

УДК 004.94

ПРЕДСТАВЛЕНИЕ ДАННЫХ В БЛОЧНО-ОРИЕНТИРОВАННОЙ ПАРАЛЛЕЛЬНОЙ СИСТЕМЕ МОДЕЛИРОВАНИЯ

Лысенко Д.В.

Донецкий национальный технический университет

Производится сравнительный анализ различных способов представления блочно-ориентированной схемы в распределенной параллельной моделирующей системе. Рассматриваются три способа – в виде «сырых» списков элементов и соединений, в виде XML-документа и в формате JSON.

Введение

Процесс моделирования объектов различной природы используется во многих предметных областях. Модели объектов постоянно усложняются, увеличивается количество вычислительной работы для проведения моделирования.

В то же время существуют некоторые тенденции в увеличении производительности вычислительных систем. Ранее практически повсеместно использовалась архитектура SIMD, увеличение производительности происходило за счет улучшения техпроцесса изготовления вычислительных элементов. Например, увеличивалась разрядность процессоров и их тактовая частота. В настоящее время увеличение производительности ЭВМ происходит за счет использования параллельных архитектур. Для персональных компьютеров все современные процессоры среднего уровня имеют от двух до восьми ядер, вычислительная работа распределяется между ними. Дальнейшее увеличение производительности (для сложных вычислительных задач, например) возможно при использовании распределенных вычислительных систем – вычислительных узлов, объединенных в кластеры.

Эта особенность развития вычислительной техники предъявляет дополнительные требования к программному обеспечению. Даже при разработке программного обеспечения для мобильных устройств, необходимо использовать распределение вычислений, например, использовать многопоточность.

Моделирующая среда должна быть организована в виде сложной распределенной компьютерной системы и состоять из нескольких подсистем [1]. Необходимыми подсистемами являются подсистемы топологического анализа, распределения нагрузки, хранения информации, диалога с пользователем и другие.

В данной статье анализируются возможные форматы для хранения данных системы хранения информации. Было рассмотрено представление в памяти ЭВМ информации о моделируемой схеме при использовании блочно-ориентированного подхода.

Для примера рассмотрим возможные способы представления схемы на рис. 1.

Практическая ценность моделирования данной схемы не велика, но на ее примере удобно рассмотреть способы представления схемы в памяти компьютера.

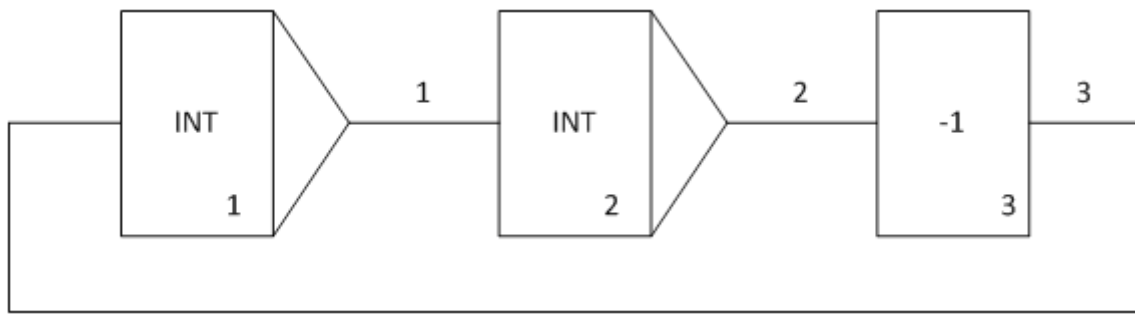


Рисунок 1. Блочная схема генератора синусоиды

1. Представление в виде «сырых» данных

Одна из форм представления графа – два списка (ребер и вершин). В данном случае ребрами можно считать соединения блоков, а вершинами – сами блоки. Так как существует достаточно много типов блоков, дополнительно нужно хранить информацию об их типе и, если необходимо, другую информацию о блоке, например исходное состояние. Так же нужно учесть, что блоки могут иметь несколько входов, в списке ребер нужно хранить дополнительно номер входа блока. Рассмотрим на практике представление схемы в виде двух списков. Тип блока интегратор будет иметь код 1, умножитель на константу – 2. Для интеграторов необходимо указывать исходное состояние, для умножителей на константу – саму константу, на которую производится умножение. Файл для описания схемы может иметь следующий вид:

```

3
1 1
1 0
2 -1
3
1 2 1
2 3 1
3 1 1

```

Сначала указывается количество блоков, 3. Затем для каждого из блоков указывается его тип (интегратор либо умножитель на константу) и параметры (их может быть разное количество для разных типов блоков) – исходные значения для интегратора и константа для умножителя. У приведенного представления есть очевидный недостаток – сложность ее восприятия человеком. Читающий (или составляющий) представление схемы человек должен помнить все типы блоков, для каждого типа помнить параметры и их очередность.

2. Представление в виде XML-документа

Альтернативный способ представления – в виде документа XML. XML – extensible markup language [2], расширяемый язык разметки. Он позволяет более наглядно представить информацию о схеме, и эту информацию можно будет достаточно легко обрабатывать, так как существуют библиотеки для разбора данных в формате XML практически для всех популярных языков программирования.

Ниже представлен пример XML-файла для описания схемы на рис. 1.

```
<bo _ scheme>
<general>
<scheme _ name>Block-oriented sine generator<>
<block _ count>3</block _ count>
<connection _ count>3</connection _ count>
</general>
<block type="integrator">
<id>int1</id> <initial _ condition>1</initial _ condition>
</block>
<block type="integrator">
<id>int2</id> <initial _ condition>0</initial _ condition>
</block>
<block type="constant _ multiplier">
<id>cmul</id> <constant>0</constant>
</block>
<connection from="int1">
<to>int2</to> <slot>1</slot>
</connection>
<connection from="int2">
<to>cmul</to> <slot>1</slot>
</connection>
<connection from="cmul">
<to>int1</to> <slot>1</slot>
</connection>
</bo _ scheme>
```

По сравнению с первым способом представления, данный способ имеет много преимуществ:

1. XML – широко используемый формат представления данных, для его обработки существует много стабильных отлаженных библиотек, которые могут проверять файл с описанием схемы на наличие некоторых ошибок, например отсутствие необходимых элементов описания.
2. Она является более понятной для человека, чем набор цифр.
3. Формат является очень гибким, можно добавить в описание схемы практически любую информацию, например программу для моделирования определенного типа блока.

Среди недостатков можно отметить хранение избыточной информации и необходимость использования существующих библиотек для разбора формата, некоторые из них потребляют достаточно много ресурсов.

3. Представление в формате JSON

Еще один вариант представления – JSON (Javascript object notation). От формата XML он отличается меньшей избыточностью. Этот формат так же является очень распространенным, в частности, для сериализации объектов. Представление схемы в этом формате может иметь следующий вид:

```
{
  "general": {
    "scheme _ name": "Block-oriented sine generator",
```

```

"block_count":3,
"connection_count":3 }
"block": {
"type":"integrator","id":"int1","initial_condition":1}
"block": {
"type":"integrator","id":"int2","initial_condition":0}
"block": {
"type":"constant_multiplier","id":"cmul","constant":0
}
"connection": {
"from":"int1","to":"int2","slot":1
}
"connection": {
"from":"int2","to":"cmul","slot":1
}
"connection": {
"from":"cmul","to":"int1","slot":1
}}

```

Данное представление схемы является более лаконичным по сравнению с XML, в то же время она остается достаточно понятной и наглядной.

Выводы

В данной статье был рассмотрен один из вопросов архитектуры системы блочно-ориентированного моделирования – представления исходных данных для системы, блочно-ориентированной схемы. Представление в виде «сырых» данных – списка вершин и списка ребер является не целесообразным, гораздо удобнее использовать один из языков разметки – XML либо JSON.

При построении распределенных компьютерных систем вышеперечисленные стандарты уже активно используются, например, для сериализации объектов. При этом разные части системы могут быть разработанными на разных языках программирования.

Например, существует интерфейс обращения к сервису appstore для поиска контента, результаты в котором возвращаются в формате JSON. Таким образом, можно создать свою программную систему, которая использовать этот интерфейс для доступа к магазину, и разработчики не будут ограничены одной технологией.

Список использованных источников

- [1] Feldmann L.P., Svjatnyj V.A., Resch M., Zeitz M. FORSCHUNGSGEBIET: PARALLELE SIMULATIONSTECHNIK / сайт национальной библиотеки им. В.И. Вернадского / Электронный ресурс. Режим доступа: <http://www.nbu.gov.ua/portal/natural/Npdntu/Pm/2008/08flpfps.pdf>
- [2] Extensible Markup Language (XML) 1.0 (Fifth Edition)./ Электронный ресурс. Режим доступа: <http://www.w3.org/TR/REC-xml/>
- [3] Introducing JSON / Электронный ресурс. Режим доступа: <http://json.org>