

Санкт-Петербургский государственный университет

Кафедра системного программирования

Балашов Вадим Антонович

Сетевой вычислитель. Организация
хранения и передачи данных

Курсовая работа

Научный руководитель:
ст. преп. Баклановский М. В.

Санкт-Петербург
2019

Оглавление

Введение	3
1. Постановка цели и задач	4
2. Обзор существующих решений	5
2.1. ATA over Ethernet	5
2.2. Direct Connect	5
2.3. BitTorrent	6
2.3.1. DHT	7
2.3.2. PEX	7
3. Описание реализации	9
3.1. Трекер	9
3.2. Клиент	10
4. Дальнейшее развитие	11
Заключение	12
Список литературы	13

Введение

Мы живём в век информационных технологий, когда количество данных, которые мы обрабатываем, передаём и храним, достигает колоссальных размеров. Поэтому сейчас особенно остро стоит вопрос о грамотном, эффективном, а главное производительном управлении ИТ-процессами в компаниях, где важным условием успешности предприятия является быстрота и надёжность передаваемой информации. Подобное управление осуществляется с помощью систем, позволяющих автоматизировать все информационные процессы внутри предприятия и избежать сбоев и неполадок на производстве. Такими системами являются ERP-системы. ERP (англ. Enterprise Resource Planning, планирование ресурсов предприятия) — это метод эффективного использования людей, аппаратного и программного обеспечения для повышения производительности и прибыли и упрощения бизнес-процессов компании. ERP-система — конкретный программный пакет, реализующий стратегию ERP [2][3]. Но повсеместному использованию ERP-систем мешает высокая стоимость качественных продуктов, представленных на рынке, и низкая производительность дешёвых аналогов. И у первых, и у вторых сети построены по топологии «звезда», где в центре находится один или несколько серверов, которые берут на себя бóльшую часть вычислений. Эти проблемы привели нас к разработке собственной конкурентоспособной и более дешёвой ERP-системы с принципиально другой архитектурой на основе сетевого вычислителя. Сетевой вычислитель — это система, позволяющая использовать все доступные ресурсы внутри сети и распределить нагрузку по ним. Иными словами, мы снимаем нагрузку с сервера и нагружаем другие доступные машины в сети, оставляя на попечение серверу работу, быстродействие которой упирается не в производительность линии связи, а в, например, скорость обращения к данным в хранилище. Важной частью создания такой системы является разработка методов хранения и протокола передачи данных внутри сети, о чём и пойдёт речь в дальнейшем.

1. Постановка цели и задач

Цель работы – разработка методов хранения и передачи данных внутри сетевого вычислителя. Для её достижения были поставлены следующие задачи:

- Изучить предметную область
- Сравнить существующие подходы
- Разработать методы хранения и обмена данными
- Реализовать систему хранения и обмена данными
- Исследовать дальнейшие перспективы развития

2. Обзор существующих решений

Рассмотрим достоинства и недостатки существующих протоколов обмена данными:

2.1. ATA over Ethernet

ATA over Ethernet – это сетевой протокол, предназначенный для высокоскоростного доступа к устройствам хранения данных SATA через сеть Ethernet [4]. Протокол позволяет отказаться от TCP/IP и предоставить работу коммутаторам, у которых отсутствуют коллизии и есть возможность управления потоком.

Достоинства:

- Высокая производительность
- Низкая стоимость
- Простота в использовании

Недостатки:

- Слабая масштабируемость (строится внутри одной Ethernet сети)
- Централизованный доступ
- Отсутствие механизмов контроля ошибок передачи данных, помимо тех, что есть у самой сети Ethernet

2.2. Direct Connect

Direct Connect (DC) — это одноранговый протокол обмена файлами. DC-клиенты подключаются к центральному хабу и могут загружать файлы напрямую друг от друга. Advanced Direct Connect можно считать преемником данного протокола [6]. Процесс обмена заключается в следующем:

1. Пользователь вместе с другими участниками подключается к хабу.
2. Отмечает файлы и папки, которыми хочет поделиться, делая их доступными для остальных подключённых к хабу.
3. Производит поиск файлов и папок по shared-данным внутри хаба и скачивает необходимое.

Достоинства:

- Распределённость

Недостатки:

- Слабая масштабируемость (большой объём лишнего трафика, излишняя централизованность)
- Один файл – один источник

2.3. BitTorrent

BitTorrent (букв. битовый поток) – это пиринговый протокол обмена данными по сети [7]. BitTorrent-трекер – это особый тип сервера, который помогает в обмене данными между равноправными узлами с использованием протокола BitTorrent [5]. Процесс обмена заключается в следующем:

1. Пользователю необходимо иметь .torrent файл, содержащий метаданные о раздаче.
2. Сделать запрос(ы) на трекер(ы), указанный(ые) в файле, и получить список пируров, участвующих в данной раздаче.
3. Установить соединение с участниками раздачи и произвести непосредственный обмен данными.

Достоинства:

- Практически неограниченная масштабируемость
- Высокая производительность
- Распределённость (снижение нагрузки на сервер)
- Возможность создания приватной инфраструктуры

Недостатки: Единственным недостатком для нашей системы в BitTorrent является недоступность раздачи. В случае, когда нет ни одного сида (клиента, у которого есть все сегменты), а данных у пиров недостаточно, необходимо ждать появления либо сида, либо пира, у которого имеются недостающие сегменты.

2.3.1. DHT

BitTorrent/DHT (англ. distributed hash table – распределённая хэш-таблица) – это протокол, который позволяет ВТ-клиентам находить пиров с необходимыми файлами без использования трекера [1]. DHT позволяет исключить из процесса обмена сервер и необходимость в .torrent файлах и искать нужные данные напрямую у участников DHT-сети. В случае использования DHT мы ищем не конкретную раздачу, а отдельные файлы из неё. Вместе с этим у DHT-сетей есть недостаток – ”порог вхождения”, т. е. для того, чтобы стать участником сети, необходимо ”подружиться” хотя бы с одним из её текущих участников. Возможной проблемой использования DHT будет доступность внутренних файлов внешнему миру. Доступ к закрытому трекеру, а, соответственно, и к данным, есть только у авторизованных пользователей. Но DHT позволяет по хешу через ”знакомых знакомых” получить запрашиваемый сегмент.

2.3.2. PEX

PEX (англ. Peer exchange) – вспомогательное расширение протокола BitTorrent для обмена списками пиров. PEX реализуется в виде дополнительных сообщений между участниками, между которыми уже есть

соединение по обмену данными. РЕХ является лишь вспомогательной технологией по той причине, что новые пиры не смогут подключиться к раздаче. Для этого им необходимы либо трекер, либо DHT.

3. Описание реализации

В виду направленности нашей работы на производственное применение, а именно использование на предприятиях и в компаниях с большим количеством участвующих в бизнес-процессах людей и соответствующим количеством находящихся в сети компьютеров, наше решение будет иметь torrent-based архитектуру. Обоснованием такого выбора является прямая зависимость производительности сети от количества участников сети при использовании децентрализованного протокола BitTorrent и обратная зависимость при использовании централизованных подходов. Чем больше участников в сети, тем выше утилизация сетевого канала у принимающего и ниже нагрузка на каждого раздающего при использовании BitTorrent, и тем выше нагрузка на центральный сервер при использовании AoE или iSCSI.

Исходя из степени изученности и частоты применения, самым удачным языком программирования для написания реализации является PHP.

3.1. Трекер

На данном этапе решено воздержаться от полного отказа от сервера и продолжить использовать его в качестве трекера и резервного хранилища данных. Также на нём будет запущен клиент для сидирования. Трекер реализован в виде нескольких простых скриптов, которые в дальнейшем интерпретируются любым удобным веб-сервером с поддержкой PHP. В виду крайне малого объёма необходимого для написания исходного кода, не были использованы никакие фреймворки для разработки веб-приложений, за исключением лишь одной библиотеки для кодирования и декодирования данных в формат bencode. Написаны два скрипта для взаимодействия по HTTP и обработки announce и scrape запросов и подключаемый к ним скрипт для взаимодействия с базой данных. (Рис. 1)

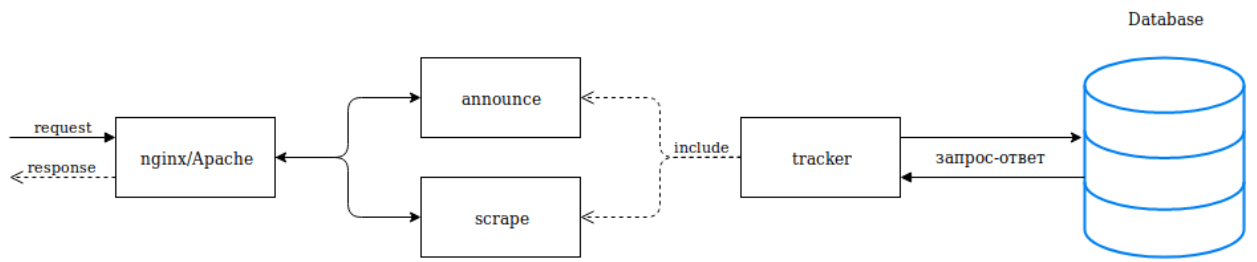


Рис. 1: Трекер

3.2. Клиент

Клиент реализован на основе ReactPHP. ReactPHP – библиотека для событийного программирования на PHP с неблокирующим вводом-выводом. Основа ReactPHP – Event Loop – бесконечный цикл, который реагирует на происходящие события – поступающие сигналы или поток ввода. Реакция заключается в вызове обработчиков происходящих событий. (Рис. 2)

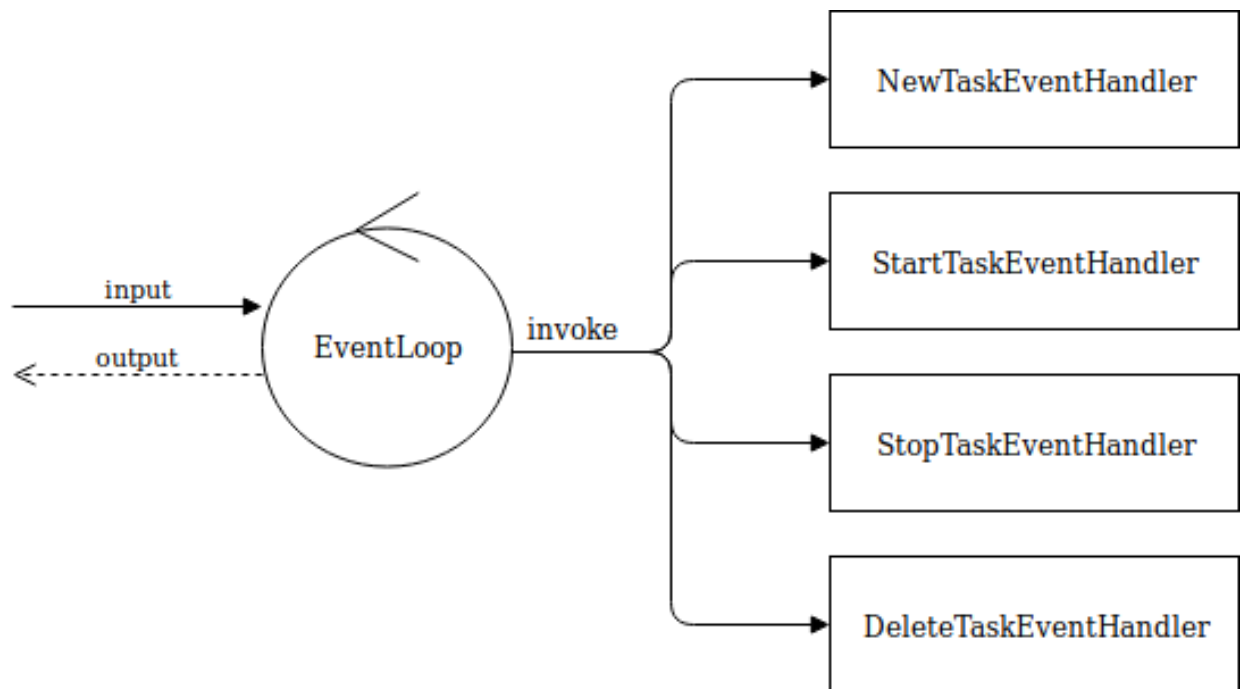


Рис. 2: Клиент

4. Дальнейшее развитие

- Разработка методов распределения данных между узлами в сети
- Отказ от трекера в пользу DHT с инициализацией в виде "знакомства" всех со всеми
- Полный отказ от сервера

Заключение

В ходе работы были достигнуты следующие результаты:

- Изучена предметная область
- Рассмотрены существующие решения
- Разработаны методы хранения и обмена данными
- Реализована система хранения и обмена данными
- Исследованы дальнейшие перспективы развития

Список литературы

- [1] Andrew Loewenstern <drue@bittorrent.com> Arvid Norberg <arvid@bittorrent.com>. DHT Protocol // BitTorrent.org. — 2008. — URL: http://www.bittorrent.org/beps/bep_0005.html (дата обращения: 14.12.2018).
- [2] Garther. ERP // Gartner, Inc. — 2004. — URL: <https://www.gartner.com/it-glossary/enterprise-resource-planning-erp/> (дата обращения: 14.12.2018).
- [3] Leon Alexis. Enterprise Resource Planning. — Tata McGraw-Hill Education, 2008. — ISBN: 0070656800, 9780070656802. — Google Books : http://books.google.ru/books?id=pTGDy2GX_sUC.
- [4] Wikipedia. ATA over Ethernet // Wikipedia, the free encyclopedia. — 2018. — URL: http://en.wikipedia.org/wiki/ATA_over_Ethernet (online; accessed: 26.12.2018).
- [5] Wikipedia. BitTorrent tracker // Wikipedia, the free encyclopedia. — 2018. — URL: http://en.wikipedia.org/wiki/BitTorrent_tracker (online; accessed: 26.12.2018).
- [6] Wikipedia. Direct Connect (protocol) // Wikipedia, the free encyclopedia. — 2018. — URL: [http://en.wikipedia.org/wiki/Direct_Connect_\(protocol\)](http://en.wikipedia.org/wiki/Direct_Connect_(protocol)) (online; accessed: 26.12.2018).
- [7] <bram@bittorrent.com> Bram Cohen. The BitTorrent Protocol Specification // BitTorrent.org. — 2008. — URL: http://www.bittorrent.org/beps/bep_0003.html (дата обращения: 14.12.2018).