

# О построении распределенной информационной системы

Д.В. Беляков, И.М. Гостев, Н.А. Давыдова, П.В. Зрелов, В.В. Иванов,  
Л.А. Калмыкова, Т.С. Сыресина

*Лаборатория информационных технологий, ОИЯИ*

## Abstract

Using a classic scheme of constructing information systems basing on a multilink “client-server” technology often leads to overload of some computers, while others remain idle. A solution to this problem related to construction of distributed data bases will not be final. However, it should be noted there are technological solutions in the field of constructing the distributed systems based, for instance, on grid technologies (Globus, Condor, etc), where the rendering of some user service is yielded by a remote service, and it does not matter where that service is located.

Such a distributed infrastructure can be also used to arrange the services serving the clients, thus reducing the cost of the system due to decreasing requirements to some sites of the system and therefore the cost of the whole system reduces, too. Besides, reliability and fault tolerance of the system essentially increases, because of backup functions of separate sites.

This work deals with the architecture of such systems in which the rendering of a user service of an information system is distributed between various sites of an electronic network within the technology suggested.

## 1 Введение

Наращивание вычислительной мощности аппаратных комплексов существенно опережает развитие программного обеспечения для них, в том числе и прикладного [1]. Такое положение дел характерно для всех крупных аппаратно-программных комплексов в мире, так как при их создании разработчики порой уделяют недостаточно внимания проблемам эксплуатации и окупаемости таких систем.

В ОИЯИ существует потребность решения не только задач теоретической и экспериментальной физики. Здесь, как и в любой организации, необходимо решать административно-хозяйственные, финансовые и иные задачи, связанные с её нормальным функционированием. В настоящее время для решения таких задач обычно применяют многозвенные информационные системы, имеющие один или два сервера баз данных, столько же серверов приложений и клиентские рабочие места [2-4]. Сейчас существует тенденция использовать в качестве клиентского приложения стандартный браузер, а услуги предоставлять через Web-интерфейс [5, 6]. Кроме того, существует тенденция управления различными производственными и административными процессами посредством распределенных приложений [7-9].

Вычислительные мощности создаваемого городского метакластерного комплекса “Дубна-Грид” можно использовать не только для расчета физических задач, но и для решения задач, необходимых как внутри организации, так и в городском хозяйстве и коммунальном управлении: при безбумажном делопроизводстве, в предоставлении справочных и аналитических материалов по жилищно-коммунальному и транспортному хозяйству, коммуникационным и инженерным сетям и т. п.

В существующей архитектуре комплекса практически отсутствуют сервисы для конечного пользователя, то есть для выполнения в ней каких-либо прикладных и информационно-аналитических расчетов. Клиенту необходимо самому создавать программное обеспечение, необходимое для решения таких задач. Часто это приводит к отказу от использования комплекса и вызывает переход к стандартному аппаратно-программному обеспечению, имеющему более дружественную среду и интуитивно понятный интерфейс.

Для решения указанных задач архитектура существующего программного обеспечения должна быть пересмотрена, в неё необходимо добавить некоторые новые конструктивные элементы и механизмы функционирования.

## 2 Предлагаемая архитектура

Основным недостатком существующих архитектур является механизм предоставления услуг, сосредоточенный на отдельных устройствах - серверах. К таким серверам относится и контроллер задач в системе Condor, распределяющий вычислительные задачи по машинам в сети, и сервера приложений в структурах типа Globus, Corba, Enterprise Java Server и т.п. Все эти и аналогичные структуры имеют свои брокеры, через которые происходит взаимодействие между клиентским программным приложением и сервером данных.

Исторически появление такого типа серверов приложений явилось большим шагом в развитии вычислительных архитектур, как в локальных сетях, так и в глобальной сети Интернет. Однако, сосредоточение места оказания всех услуг на одной машине (или даже двух-трех) с ростом числа клиентов приводит к чрезмерным требованиям к ним, в то время как загрузка остальных ресурсов остается низкой. Кроме того, существенно возрастает время ответа из-за увеличения длины очереди запросов. Существующие аппаратно-программные решения основаны именно на такой архитектуре и не позволяют равномерно загружать все компьютеры распределенной среды при решении административно-хозяйственных задач.

Поэтому в настоящее время именно многозвенная архитектура клиент-сервер становится тормозом в развитии распределенных вычислительных систем. Именно в ней заложена неравномерная загрузка любого аппаратно-программного вычислительного комплекса. Для решения задачи по равномерной загрузке всех компьютеров распределенной вычислительной среды необходимо пересмотреть механизм загрузки вычислительной системы, то есть пересмотреть механизм функционирования серверов приложений.

Возможным решением, удовлетворяющим поставленной задаче, является архитектура, основанная на виртуальных серверах услуг и виртуальных процессах оказания услуг. Рассмотрим схему предлагаемого механизма функционирования такой системы, представленную на рис. 1.

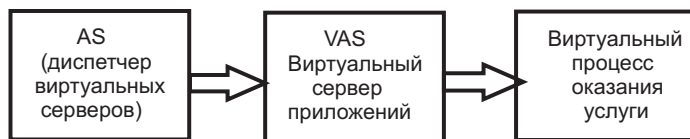


Рис. 1: Механизм получения услуги

В нем для доступа к некоторой услуге существует некоторый сервер приложений - AS (механизм доступа к серверу не имеет значения и может быть HTTP, Java application, C++ Stubs, и т. п.). При установлении соединения с клиентом и получении запроса на обслуживание (запрос на оказание услуги) в такой распределенной среде происходит создание некоторого виртуального сервера (VAS), как некоторого процесса на отдельном компьютере. Особенностью этого процесса будет его предназначение только для оказания услуг определенного типа для конкретного клиента. Для выполнения самой услуги необходимо создать виртуальный процесс, размещаемый еще на одном компьютере, который эту услугу будет предоставлять соответствующему клиенту. По окончании предоставления услуги процесс прекращается. По завершению работы клиента прекращается процесс виртуального сервера приложений. Если клиенту необходимо оказывать несколько услуг в одно и то же время, создается столько виртуальных процессов, сколько требуется услуг. Каждый такой процесс будет размещен на отдельном компьютере распределенной системы.

Например, если некоторому клиенту необходимо запустить на счет трудоемкую задачу, то для оказания такой услуги может быть запущен процесс, взаимодействующий с сервером типа Condor и осуществляющий выполнение заданий в распределенной среде. Результат будет представлен пользователю через этот виртуальный процесс. В другом случае может оказаться полезным формирование виртуального сервера, обеспечивающего получение информации из сети Интернет (например, о расписании транспорта). При этом будет создан виртуальный процесс, предназначенный для отправки запроса по определенному адресу (IP), получения информации и предоставления её пользователю в определенном формате. В третьем случае пользователю необходима, например, информация о нахождении силового электрического кабеля в конкретном месте г. Дубны. Для получения такой информации будет загружен соответствующий виртуальный сервер приложений, обеспечивающий работу с двумя базами данных - картографической и технологической Мособлэнерго по г. Дубна. При получении результата процесс, созданный для реализации этого запроса, будет окончен, а управление передано на виртуальный сервер приложений, обслуживающий данного клиента. Если этот клиент желает получить информацию о наличии в этом месте еще и водопроводных сетей<sup>1</sup>, то произойдут следующие действия: работающий виртуальный сервер приложений создаст новый виртуальный процесс для связи с базой данных Мособлводоканала, получит оттуда необходимую информацию, преобразует ее в соответствующую форму и предоставит клиенту. По окончании выполнения запроса процесс заканчивает свою работу.

В такой системе, как и в любой другой многозвенной архитектуре клиент-сервер, решающим фактором быстродействия является время установления соединения с базой данных и выполнения запроса. Известно, что наиболее затратным по времени механизмом в работе многозвенных систем является установление связи с базами данных, которые требуют аутентификации и некоторых других действий при подключении к СУБД [2]. Поэтому для обеспечения быстродействия всей системы необходимо создавать виртуальные процессы точек доступа к распределенной базе данных.

Механизм функционирования такого доступа будет описываться следующим об-

---

<sup>1</sup>При наличии соответствующих полномочий. Далее эти полномочия подразумеваются, но механизм их обеспечения не рассматривается. Вопросы безопасности будут рассмотрены в отдельной работе.

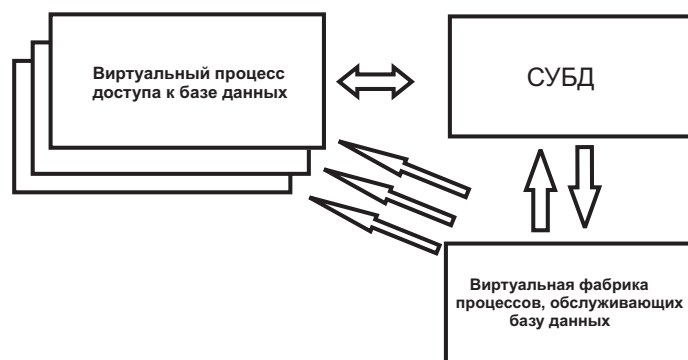


Рис. 2: Архитектура доступа к СУБД

разом. Пусть имеется распределенная база данных<sup>2</sup>. Для обслуживания запросов к ней имеется виртуальная фабрика процессов, то есть процесс, порождающий процессы, которые будут зарегистрированы и связаны с базой, но пассивны по отношению к ней. В случае порождения некоторого виртуального процесса, реализующего услугу клиенту (см. рис. 1), он устанавливает соединение через фабрику процессов с порожденным ею виртуальным процессом доступа к базе данных. При этом подразумевается, что регистрация в базе уже выполнена и СУБД готова к выполнению запроса. Результат выполнения запроса передается виртуальному процессу оказания услуги. По окончании работы последнего уничтожается и процесс доступа к базе данных. Фабрика виртуальных процессов при этом создает новый процесс, который регистрируется в базе данных и ждет соответствующего запроса.

Таким образом, опуская некоторые подробности предлагаемой архитектуры, можно сказать, что реализация предоставления услуги клиенту будет происходить в нескольких разделенных процессах, каждый из которых максимально оптимизирован для его исполнения. При этом достигается равномерная загрузка всей вычислительной системы.

Важным аспектом такой архитектуры является тот факт, что она может работать на существующей аппаратно-программной платформе.

### 3 Особенности предлагаемой архитектуры

При реализации предлагаемой системы возможно получение следующих преимуществ:

- Существенное уменьшение времени задержки на получение ответа на запрос клиента, так как в такой системе практически исключено появление очереди на обслуживание.
- Равномерную загрузку всей вычислительной системы, включая сетевые и серверные ресурсы.
- Высокую отказоустойчивость системы, так как ее работа не зависит от одного или нескольких компьютеров.
- Снижение требований к отдельным объектам вычислительной среды: серверы приложений, серверы баз данных, коммуникационное оборудование и т.п.

<sup>2</sup>Не имеет значение ни ее архитектура, ни предметная направленность, ни способ хранения информации.

- Высокая масштабируемость системы, позволяющая наращивать мощность при необходимости.
- Низкая себестоимость отдельных элементов и высокая их взаимозаменяемость.
- Территориальная распределенность системы означает высокую защищенность от повреждений и нарушений в работе.
- Единый механизм доступа ко всем ресурсам информационной системы, включая единый платформу-независимый Web-интерфейс (унификация и стандартизация).
- Возможность организации безбумажного документооборота на любом уровне управления ОИЯИ.
- Получение услуг в любой точке земного шара.

Однако, такая архитектура имеет и некоторые недостатки. К ним необходимо отнести повышенную сложность разработки по сравнению с традиционными архитектурами.

## 4 Выводы

Предлагаемая технология позволяет оптимизировать нагрузку на распределенные вычислительные комплексы и повысить их эффективность за счет решения на них административно-управленческих задач. Введение распределенных вычислений на такой архитектуре существенно повышает надежность функционирования всей системы.

Кроме того, предлагаемая архитектура обладает универсальностью и может быть использована при решении широкого круга прикладных задач в распределенной вычислительной среде.

## Список литературы

- [1] Воеводин В.В., Воеводин Вл.В. “Параллельные вычисления” - СПб.: БХВ-Петербург, 2002.
- [2] Гринвальд Р., Стаковьяк Р. и др. Программирование баз данных Oracle для профессионалов. М.: “Диалектика” 2007.
- [3] Пуха Ю. Объектные технологии построения распределенных информационных систем. Jet Info 16(47) 1997.
- [4] Столяров М., Трифаленков И. Распределенная информационная бизнес-среда, или пять Больших Проблем Администратора. Jet Info 3(70) 1999.
- [5] Шрайберг Я.Л., Воройский Ф.С. Автоматизированные библиотечно-информационные системы. М.: Либерея: ГПНТБ России, 1996. 273 с.
- [6] Буравлев В.А., Сережников С.В., Майоров А.А. и др. Горизонтальные информационные системы для органов государственной власти на платформе TrisoftDDS. Информационные технологии, №6, 2003 г.
- [7] Тимофеев А.В. Проблемы и методы адаптивного управления потоками данных в телекоммуникационных системах. Информатизация и связь. 2003. № 1–2. с. 68–73.
- [8] Syrtzev A.V., Timofeev A.V. Neural and Multi-Agent Routing in Telecommunication Networks. International Journal Information Theories and Their Applications. 2003. Vol.10, № 2. p. 167–172.
- [9] Timofeev A.V. Multi-Agent Information Processing and Adaptive Control in Global Telecommunication and Computer Networks. International Journal Information Theories and Their Applications. 2003. Vol. 10, № 1.p. 54–60.