

УДК 004.424

# ПРИМЕНЕНИЕ ГРИД-СИСТЕМ ПРИ РАЗВЕРТЫВАНИИ WEB-САЙТА

**А. М. Токарчук<sup>1</sup>,**

аспирант

Московский государственный университет путей сообщений

Рассматривается построение отказоустойчивой распределенной грид-системы по обработке запросов к web-сайту.

**Ключевые слова** — грид, web-сайт, распределение нагрузки, отказоустойчивые системы.

## Введение

Очень часто создание web-сайта (под web-сайтом в данной статье подразумевается служба, которой может пользоваться как человек через GUI-интерфейс интернет-браузера, так и компьютер через набор API-вызовов, используя такие протоколы, как REST, SOAP) начинается с создания одной страницы в Интернете. Постепенно к ней добавляются другие. Происходит переход сайта от статичной архитектуры (при статичной архитектуре сайт представляет собой набор html-страниц) к использованию динамической системы управления контентом, сайтом (*Content Management System* — CMS) [1]. Сайт разрастается, и CMS уже не справляется с потоком запросов. Далее сайт, как правило, переносится из хостинг-компании (компания, предоставляющая услуги размещения сайтов на своих серверах) на собственный сервер, предоставляемый ему полностью, и уже он начинает обрабатывать http-запросы пользователей. Программисты с системными администраторами и инженерами развертывают необходимые для работы сайта службы.

Обычно это система управления базами данных (СУБД) (как правило, MySQL), web-сервер (например, Apache) и интерпретатор языка web-программирования (например, PHP). На некоторое время ситуация стабилизируется, СУБД обрабатывает запросы, выдает их интерпретатору, далее страница через web-сервер выдается пользователю. Но рано или поздно наступает ситуа-

ция, при которой нагрузка превышает допустимые значения для новой системы.

Следующим шагом является установка системы распределения нагрузки. Для этого используется «легкий» web-сервер (например, сервер Nginx разработки Игоря Сысоева или Light-Httpd).

## Выбор программного обеспечения

Первоначальная задача проектирования грид-системы состоит в выборе программного обеспечения (ПО) для установки на ее узлах. Программное обеспечение должно отвечать следующим требованиям:

- быть бесплатным. При построении грид-системы расходы планируются только на оборудование и оплату труда сотрудников, при самостоятельном построении — только на оборудование;
- иметь открытый исходный код для внесения в него при необходимости изменения;
- иметь развитое сообщество (community) для решения проблем, возникающих в процессе развертывания и эксплуатации системы;
- быть хорошо документировано.

## СУБД

В качестве сервера баз данных для построения грид-системы была выбрана СУБД MySQL, так как она является наиболее приспособленной для применения в среде web. Не секрет, что для исполнения приложений клиента на большинстве хостинг-площадок провайдеры предоставляют небольшое количество ресурсов (как вычислительных, так и дисковых). Поэтому для данного

<sup>1</sup> Научный руководитель — доктор технических наук, профессор кафедры вычислительных систем и сетей Московского государственного университета путей сообщений А. Б. Барский.

применения необходима высокоэффективная СУБД, обладающая при этом высокой надежностью (большинство web-приложений и сайтов должны работать в режиме 24/7).

По всем этим причинам MySQL стала незыблемым стандартом в области СУБД для web, а теперь в ней развиваются возможности для использования ее в любых критичных бизнес-приложениях, т. е. MySQL конкурирует на равных с СУБД таких производителей, как Oracle, IBM, Microsoft и Sybase.

Основные преимущества MySQL:

- многопоточность, поддержка нескольких одновременных запросов;
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- ODBC<sup>2</sup>-драйвер;
- гибкая система привилегий и паролей;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- интерфейс с языками C и Perl, PHP;
- быстрая работа, масштабируемость;
- совместимость с ANSI SQL;
- бесплатность в большинстве случаев;
- хорошая поддержка со стороны провайдеров услуг хостинга;
- быстрая поддержка транзакций через механизм InnoDB<sup>3</sup>.

## Web-сервер

В качестве web-сервера было выбрано ПО Apache. Основными достоинствами Apache считаются надежность и гибкость конфигурации. Оно позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

Apache имеет развитые механизмы обеспечения безопасности и разграничения доступа к данным. Основными являются:

- ограничение доступа к определенным директориям или файлам;
- механизм авторизации пользователей для доступа к директории по методу http-авторизации

<sup>2</sup>ODBC (Open DataBase Connectivity) — это программный интерфейс (API) доступа к базам данных, разработанный фирмой X/Open (по другим данным — Microsoft). Позволяет единообразно оперировать с разными источниками данных, отвлекаясь от особенностей взаимодействия в каждом конкретном случае.

<sup>3</sup>InnoDB — одна из выбираемых подсистем низкого уровня в СУБД MySQL, входит во все стандартные сборки для различных операционных систем. Основным отличием InnoDB от других подсистем низкого уровня MySQL является наличие механизма транзакций.

(mod\_auth\_basic) и digest-авторизации (mod\_auth\_digest);

- ограничение доступа к определенным директориям или всему серверу, основанное на IP-адресах пользователей;
- запрет доступа к определенным типам файлов для всех или части пользователей, например запрет доступа к конфигурационным файлам и файлам баз данных;
- наличие модулей, реализующих авторизацию через СУБД или PAM.

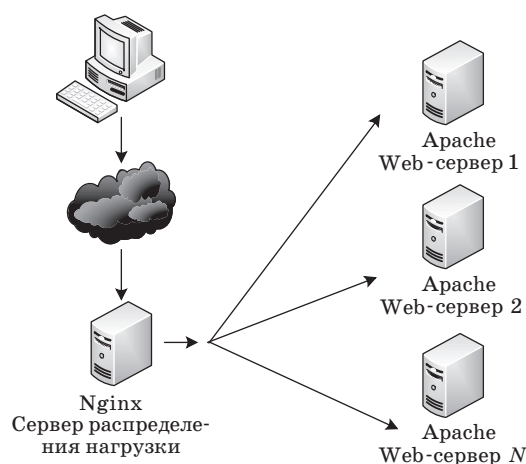
Преимущества Apache также заключаются в модульной организации, стабильности и быстрой работе. При наличии соответствующих аппаратных средств и настройке оно может выдерживать самые высокие загрузки. Yahoo, Altavista, GeoCities, Hotmail основаны на различных версиях этого сервера.

## Балансировщик нагрузки

В качестве сервера распределения нагрузки (балансировщика) для построения грид-системы был выбран Nginx (рис. 1) из-за большей по сравнению с Apache производительностью.

Основные преимущества http-сервера Nginx:

- обслуживание статических запросов, индексных файлов, автоматическое создание списка файлов, кэш-дескрипторов открытых файлов;
- акселерированное проксирование с кэшированием, простое распределение нагрузки и отказоустойчивость;
- акселерированная поддержка удаленных FastCGI-серверов с кэшированием, простое распределение нагрузки и отказоустойчивость;
- модульность, фильтры, в том числе сжатие (gzip), («докачка») byte-ranges, SSI-фильтр, преобразование изображений; несколько подзапросов на одной странице, обрабатываемых в SSI-



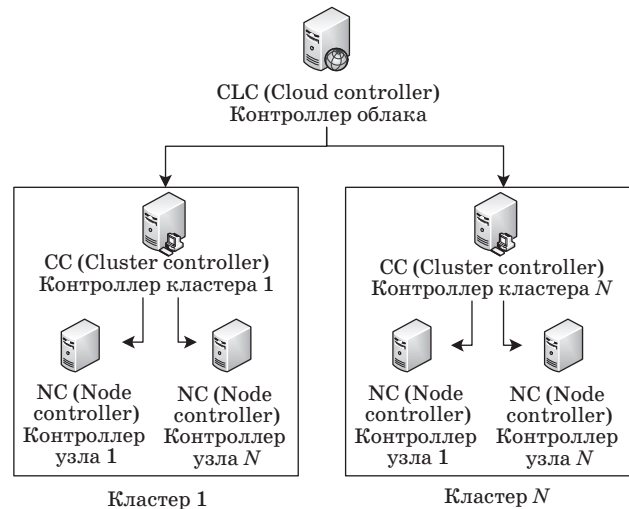
■ Рис. 1. Применение балансировщика нагрузки

филт্রে через прокси или FastCGI, выполняются параллельно;

- один главный процесс и несколько рабочих, рабочие процессы работают под непривилегированным пользователем;
- на 10 000 неактивных http-соединений расходуется около 2,5 МБ памяти;
- минимум операций копирования данных;
- выполнение разных функций в зависимости от адреса клиента;
- ограничение доступа в зависимости от адреса клиента и по паролю (Basic-аутентификация);
- ограничение скорости отдачи ответов;
- ограничение числа одновременных соединений и запросов.

### Кластерная система

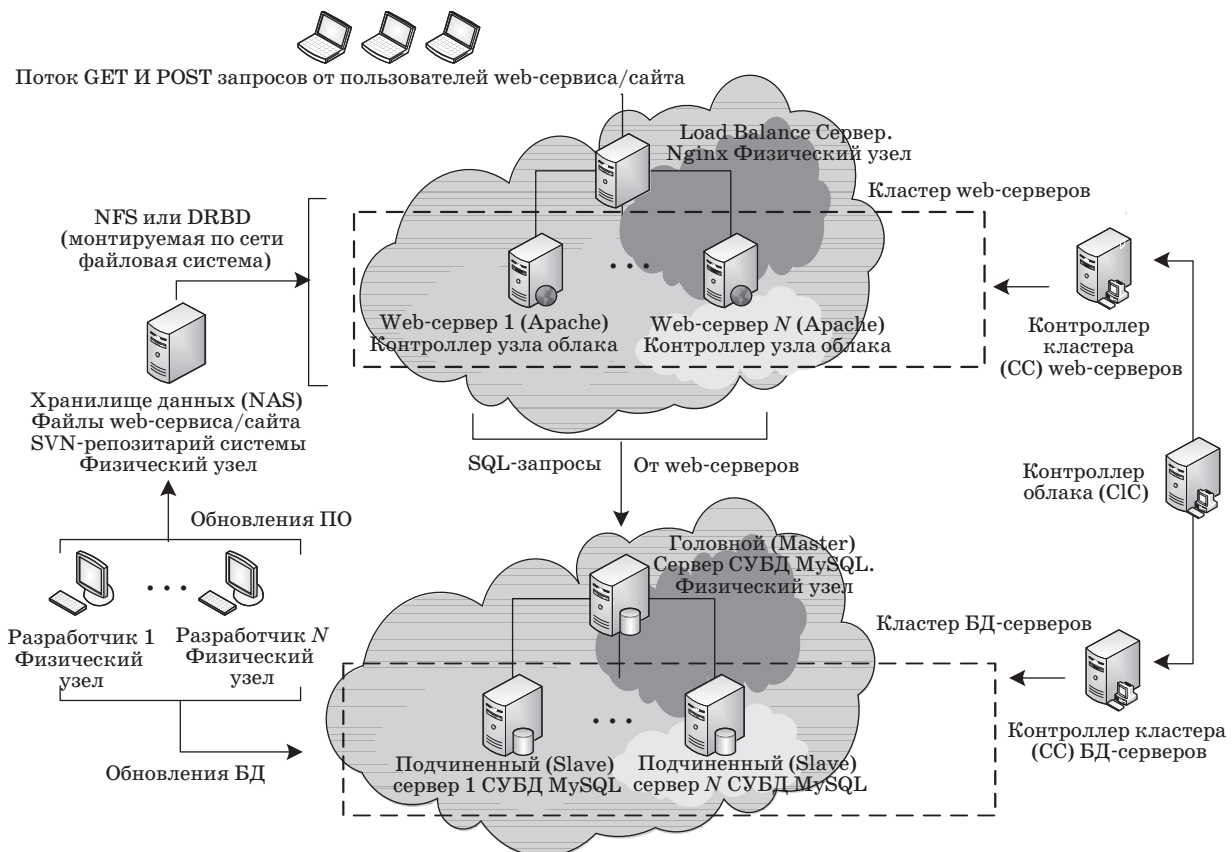
Eucalyptus позиционируется как открытое (open source) решение для организации доступа к вычислительным ресурсам с возможностью динамического масштабирования системы и балансировки нагрузки [2]. Важно заметить, что Eucalyptus на данный момент является единственным доступным бесплатным открытым решением построения вычислительного кластера (рис. 2) при создании грид-системы для web-сайта.



■ Рис. 2. Применение вычислительного кластера

### Развертывание грид-системы

Также одним из методов эффективного повышения вычислительной мощности системы и увеличения скорости обработки запросов является развертывание грид-системы (рис. 3), в которой вычислительные элементы помещаются в «облако», где действует система предоставления ресур-



■ Рис. 3. Архитектура отказоустойчивой распределенной грид-системы

сов по требованию. Облачные (рассеянные) вычисления<sup>4</sup> (cloud computing) — технология обработки данных, в которой компьютерные ресурсы и мощности предоставляются пользователю (или серверу) как интернет-сервис [3]. Пользователь имеет доступ к собственным данным, но не может управлять и не должен заботиться об инфраструктуре, операционной системе и собственно программном обеспечении, с которым он работает. Термин «облако» используется как метафора, основанная на изображении Интернета на диаграмме компьютерной сети, и как абстракция сложной инфраструктуры, за которой скрываются все технические детали. На данный момент существует разнообразное ПО для построения «облачной» системы. Одно из них — система Eucalyptus (см. рис. 3).

В «облаке» присутствуют разные типы узлов. В самом низу иерархии находится контроллер узла. Он запускается на каждом физическом сервере, подключенном к «облаку», и отвечает за запуск, работу и остановку виртуальных машин. Контроллер кластера управляет контроллерами узлов, определяет, на каких узлах будут запущены виртуальные машины. Контроллер «облака» устанавливается на машине, имеющей доступ к внешней сети (можно установить его на фронт-сервере системы) и выступает в роли головного интерфейса для доступа к «облаку». Обрабатывает пользовательские запросы на запуск виртуальных машин и собирает данные о загруженности узлов от контроллеров кластеров.

Для начала необходимо определить, какие именно вычисляющие элементы будут помещены в «облако». Во время работы web-сайта самыми ресурсоемкими задачами являются обработка SQL-запроса СУБД [4], а также интерпретация программы на языке web-программирования. Значит, в «облако» необходимо поместить сервер(а) СУБД MySQL и web-сервер(а) Apache. Задачи по выдаче статической информации (картинки, css-стили, скрипты на языке Javascript) можно переложить на быстрый web-сервер, например Nginx или LightHttpd. Это и есть решение задачи классификации трафика на статический и динамический. Nginx также может балансировать нагрузку между вычисляющими web-серверами.

## Типы узлов и используемое ПО

Рассмотрим основные типы узлов предлагаемой системы.

1. Сервер распределения нагрузки может быть построен на базе «легких» http-серверов, таких как Nginx, LightHttpd и др. Он выполняет функ-

<sup>4</sup> Используется также термин «облачная» (рассеянная) обработка данных.

ции по распределению запросов на статические и динамические. Статические запросы сервер выполняет сам (отдавая статический контент), а динамические передает вычисляющим узлам. Сервер распределения нагрузки может быть размещен как на физической машине, так и в «облаке», но обязательно со статическим IP-адресом и портом, куда будут поступать http-запросы от браузеров пользователей.

2. Web-сервер (вычисляющий узел) — виртуальная машина, расположенная в «облаке», на которой установлен программный web-сервер Apache. Он интерпретирует программный код сайта, написанный на языках web-разработки (таких как PHP, Perl). При выходе из строя одного из вычисляющих узлов сервер распределения нагрузки будет посылать запросы на другие. Таким образом достигается отказоустойчивость системы. При необходимости дополнительных ресурсов есть возможность добавления нового узла. Если же добавочная нагрузка небольшая (относительно нагрузки всего кластера web-серверов), то «облако» (а именно контроллер кластера) само выделит необходимые ресурсы. Подключение нового узла имеет смысл только при явной перегруженности всего «облака». Предлагаемая архитектура такова, что ресурсы для вычисляющих узлов выделяются автоматически, по требованию. За этим следит контроллер кластера.

3. Головной (master) сервер СУБД получает SQL-запрос и транспортирует его одному из slave-серверов.

4. Подчиненный (slave) сервер СУБД выполняет SQL-запрос и возвращает результат Apache-серверу.

5. Хранилище данных NAS содержит репозиторий с программным кодом сайта. Его каталог монтируется к вычисляющим web-серверам. Также он задействуется разработчиками для обновления ПО.

6. Контроллер кластера управляет контроллерами узлов, определяет, на каких узлах будут запущены виртуальные машины. Он построен на базе ПО Eucalyptus, так же как и контроллер «облака».

7. Контроллер «облака» — именно он осуществляет централизованное управление «облаком» через контроллеры кластеров.

## Принцип работы грид-системы

Рассмотрим цикл работы web-сайта. Сначала разработчики создают программную часть web-сайта, они коллективно работают с системой SVN. Subversion (SVN) [1] — свободная централизованная система управления версиями, созданная в 2000 г. компанией CollabNet Inc. Ката-



лог с программой монтируется каждым из web-серверов в качестве папки для данного сайта. После того как http-запрос пользователя (это может быть запрос главной страницы сайта) попадает на 80-й порт сервера распределения нагрузки, он определяет, какому из вычисляющих узлов (в данном случае это web-сервер Apache) направить его для обработки. В качестве базового используется алгоритм выбора наименее загруженного в данный момент сервера. Далее запрос идет на вычисляющий узел. Он соединяется с мастер-сервером СУБД и отправляет ему SQL-запрос, после чего мастер-сервер аналогичным образом определяет, какому из своих подчиненных вычислительных узлов направить запрос для вычисления. После обработки SQL-запроса результат обработки возвращается мастер-серверу СУБД, а затем вычислительному узлу (web-серверу), который его инициировал. Вычисляющий web-сервер интерпретирует программный код дальше. В результате интерпретации пользователь получает html-страницу. Далее браузер пользователя делает запросы к фронт-серверу (серверу распределения нагрузки). Он на основе типа запроса (запрос на статический или динамический контент) определяет, передавать ли запрос вычисляющему узлу или выполнять его самостоятельно. Во втором случае фронт-сервер самостоятельно находит запрашиваемый ресурс (картинка, css-файл, js-файл) и выдает его браузеру пользователя. За счет меньшей ресурсоем-

кости сервера Nginx нагрузка на вычисляющие узлы значительно снижается.

### Заключение

Предложенная архитектура дает возможность построить быструю, отказоустойчивую и масштабируемую распределенную систему обработки большого количества запросов для web-сайта. Данные параметры распределенной системы достигаются благодаря применению метода классификации трафика и помещению вычисляющих узлов в «облако», а стоимость решения снижается благодаря использованию открытого ПО.

### Литература

1. Фаулер М., Райса Д. Архитектура корпоративных программных приложений: Пер. с англ. — М.: Вильямс, 2008. — 544 с.
2. Хоп Г., Вульф Б. Шаблоны интеграции корпоративных приложений: Пер. с англ. — М.: Вильямс, 2007. — 672 с.
3. Веллинг Л., Томсон Л. Разработка Web-приложений с помощью PHP и MySQL: Пер. с англ. — М.: Вильямс, 2005. — 880 с.
4. Дюбуа П. MySQL. Сборник рецептов: Пер. с англ. — М.: Вильямс, 2004. — 1056 с.

## 63-Я МЕЖДУНАРОДНАЯ СТУДЕНЧЕСКАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ — «63 МСНК» 19–23 апреля 2010 г.

С 19 по 23 апреля 2010 года Санкт-Петербургский государственный университет аэрокосмического приборостроения провел 63-ю Международную студенческую научную конференцию, посвященную Всемирному дню космонавтики и 90-летию профессора Виталия Васильевича Хрущева.

Ежегодное проведение студенческих научных конференций и выставок научно-технического творчества студентов в ГУАП является одним из важнейших итоговых научных студенческих мероприятий. Научные работы студентов посвящены актуальным проблемам авиационного и аэрокосмического приборостроения, развития информационных технологий, радиотехники, электроники и связи, современным вопросам экономики, управления, философии и права.

Традиционно мероприятия конференции были включены в программу Недели университета. Работа конференции проходила по 12 научным на-

правлениям и была организована в 54 научных секциях, в которых приняли участие свыше 600 студентов. В программу конференции вошли 557 докладов студентов. В Круглом зале библиотеки на Большой Морской улице прошла выставка дипломных проектов ГУАП-2010.

В демонстрационном зале СЗЦИТ была проведена выставка научно-технического творчества молодежи и III Международная студенческая научная интернет-конференция Международного общества автоматизации (ISA), в которой приняли участие студенты и профессора из университетов Италии, Испании, США, России и Чили. На ней с докладами выступили победители V Европейского конкурса ISA на лучшую научную работу студентов (ESPC-2010). Рабочим языком интернет-конференции был английский язык.

По решению научных секций 63 МСНК лучшие доклады будут опубликованы в сборнике студенческих научных работ.