

- 
17. Гольдштейн Г.Я. Стратегический инновационный менеджмент: учеб. пособие. – Таганрог: Изд-во ТРТУ, 2004. – 267 с.
  18. Knowledge Management in Research and Development / F.M. Armbrecht [et al.] // RTM. – 2001. – V. 44, №4.
- 



УДК 519.67

В.В. Тараскин

## РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ ПРИ МОДЕЛИРОВАНИИ ПРОЦЕССОВ

В статье рассматривается необходимость использования распределенных вычислений при реализации расчетов сложных соединений и возможные пути решения. Рассмотрена методология использования распределенных вычислений для решения сложных прикладных задач.

**Ключевые слова:** распределенные вычисления, модель системы, имитационное моделирование, методики.

V.V. Taraskin

## THE DISTRIBUTED CALCULATIONS IN PROCESSES MODELING

*The necessity of distributed calculations use while realizing difficult connections calculations and possible solutions is considered in the article. The methodology of distributed calculations use for the complex applied challenges solution is given.*

**Key words:** distributed calculations, system model, imitating modeling, techniques.

---

Распределенные вычислительные системы играют значительную роль в современном мире. Распределенные вычисления используются в науке, на производстве, в энергетике, в военном деле, в корпоративном программном обеспечении и во множестве других областей. Область применения распределенных систем быстро расширяется с развитием интернета.

В силу высокой сложности распределенных систем практически невозможно спроектировать и разработать надежную систему, не содержащую ошибок. Вероятность возникновения программного или аппаратного сбоя при эксплуатации распределенной системы довольно высока и может иметь очень серьезные негативные последствия. Таким образом, актуальной является задача повышения надежности эксплуатации или задача построения отказоустойчивых распределенных систем, которые могут самостоятельно устранять последствия сбоев без прекращения работы системы. Неотъемлемой частью отказоустойчивой системы является система мониторинга, обнаруживающая сбои в целевой системе и передающая сведения о сбоях системе восстановления.

Существует значительное количество методик построения отказоустойчивых систем [2]. Одним из наиболее перспективных направлений является возвратное восстановление, заключающееся в восстановлении системы к некоторому ранее сохраненному стабильному состоянию в случае сбоя. Стоит отметить, что в некоторых случаях применение системы возвратного восстановления может быть более дешевым решением, чем устранение причин отказа.

Построение модели отказоустойчивой системы с применением методики возвратного восстановления является достаточно сложной задачей, имеющей значительное количество особенностей. Создание модели восстановления требует решения следующих проблем: обеспечение целостности взаимодействия элементов системы, обеспечение целостности взаимодействия системы и внешних устройств, обеспечение надежного сохранения и восстановления информации о состояниях системы, обеспечение высокой производительности процесса восстановления.

Для многих современных работ в области возвратного восстановления характерны следующие недостатки [4]:

1. Сильные предположения относительно характера возможных сбоев. В частности, в большинстве работ рассматриваются только фатальные (например, аппаратные) сбои, приводящие к немедленной остановке процессов, хотя для многих реальных систем более характерны распространяющиеся программные сбои.

2. Сильные предположения относительно возможностей детектирования сбоев. Как правило, считается, что система мониторинга является централизованной, обладает полными сведениями об ошибках и минимальным временем реакции.

Актуальность данных предположений обусловлена тем, что одной из важнейших проблем современных распределенных систем являются ошибки синхронизации процессов, приводящие к распространяющимся сбоям. В случае ошибок синхронизации или ошибок, связанных с переполнением буфера, как правило, сложно установить истинную причину ошибки. Использование централизованной системы мониторинга в высоко распределенных системах, состоящих из большого количества узлов, может быть крайне неэффективным. Важной особенностью распределенной системы мониторинга является вероятностный характер сообщений об ошибках в связи с конечным временем синхронизации сведений, получаемых из разных узлов системы.

Отмечается высокая стоимость отказов распределенных систем, обусловленная высокой стоимостью владения такими системами и важностью решаемых с их помощью задач. В качестве причины отказов распределенных систем рассматриваются ошибки, допускаемые на этапах проектирования, разработки и эксплуатации. Для этапов проектирования и разработки существует значительное количество методик устранения ошибок, как правило, сводящихся к автоматизации процесса разработки. На этапе эксплуатации помимо сбоев в результате ошибок, допущенных при разработке, существует возможность аппаратных сбоев, таких как отказ узла сети или нарушения связи между узлами. Основным выводом из этого является вывод о необходимости построения отказоустойчивых систем, то есть систем, способных ликвидировать последствия сбоя без прекращения работы.

Популярные средства параллельного программирования, такие как MPI и OpenMP, требуют от программиста подробного описания большого количества сущностей [3]. Ему необходимо заботиться о распределении вычислительных задач, синхронизации, обмене данными и так далее. Существуют различные подходы к упрощению процесса программирования и исполнения параллельных вычислений. С одной стороны, создаются универсальные средства автоматического распараллеливания программ (как для исполнения в системах с общей памятью, так и в многомашинных конфигурациях). С другой стороны, создаются среды для решения определенных классов задач (в основном это касается задач, для которых применим параллелизм «по данным»). Также разрабатываются универсальные инструменты, пытающиеся упростить технические аспекты процесса программирования параллельных и распределенных систем.

Рассмотрим, как компонента осуществляет синтез функционирования сложной системы. Модель сложной системы должна быть готова к моделированию с переменным шагом времени [2]. Иначе поезд не доедет до станции или проедет ее, ракета пролетит мимо цели и т. д. Считается, что для компоненты задан некий шаг моделирования  $D t$  по умолчанию.

1. В момент  $t$  в начале каждого шага моделирования вызываются все методы прогноза событий для всех процессов компоненты.

2. Если есть наступившие события, в соответствии с автоматными функциями перехода выбираются новые текущие элементы.

3. Если среди текущих элементов есть мгновенные (не занимающие модельного времени) – они выполняются. В этом случае считается, что выполнился шаг моделирования нулевой длины. Повторяется вызов прогнозов наступления событий (т. е. возвращаемся к пункту 1).

4. Если мгновенных элементов нет, повторяется вызов прогнозов событий (т.е. возвращаемся к пункту 1).

5. Если наступивших событий нет – выбирается минимальное значения прогноза события  $t^*$ . Если  $t < D t$  – моделирование идет со стандартным шагом  $D t$ , если же нет, в качестве шага моделирования берется  $t^* - t$ . С выбранным шагом времени вызываются все распределенные элементы. Этим заканчивается очередной шаг моделирования.

Может возникнуть вопрос, а не окажется ли у множества событий точек накопления, при предлагающей системе выбора шага моделирования? Отчасти на него отвечает утверждение предыдущего раздела, где рассматривалась связь динамических и объектных моделей. Например, если элементы компоненты описаны дифференциальными уравнениями и все правые части этих дифференциальных уравнений удовлетворяют условию Липшица – точек накопления не будет.

Система распределенного имитационного моделирования представляется пиринговой (например, <http://www.ip2p.ru/articles/>) сетью средней степени централизации.

Несколько десятков серверов глобальной сети имитационного моделирования должны выполнять следующие функции обслуживания рабочих станций (количество рабочих станций зависит от количества компьютеров, на которых содержатся модели компонент, предназначенных для генерации распределенного моделирующего комплекса; при достаточном развитии предлагаемой технологии распределенного имитационного моделирования их может быть очень много – до нескольких тысяч):

- Регистрация клиентских рабочих станций и предоставляемых ими в сеть разделяемых ресурсов, т.е., моделей, компонент моделей, данных, алгоритмов и т.д.

- Хранение и постоянное поддержание баз данных модельных ресурсов в сети, т.е. какими рабочими станциями предоставлены в сеть те или иные модели, компоненты моделей и прочие ресурсы. Какие из рабочих станций реально присутствует в сети, кто из них и когда был доступен в последний раз, возможно, расписаний присутствия рабочих станций в сети.

- Поиск ресурсов в сети, т.е. уже существующих моделей, компонент моделей, необходимых данных и алгоритмов (примерно так, как осуществляется поиск различных файлов в современных пиринговых файлообменных сетях).

- Осуществление функций посредников стандартных запросов рабочих станций.

- Осуществление некоторых организационных функций, например, возможности нескольким рабочим станциям договориться о проведении совместной сессии имитационных экспериментов определенной продолжительности, с началом в определенное время, или же спланировать время проведения такой сессии в соответствии с их расписаниями присутствия в сети.

Клиентское программное обеспечение рабочих станций такой сети вполне могло бы базироваться на описанных выше принципах анализа и синтеза модели. А именно, основу программного обеспечения рабочей станции составляет «система программирования» на упомянутом выше языке ЯОКК. Эта система программирования должна включать редактор (в том числе желательно и с графическим интерфейсом) описания моделей и их компонент. При описании компонент модели той или иной конструкции, должна быть возможность ссылки на найденные уже существующие в сети компоненты, подходящие на роль компонент описываемой конструкции. Далее, система программирования клиентской части должна включать средства отладки описаний отдельных компонент и всей модели в целом. Затем в клиентской части должен присутствовать сборщик модели, создающий базу характеристик компонент модели, проверяя при этом ее целостность.

Блок клиентской части инструментальной системы, ответственный за организацию имитационных экспериментов, должен с помощью существующих интеграционных технологий вызывать как локальные, так и удаленные компоненты модели по их известным адресам, передавая им необходимые характеристики из базы данных модели и принимая от них обратно в базу обработанные характеристики.

Служебные методы, предоставляемые системой, также должны быть рассчитаны как на локальное, так и на удаленное использование. На этапе имитационного эксперимента рабочая станция, проводящая эксперимент, работает напрямую, минуя сервера, с рабочими станциями, содержащими используемые в модели компоненты. Как и в современных пиринговых сетях, серверы здесь нужны лишь на этапе поиска партнеров.

Уже отлаженные, работающие модели и их компоненты могут быть объявлены доступными для всеобщего использования, и пополнить фонд доступных в сети моделей и их компонент. При этом автоматически становятся доступными также и описания модели и ее компонент на языке описаний, по которым можно судить о ее адекватности тем или иным способам применения в других моделях. Таким образом, каждая рабочая станция может предоставлять в общее пользование, как свой вычислительный потенциал, так и существующие наработки в области моделирования.

Иногда при создании подобных средств разработчики пытаются использовать нестандартные парадигмы вычислений. Одной из таких парадигм является поток данных – Dataflow [5]. В различных вариантах методики, основанные на парадигме потока данных, применяются для создания процессорных архитектур, суперкомпьютеров в целом, для программной организации вычислительных потоков в рамках одного процесса и взаимодействия процессов в распределенной вычислительной среде.

Здесь рассматривается методика и технические средства для программирования в параллельных распределенных средах. Методика основана на анализе различных, в том числе и собственных, моделей потока данных. Цель данной разработки – упростить процесс создания параллельных программ, и сделать это не в ущерб эффективности исполнения вычислительных кодов. Предлагаемая методика вычислений

возникла в результате продолжительной теоретической работы над архитектурой операционной системы для распределенных вычислений [1].

Методика базируется на понятиях хранилища, задач и правил. Хранилище содержит в себе именованные данные, по отношению к которым доступны три операции – запись (создание), чтение и удаление. Хранимые данные являются самодостаточными – это не очереди, но некие единицы информации с уникальными именами. Задачей называется программа, которая во время исполнения считывает данные с определенными именами из хранилища и в результате своего исполнения формирует новые данные, которые записываются в хранилище. Правилом называется такая конструкция, которая определяет условия и параметры запуска задач. Правило содержит в себе:

1. Список имен данных, которые необходимы для выполнения задачи.
2. Список соответствия глобальных имен данных (находящихся в хранилище) локальным именам (с которыми и будет работать задача).
3. Список задач (программ), которые необходимо запустить.
4. Действия, совершаемые в случае успешного выполнения задач (3).

Правило считается готовым к исполнению, когда в хранилище присутствуют все данные с именами из списка (1). После успешного исполнения правило удаляется из списка выполняемых правил.

Процесс программирования и проведения вычислений происходит следующим образом. Прежде всего, разрабатываются программные коды задач. При этом в рамках одного вычисления могут использоваться любые комбинации языков, а также целевых аппаратных сред. Например, часть задач можно реализовать на графических ускорителях. Затем формируется файл инициализации, в котором описываются начальные правила системы. В дальнейшем эти правила могут дополняться – при выполнении задачах или завершающих действий других правил. Кроме правил, в файле инициализации указываются начальные данные, которые помещаются в хранилище.

После подачи команды на запуск вычислительная среда ищет правила, готовые к исполнению, и запускает указанные в них задачи на подходящих свободных вычислительных ресурсах. В результате часть правил исполняется, формируя новые данные и освобождая ресурсы для других правил. Среда продолжает поиск и выполнение правил вплоть до исчерпания всех правил, приостановки работы с внешней стороны или выявления ошибки.

Предлагаемая методика позволяет достаточно просто и эффективно реализовать проведение вычислительного эксперимента на гибридных архитектурах, динамическое изменение количества вычислительных узлов во время самого вычисления, работу в глобально-распределенных условиях, автоматическое создание контрольных точек, приостановку и продолжение вычисления прозрачным для программиста образом, использование распределенные хранилища данных, а также обеспечивает ряд других преимуществ.

На основе предложенной методики в рамках проекта RIDE разрабатывается прототип среды параллельного программирования. Первые версии показывают реализуемость предлагаемых идей и лаконичность программных конструкций для описания правил. Можно надеяться, что в результате развития этой среды удастся достичь главной цели – сделать процесс создания распределенных вычислительных программ более простым и эффективным.

Таким образом, моделирование физико-химических процессов находится на грани современных вычислительных возможностей ввиду необходимости учета огромного количества взаимосвязанных процессов, действующих в широком диапазоне временных и пространственных масштабов. Высокие требования к однородности и качеству получаемых материалов обуславливают необходимость решать трехмерные задачи расчета течения химически реагирующей газовой смеси, плазменного разряда и гетерогенных процессов на поверхности раздела фаз и т.д.. Для получения результатов на реальных временах, такие расчеты должны опираться на технологии высокопроизводительных распределенных вычислений.

### Литература

1. Бахтерев М.О. Описание параллельных вычислений при помощи замыканий // Супервычисления и Математическое моделирование: тез. 10-го междунар. семинара / РФЯЦ-ВНИИЭФ. – Саров, 2008. – С. 31–32.
2. Дубовик Г.А., Миногин А.В., Устюжанин А.Е. Методика мониторинга распределенных многоагентных систем // Современные технологии в задачах управления, автоматики и обработки информации: тр. XVI междунар. науч.-техн. семинара. – Тула: Изд-во ТулГУ, 2007. – С. 17.

3. Реализация Grid-вычислений в среде IARnet / С.В. Емельянов [и др.] // Информационные технологии и вычислительные системы. – 2005. – №2. – С. 61–75.
4. Кривцов В. Е., Полков Е.Ю. Схема распределенного решения системы дифференциальных уравнений // Проблемы вычислений в распределенной среде: прикладные задачи // Тр. Ин-та системного анализа Российской академии наук (ИСА РАН). – М.: УРСС, 2004.
5. Dennis J. Data Flow Supercomputers // Computer. – 1980. – Vol.13, № 11. – P.48–56.



УДК 338

К.П. Шапоров, А.Ф. Крюков

## ИНФОРМАЦИОННОЕ ОБЩЕСТВО КАК ФАКТОР ЕГО СОЦИАЛЬНО-ЭКОНОМИЧЕСКОГО РАЗВИТИЯ\*

В статье рассматриваются предпосылки единой комплексной информационно-телекоммуникационной отрасли.

Реализация российских перспектив в области формирования рынка информационных услуг для устойчивого развития информационного общества повышает качество жизни граждан, обеспечивает конкурентоспособность России, развитие экономической жизни информационного общества.

**Ключевые слова:** информатизация, общество, рынок, инфраструктура, портал услуг, телекоммуникационные технологии.

*K.P. Shaporov, A.F. Krukov*

## SOCIETY INFORMATIZATION AS FACTOR OF ITS SOCIAL AND ECONOMIC DEVELOPMENT

*Preconditions for uniform complex information and telecommunication branch are considered in the article.*

*Russian prospects implementation in the field of information services market formation for the information society sustainable development increases citizens' life quality, provides Russia competitiveness, and information society economic life development.*

**Key words:** *informatization, society, market, infrastructure, portal of services, telecommunication technologies.*

---

Информатизация общества в XXI веке приобрела глобальный характер. Она проникла во все сферы жизни и профессиональной деятельности человека: в экономику, науку, образование, культуру, здравоохранение, бытовую сферу. Информатизация приводит к комплексным изменениям в сфере услуг. Эти преобразования человеческой жизни обычно называют информационной революцией. Она базируется на широком использовании информационных технологий для решения потребителями возникающих нужд с помощью современных электронных средств, связанных между собой программными продуктами телекоммуникационных сетей. Революционное и устойчивое расширение сфер воздействия современных достижений в области развития электронных телекоммуникационных сетей обеспечивает становление информационного общества с формированием рынка информационных услуг.

Анализ проблем формирования и сам рынок информационных услуг для развития информационного общества становится стабильным в течение последних пятнадцати лет. Подтверждением этого является повышенное внимание международного сообщества к вопросам информатизации. В конце 2003 года лидеры 150 стран мира собрались в Женеве на Всемирный саммит по рождающемуся информационному обще-

---

\* Исследование осуществлено при поддержке Федеральной целевой программы «Научные и научно-педагогические кадры инновационной России» на 2009-2013 гг. (госконтракт № 02.740.11.0585.)