

Описание программного комплекса для моделирования кубантов.

Предисловие.

Для того, чтобы моделировать кубанты или кубические комплексы (комплексы кубантов) было создано специальное программное обеспечение. Оно позволяет, с одной стороны производить вычисления и получать некоторые результаты над кубантами, и, с другой стороны — визуализировать полученные результаты.

Для того, чтобы эта многофункциональность было доступна, была разработана архитектура, состоящая из трёх практически независимых частей:

1. «Ядро»
2. «Интерпретатор»
3. «Визуализатор»

Опишу каждую из этих частей:

Ядро.

Ядро — это основная часть программы, которая предоставляет возможность создавать кубанты, производить с ними различные операции, получать результат. Ядро системы написано на языке программирования C++, для того чтобы обеспечить кроссплатформенность. За счёт этого, все программы, написанные с использованием библиотеки-ядра могут работать как и на обычных компьютерах, так и на суперкомпьютерах.

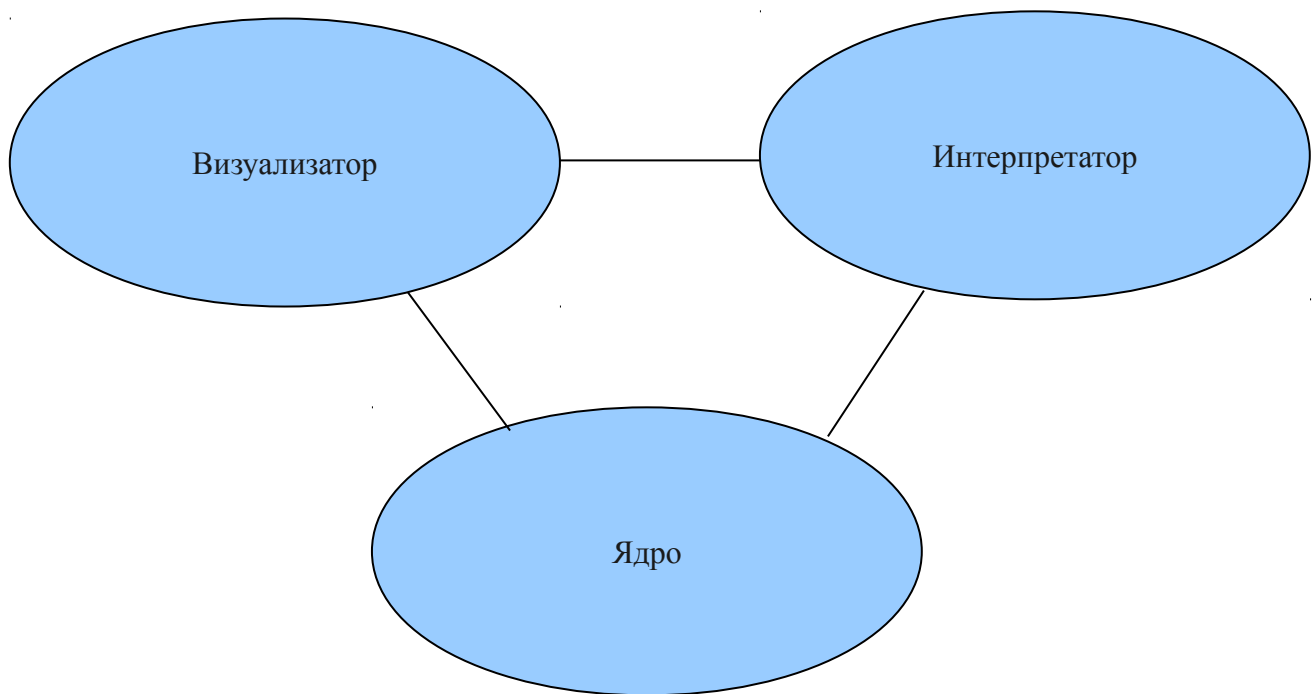
Интерпретатор.

Для того, чтобы произвести простой расчёт или просто произвести вычисления, не требующие больших вычислительных мощностей, можно использовать интерпретатор. Его использование позволяет быстро написать программу, не компилируя её. В качестве интерпретируемого языка в системе моделирования кубантов используется JavaScript. Интерпретатор может использоваться для предварительных расчётов, для поиска вариантов решения. После того, как подходящий алгоритм найден, мы можем реализовать его, используя непосредственно ядро, и прогонять его как на обычном настольном компьютере, так и на кластере.

Визуализатор.

Чтобы более полно представлять картину происходящего, в систему был добавлен визуализатор. Он используется для непосредственного отображения кубантов. С помощью визуализатора мы можем отобразить репер, кубант или несколько кубантов, задать различные цвет для каждого кубанта.

Все три компонента этой системы являются легко заменяемыми на другие, с аналогичным интерфейсом. За счёт этого достигнута гибкость — интерпретатор не обязательно должен быть интерпретатором языка Javascript, а может быть интерпретатором любого скриптового языка, например Lua или ActionScript. Практически это свойство заменяемости использовалось для построения двух различных визуализаторов: один 2D, а другой 3D, между которыми мы можем свободно переключаться. Так, рисуя кубант или кубический комплекс, мы получаем два (в текущей реализации) варианта его представления. Первый — это двухмерный вариант с плоским репером (для плоского варианта в текущей реализации репер мы можем задавать самостоятельно) и трёхмерный вариант, который отображает кубант, используя конусообразный репер.



Текущая схема Ядро — Интерпретатор — Визуализатор является реализацией известного паттерна проектирования MVC — Model — Controller — View , где модель — это ядро, контроллер — это интерпретатор, а View — это визуализатор.

Поддерживаемые операции.

Так как система моделирования кубантов является хорошо расширяемой системой, то количество операций в Ядре постоянно увеличивается. Поэтому я опишу некоторые из них, которые уже давно реализованы и стабильны.

Операция «Оболочка кубантов из гиперграней»

Данная операция по одному кубанту размерности больше 0 (т. е. не точке) создаёт множество кубантов, являющимися его гипергранями. То есть, для кубанта $/2,0,1/$ результат выполнения этой операции будет множество кубантов $(/0,0,1/, /1,0,1/)$. С точки зрения языка Javascript (компонент интерпретатор) эта операция выглядит так:

```
var cubant=createCubant("/2,0,1/");
System.print(cubantFacet(cubant));
```

В первой строке создаётся кубант $/2,0,1/$, и сохраняется в переменной cubant. Во второй строке к переменной cubant применяется операция «Оболочка кубантов из гиперграней», и сразу после выполнения этой операции выводится результат. Полученный результат — это Javascript-массив:

```
[ "/0,0,1/" , "/1,0,1/" ]
```

Таким образом с помощью интерпретатора мы можем достаточно быстро необходимую нам информацию.

