

2. История развития распределенных вычислений

Проблема выбора между централизованной и распределенной моделями предоставления вычислительных ресурсов является одной из ключевых проблем организации вычислительных систем. Примером этого может являться статья «Централизованные и децентрализованные вычислительные системы: организационные соображения и варианты управления», опубликованная еще 1983 г. [43].

С того момента, как появилась возможность соединять компьютеры между собой, многие группы исследователей занялись изучением возможностей, предоставляемых системами распределенных вычислений, создавая множество библиотек и промежуточного программного обеспечения. Разработанное ППО предназначалось для того, чтобы обеспечить организацию и управление географически-распределенными ресурсами таким образом, чтобы они представляли собой единую виртуальную параллельную вычислительную машину.

До середины 70-х годов прошлого века по причине высокой стоимости телекоммуникационного оборудования и относительно слабой мощности вычислительных систем доминировала централизованная модель. В конце 70-х годов появление систем разделения времени и удаленных терминалов, явилось предпосылкой возникновения *клиент-серверной архитектуры*, обеспечивающей предоставление ресурсов мейнфреймов (больших вычислительных машин) конечным пользователям посредством удаленного соединения. Дальнейшее развитие телекоммуникационных систем и появление персональных компьютеров дало толчок развитию клиент-серверной парадигмы обработки данных. На последующих этапах произошло уточнение и переосмысление задачи распределенных вычислений.

В 2000-м году Ян Фостер¹ определил задачу распределенных вычислительных систем как «гибкое, безопасное, координированное распределение ресурсов среди динамических наборов пользователей, организаций и ресурсов» [31]. Он предложил называть такие распределенные вычислительные системы термином *грид* (более формальное определение грид будет дано в главе 11 «Технологии Грид»). Коммерческие разработчики, развивая идею грид, в 2007-2008 годах представили концепцию «облачных вычислений», в основе ко-

¹ Ян Фостер – директор института вычислений, старший научный сотрудник Арагонской национальной лаборатории (США). Является автором множества научных работ, посвященных параллельным и распределенным вычислениям. Руководил разработкой нескольких международных распределенных вычислительных систем. Автор концепции грид.

торой было предоставление высокомасштабируемых виртуальных вычислительных ресурсов конечному пользователю через интернет в виде услуг.

На текущий момент, использование распределенных вычислений в виде технологий грид и облачных вычислений набирает обороты. Они применяются для решения различных задач, их использование становится все проще. Распределенные вычисления становятся неотъемлемой частью научных и коммерческих высокопроизводительных вычислений.

2.1 Первое поколение систем распределенных вычислений

Первые проекты по распределенным вычислениям, появившиеся в начале 1990-х годов, основывались на объединении вычислительных возможностей суперкомпьютеров. Основной целью данных проектов было предоставление вычислительных ресурсов для определенного ряда высокопроизводительных приложений. В качестве типичных проектов того времени можно рассмотреть проекты FAFNER [34] и I-WAY [61]. Эти проекты стали базовыми, для всей отрасли распределенных вычислений в дальнейшем. На них основывались первые попытки стандартизации распределенных вычислений в гетерогенных вычислительных средах.

2.1.1 Проект FAFNER

Проект FAFNER был создан для решения задачи разложения больших чисел на основе мощностей географически-распределенных вычислительных систем. Нахождение простых множителей больших чисел является позволяет расшифровать данные, зашифрованные на основе алгоритма RSA.

Для шифрования секретной информации широко используется метод кодирования, основанный на публичном ключе RSA (аббревиатура из первых букв фамилий разработчиков данного метода: Rivest, Shamir и Adleman). Метод работы данного ключа основан на том, что разложение на множители больших чисел (сто и более знаков) – чрезвычайно сложная вычислительная задача. В марте 1991 корпорация RSA Data Security основала конкурс по поиску и реализации методов разложения больших чисел на множители. Это состязание обеспечило создание крупнейшей библиотеки по методикам поиска простых множителей от крупнейших ученых со всего земного шара.

Все алгоритмы поиска простых множителей, известные на сегодняшний день, требуют чрезвычайно большого количества вычислений (поэтому этот метод и используется для шифрования). Но особенность параллельной реализации этих алгоритмов состоит в том, что процессы поиска делителей вычисли-

тельно независимы, и не требуют обмена информации во время расчета. Первые попытки реализовать подобный алгоритм на распределенных вычислительных системах основывались на обмене электронными письмами.

В 1995 г. консорциумом организаций в области информационных технологий был запущен проект FAFNER – Factoring via Network-Enabled Recursion (Сетевое разложение на множители посредством рекурсии) по решению задачи разложения больших чисел посредством Веб-серверов.

Можно выделить следующие особенности, отличавшие этот проект от многих других [34]:

- реализация NFS – Network File System (Сетевая Файловая Система) позволяла даже малым рабочим станциям (с 4 Мб оперативной памяти) выполнять полезную работу, рассчитывая свой маленький фрагмент задачи;
- проект FAFNER поддерживал анонимную регистрацию участников. Пользователи могли поделиться своими вычислительными ресурсами без боязни раскрытия своей личной информации;
- консорциум сайтов, представлявших костяк вычислительной системы, формировали иерархическую структуру веб-серверов, что уменьшало возможность возникновения «узкого места» в вычислительной системе.

Данная система доказала свою надежность и качественность, заняв первое место по производительности в конкурсе, проводимом в рамках конференции Supercomputing'95.

2.1.2 Проект I-WAY

I-WAY – Information Wide Area Year (Год Информации Глобальных Сетей)

– экспериментальная высокопроизводительная сеть, которая объединила множество высокопроизводительных компьютеров и передовые средства визуализации в США. Она была спроектирована в начале 1995, с целью объединения высокоскоростных сетей, существующих на тот момент. Данные и компьютерные ресурсы были распределены по 17-и локациям в США и объединены 10-ю сетями, с различной пропускной способностью, различными протоколами соединения и с использованием различных сетевых технологий для их построения [23].

В рамках проекта, была построена аппаратная инфраструктура, посредством которой осуществлялся доступ к ресурсам сети I-WAY. Она состояла из базовых рабочих станций под управлением операционной системы UNIX, на которые было установлено специальное ППО (сервер I-POP). Система I-POP брала на себя функции шлюза к ресурсам сети I-WAY. Каждый такой сервер

поддерживал стандартные процедуры аутентификации, резервирования ресурсов создания процессов и коммуникации.

Также, в рамках данного проекта был разработан планировщик ресурсов, известный как «Брокер Вычислительных Ресурсов» (CRB – Computing Resources Broker). Он обеспечивал выполнения функций управления очередями задач, распределения заданий между компьютерами и слежения за ходом решения.

Для поддержки пользовательских приложений была адаптирована библиотека передачи данных Nexus. В нее были включены механизмы, поддерживающие автоматическое конфигурирование работы пользовательского приложения, в зависимости от методов передачи данных, базовой операционной системы и т. п.

Проект I-WAY использовался для решения следующих задач [23]:

- суперкомпьютерные вычисления;
- доступ к удаленным ресурсам;
- задачи виртуальной реальности.

2.2 Второе поколение систем распределенных вычислений

2.2.1 Грид

Проекты высокопроизводительных вычислительных систем, реализованные в начале 1990-х годов, выявили основные проблемы, которые необходимо было решить для развертывания стабильных распределенных вычислительных систем. Для решения всех указанных задач была разработана концепция *grid* – РВС, обеспечивающей «гибкое, безопасное, координированное распределение ресурсов среди динамических наборов пользователей, организаций и ресурсов». Основной задачей грид было построение инфраструктуры, обеспечивающей «*вычисления по требованию*» (*utility computing*), аналогично тому, как мы сейчас получаем доступ к коммунальным услугам, например, к электричеству. В конце 1990-х, в результате исследований систем распределенных вычислений, Ян Фостер вывел три основных требования, которым должны соответствовать РВС для того чтобы называться грид [27].

1. *Гетерогенность*. Вычислительная среда грид может состоять из множества различных ресурсов, обладающих различными характеристиками и параметрами.
2. *Масштабируемость*. Грид может состоять из сколь угодно большого числа ресурсов. И если поставленная задача решается на 10-и компьютерах за определенное время, существует опасность, что на 100 компьютерах она

будет решаться в 100 раз медленнее, в связи с многократным возрастанием расходов на передачу данных между узлами.

3. *Приспособляемость.* При работе с грид-системами необходимо учитывать, что ошибки при работе с ресурсами – это не исключение, а правило. Среда грид может состоять из сотен компьютеров, и ошибки в работе десятка из них не должны влиять на полученные результаты решения.

Дальнейшее развитие грид происходило в рамках определенных Фостером требований. Применение технологии грид позволило объединить крупные комплексы обработки и хранения данных, обеспечивая их доступность для различных пользователей, включая правительственные и научные организации. Для реализации предложенных концепций в 1997 году был запущен проект Globus [20], направленный на разработку и предоставление инфраструктуры для грид-вычислений. Первоначально, Globus был развитием проекта I-WAY, который мы рассмотрели выше. Но в процессе эволюции проекта Globus, основной акцент разработчиков был перенесен с поддержки высокопроизводительных вычислений в сторону сервисов поддержки *виртуальных организаций* и предоставление возможности приложениям работать с распределенными разнородными вычислительными ресурсами как с единой виртуальной машиной. Более подробно о конкретной технологии грид и особенностях их разработки мы поговорим в главе 11 «Технологии Грид».

2.2.2 Проект Legion

Проект Legion – это объектно-ориентированная система, предоставляющая программную оболочку для организации однородного взаимодействия гетерогенных распределенных высокопроизводительных вычислительных систем [35]. Первая реализация системы была выпущена в ноябре 1997-го. Основной целью проекта было предоставление пользователям единой интегрированной инфраструктуры РВС, независимо от масштаба, географического положения, языка или операционной системы. В отличие от Globus, система Legion основывалась на объектно-ориентированном подходе, включая обязательную поддержку абстракций данных, инкапсуляции, наследования и полиморфизма.

Legion предоставлял пользователю набор объектов, предоставляющих базовые сервисы:

- *объекты вычислителей* – абстракции, реализующие базовые принципы работы с вычислительными ресурсами;
- *объекты систем хранения данных* – абстракции, предоставляющие базовые методы работы с системами хранения данных;

- *объекты связывания* – объекты, обеспечивающие связи между абстрактным идентификатором объекта и его физическим адресом;
- *объекты контекста* – объекты, реализующие проекцию пользовательских имен объектов на абстрактные идентификаторы объектов в системе Legion.

В течение года система развилась настолько, что было принято решение о коммерциализации проекта, и в августе 1998 г компания Applied Metacomputing выпустила коммерческую версию проекта. Проект активно разрабатывался вплоть до 2001 года, после чего развитие проекта остановилось.

2.2.3 Распределенные объектные системы

При рассмотрении второго поколения распределенных вычислительных систем, нельзя не затронуть такой класс, как распределенные объектные системы. Они предоставляют базовые методы для регистрации, сериализации и десериализации объектов обеспечивая удаленный вызов методов.

В середине 1990-х годов одним из наиболее распространенных методов построения распределенных объектных систем являлась архитектура *CORBA* (*Common Object Request Broker Architecture* — общая архитектура брокера объектных запросов) [37]. В 1997-1998 годах консорциум OMG (Object Management Group) опубликовал вторую версию спецификации стандарта CORBA, который обеспечивал стандартный протокол взаимодействия объектно-ориентированных систем. Вместе со стандартом были выпущены отображения для наиболее распространенных ОО-языков: C++ и Java. В состав решения CORBA было включено множество сервисов, которые можно было применить к отраслям электронной коммерции и науки. Также, на базе CORBA возможно было реализовать концептуальные модели архитектуры распределенной вычислительной среды, т.к. она обеспечивала взаимосвязь с языком UML (Unified Modeling Language – Унифицированный Язык Моделирования). Продержавшись на рынке около 5 лет, популярность CORBA пошла на спад. Частично это объясняют сложностью процесса разработки на основе объектно-ориентированной модели, предлагаемой CORBA, частично – высокой стоимостью приобретения и поддержки таких систем. Также, многие считают, что CORBA ушла в тень, в связи с тем, что не соответствовала стандартам Веб-систем, мода на которые пришла в начале 2000-х. Более подробно роли Веб в области распределенных вычислительных систем будет рассказано в главе 3 «Веб».

Также следует упомянуть о модели Java. Тогда как, для борьбы с гетерогенностью вычислительной среды CORBA реализует высокоуровневые стан-

дарты взаимодействия, модель Java использует единую виртуальную среду. В определенной степени JVM (Java Virtual Machine – Виртуальная Машина Java), в совокупности с Java-приложениями и сервисами, преодолевают проблемы гетерогенности вычислительных систем, предоставляя методы удаленного вызова процедур посредством технологии RMI (Remote Method Invocation – Удаленный Вызов Методов).

Более подробно об архитектуре CORBA и методах работы с удаленными объектами мы поговорим в главе 5 «Объектные распределенные системы».

2.3 Современные РВС

На сегодняшний день, РВС отходят от традиционных понятий высокопроизводительных распределенных вычислений в сторону развития виртуального сотрудничества и виртуальных организаций. *Виртуальная организация* – это ряд людей и/или организаций, объединенных общими правилами коллективного доступа к определенным вычислительным ресурсам [31]. Методы предоставления доступа к вычислительным ресурсам становятся сервисно-ориентированными, что позволяет гибко использовать одни и те же вычислительные ресурсы различными потребителями.

Значительно расширились области автоматизированного управления ресурсами. Человек не в силах вручную решить задачу распределения вычислений в системах такого масштаба и гетерогенности. Таким образом, необходимо использование автоматизированных систем управления задачами, которые берут на себя задачи управления предоставляемой системой. Также, с возрастанием масштаба вычислительных сетей, необходимы автоматизированные средства обработки ошибок и восстановления вычислительного процесса.

2.3.1 Одноранговые (peer-to-peer) сети

В 1999 году, в Северо-восточном Университете (Массачусетс, США) первокурсник Шон Фэннинг написал систему обмена MP3 файлами между пользователями. Этот проект получил название Napster. Он стал первым проектом, положившим начало технологии одноранговых (Peer-to-peer (P2P) – англ. «равный-к-равному») распределенных вычислительных сетей. Через 2 года Napster был закрыт совместными усилиями владельцев авторских прав на музыкальные произведения, распространявшиеся через эту сеть. Но по примеру Napster развился целый класс P2P систем нового, децентрализованного типа, которые закрыть было значительно сложнее.

В 2000 году Джастин Франкел (20-ти летний хакер из США, в 1997 году выпустил бесплатный MP3 плеер WinAmp) написал Gnutella – P2P протокол передачи файлов. В отличие от Napster, который использовал центральный сервер для установления связи между пирами, Gnutella полагалась исключительно на системы конечных пользователей для организации сети. Таким образом, в отличие от Napster, сеть Gnutella оказалось невозможно «закрыть», отключив центральный сервер. Миллионы человек до сих пор пользуются данной системой.

При работе в рамках парадигмы P2P, компьютеры обмениваются ресурсами непосредственно друг с другом, без использования центрального сервера. Подход P2P обеспечивает решение проблем, возникших в результате экспоненциального роста интернет и Веб. Применение P2P позволило множеству пользователей, которые раньше были простыми потребителями информации, участвовать в предоставлении контента. В момент своего появления, P2P было скорее модным словом, чем продуманной концепцией. В результате мощного продвижения с помощью средств массовой информации, технология P2P распространилась в академических и промышленных кругах.

В рамках концепции P2P все входящие в сеть компьютеры взаимодействуют друг с другом напрямую, без использования централизованных серверов. Основные достоинства одноранговых вычислительных систем:

- упрощается поддержка масштабируемости при значительном росте количества узлов в вычислительной сети;
- повышается отказоустойчивость сети, т.к. сбой любого вычислительного узла не может привести к остановке функционирования сети целиком.

Тем не менее, существует ряд препятствий при построении P2P сетей:

1. При работе с P2P приложениями, вычислительный узел берет на себя функции, как клиента, так и сервера. Это приводит к *увеличению требований к производительности* каждого компьютера, включенного в такую сеть.
2. *Низкая степень защищенности машин*, участвующих в P2P сети объясняется тем, что они предоставляют открытый доступ к своим ресурсам (таким как такты процессора, определенные папки на жестком диске и т. п.). Таким образом, при отсутствии средств защиты, компьютеры, включенные в P2P подвержены риску взлома или заражения со стороны недобросовестных участников.

3. При построении P2P сети приходится преодолевать возможную *гетерогенность аппаратного и программного обеспечения* ее потенциальных участников. Этот вопрос может быть решен путем применения таких технологий как XML или Java.
4. Основная проблема P2P сетей – это *поиск доступных ресурсов*, без использования централизованной точки управления. Каждому узлу приходится производить поиск среди сотен и тысяч ресурсов внутри сети, что является очень трудоемкой и ресурсоемкой задачей.

Шумиха и споры о легальности P2P привели к недоверию к этой технологии, хотя на самом деле, она обладает существенным потенциалом. В настоящее время P2P – это общепринятая концепция построения распределенных высокомасштабируемых вычислительных систем. Несмотря на все трудности, происходит бурное развитие и использование P2P сетей. Среди наиболее известных и значимых примеров, стоит отметить такие проекты как BitTorrent, Napster, Skype, SETI@home (хотя принадлежность SETI@home к роду P2P систем вызывает серьезные споры среди разработчиков распределенных приложений).

Несмотря на то, что базовая философская концепция грид и P2P различается, обе технологии пытаются решить одну и ту же проблему – создание виртуального слоя [69] над существующей инфраструктурой Интернет для обеспечения совместной работы и использования совместных ресурсов [24]. Тем не менее, в реализации, подходы сильно отличаются. Грид скорее ориентирована на объединение виртуальных организаций для обеспечения совместной работы их участников, а P2P связывает отдельных пользователей находящихся за конечными узлами сети интернет (то есть за NAT, брандмауэрами и др.). Более подробно о концепции P2P вычислений и будет рассказано в главе 10 «Технологии одноранговых сетей».

2.3.2 Сервис-ориентированная архитектура

В начале 2000-х годов бизнес-сообщество занялось разработкой следующего поколения спецификаций, призванных решить проблемы ранних стандартов распределенных объектных технологий посредством *веб-сервисов* и *сервис-ориентированной архитектуры* (*Service-Oriented Architecture – SOA*).

Стандарты веб-сервисов были разработаны по инициативе организаций, занимающихся предоставлением удаленного доступа к определенным вычислительным ресурсам, и закреплены консорциумом W3C. К основным стандартам разработки и функционирования веб-сервисов можно отнести:

- SOAP – основанный на XML протокол взаимодействия веб-сервисов;
- WSDL (Web Services Description Language – Язык описания веб-сервисов) – это методология описания ресурсов, предоставляемых веб-сервисом;
- UDDI (Universal Description Discovery and Integration – Универсальный метод поиска и интеграции) – метод описания, поиска, взаимодействия и использования веб-сервисов.

На сегодняшний день, сервис-ориентированный подход является стандартом «де-факто» при разработке распределенных вычислительных систем. Более подробно о сервис-ориентированных системах и технологиях веб-сервисов будет рассказано в главе 8 «Сервис-ориентированная архитектура» и главе 9 «Веб-сервисы».

2.3.3 *Агенты*

Несмотря на все преимущества технологии веб-сервисов, они не предоставляют новых методологий и решений построения широкомасштабных вычислительных сетей. Для поиска решений в этом направлении, необходимо рассмотреть агентно-ориентированную парадигму построения РВС.

Вычислительные сети на основе так называемых *агентов* – это принципиально иной подход к организации РВС. *Программный агент* – это автономный процесс, способный реагировать на среду исполнения и вызывать изменения в среде исполнения, возможно, в кооперации с пользователями или другими агентами. Рассмотрим основные принципы работы агентных сетей [40]:

- *автономность* – агенты функционируют автономно, без возможности постороннего вмешательства в их внутреннее состояние;
- *социальное поведение* – агенты взаимодействуют друг с другом посредством определенного языка;
- *активность* – агенты взаимодействуют с окружающей средой, получая определенные сигналы и отвечая на них;
- *про-активность* – агенты действуют целенаправленно.

Агентные сети принципиально приспособлены для функционирования в динамически-изменяемой окружающей среде. В этом случае, автономность агентов позволяет организовать динамическую подстройку вычислительного алгоритма под условия вычислительной среды. Таким образом, РВС можно представить как набор взаимодействующих компонентов, а информация, которой они обмениваются, разбивается на определенные категории:

- информация о компонентах и их функциональных возможностях, в рамках определенной области;
- информация о взаимодействиях между компонентами;
- обобщенная информация о рабочем процессе и более конкретная информация по той или иной задаче.

Для обеспечения функционирования такой системы, необходима стандартизация методов взаимодействия между компонентами. Для решения этой задачи разрабатываются и стандартизируются языки взаимодействия агентов (Agent Communication Languages, ACLs). Одним из наиболее известных, является архитектура взаимодействия FIPA (Foundation for Intelligent Physical Agents, базис интеллектуальных физических агентов). Эта архитектура стандартизует методы взаимодействия агентов и агентных систем.

Более подробно, принципы организации агентных систем будут рассмотрены в главе 6 «Агентные технологии».

2.3.4 Облачные вычисления

Облако – это парадигма крупномасштабных распределенных вычислений, основанная на эффекте масштаба, в рамках которой пул абстрактных, виртуализованных, динамически-масштабируемых вычислительных ресурсов, ресурсов хранения, платформ и сервисов предоставляется по запросу внешним пользователям через Интернет [29].

Не смотря на то, что метафора «облако» уже давно используется специалистами в области сетевых технологий для изображения на сетевых диаграммах сложной вычислительной инфраструктуры (или же Интернета как такового), скрывающей свою внутреннюю организацию за определенным интерфейсом, термин «Облачные вычисления» появился на свет совсем недавно. Согласно результатам анализа поисковой системы Google, термин «Облачные вычисления» («Cloud Computing») начал набирать вес в конце 2007 – начале 2008 года, постепенно вытесняя популярное в то время словосочетание «Грид-вычисления» («Grid Computing»). Судя по заголовкам новостей того времени, одной из первых компаний, давших миру данный термин, стала компания IBM, развернувшая в начале 2008 года проект «Blue Cloud» и ставшая спонсором Европейского проекта «Joint Research Initiative for Cloud Computing».

На сегодняшний день уже можно говорить о том, что облачные вычисления прочно вошли в повседневную жизнь каждого пользователя Интернета (хотя многие об этом и не подозревают). Однако до сих пор нет единого мнения о

том, что такое «Облачные Вычисления» и каким образом они соотносятся с парадигмой «Грид-вычислений».

Более подробно, принципы организации агентных систем будут рассмотрены в главе 12 «Облачные вычисления».

2.4 Заключение

Распределенные вычислительные системы – это сформировавшаяся сфера высокопроизводительных вычислений, обладающая своей спецификой, ярко выраженным классом решаемых задач и методами их решения. Разрабатываются и внедряются новые концепции построения распределенных систем, расширяется круг решаемых ими задач, упрощается процесс организации, разрабатываются более простые методы использования ресурсов конечными пользователями.