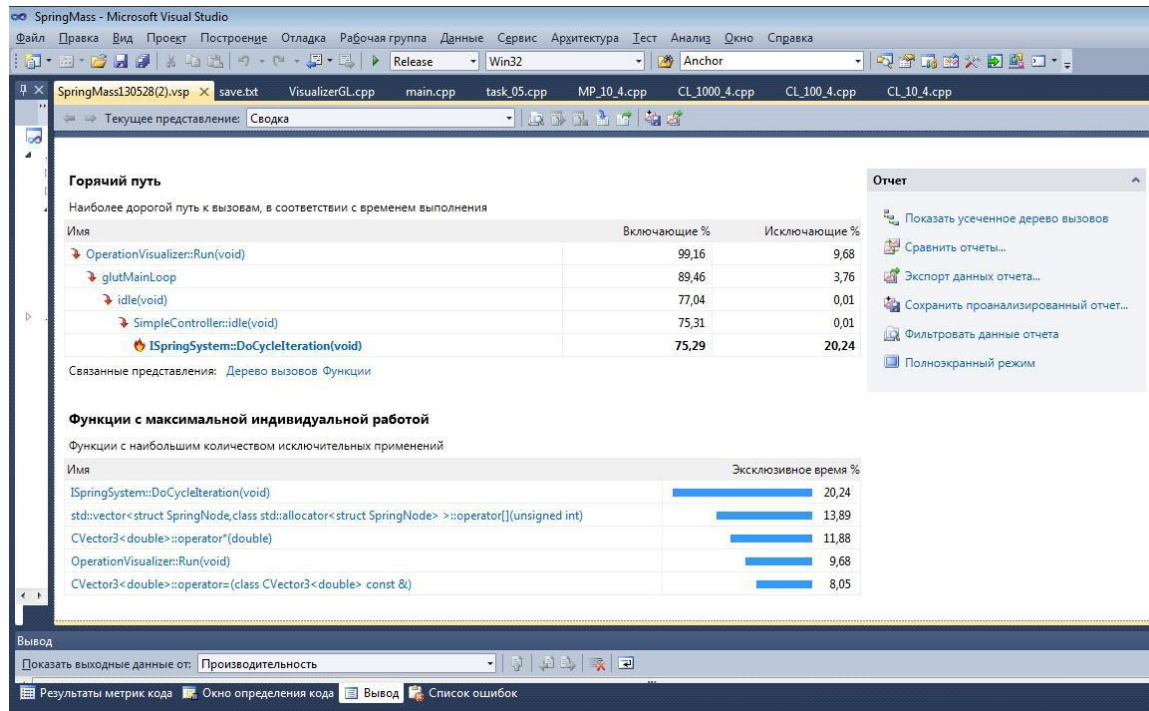
Algorithm 1 DoCycleIteration()

---

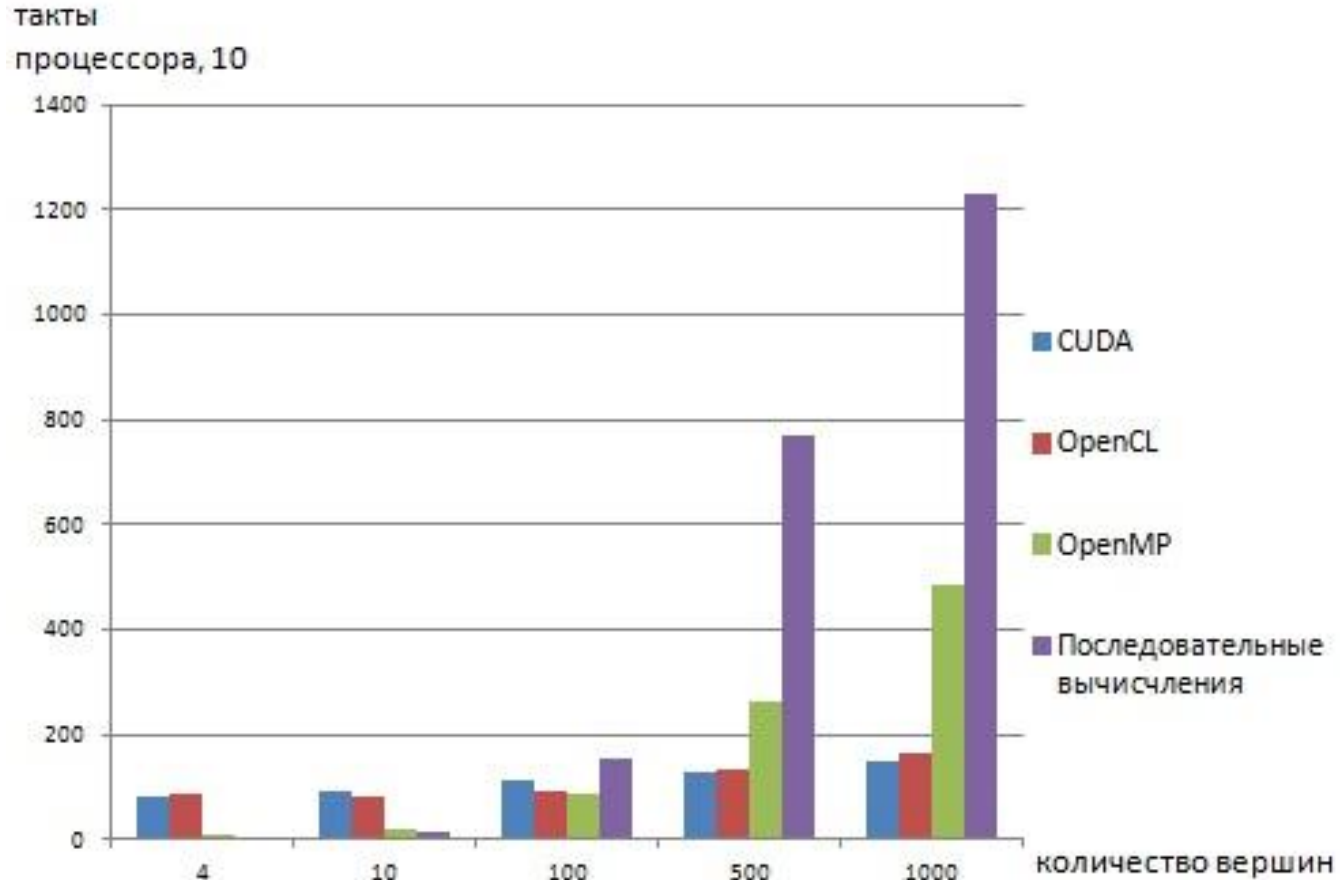
```
1: time  $\leftarrow$  cureTime
2: for k  $\leftarrow$  1 to Nodes.size() do
3:   if Nodes[k].isFixed = false then
4:     Nodes[k].position  $\leftarrow$  newPosition
5:     Nodes[k].velocity  $\leftarrow$  newVelocity
6:   end if
7: end for
8: sum  $\leftarrow$  sum + (cureTime - time)
```

# Профилирование

- Профилирование применялось при различном количестве вершин объекта: 8, 20, 200, 2000...



# Полученные оценки по времени



# Результаты

- Осуществлено профилирование модуля масс-с-пружинками
- Реализован фреймворк для распараллеливания модели масс-с-пружинками
- Сделано сравнение последовательных вычислений, вычислений на OpenMP, OpenCL и CUDA
- <https://github.com/ArturGudiev/SpringMass>