

Эта часть работы выложена в ознакомительных целях. Если вы хотите получить работу полностью, то приобретите ее воспользовавшись формой заказа на странице с готовой работой:

<https://stuservis.ru/referat/175832>

Тип работы: Реферат

Предмет: Высшая математика (другое)

Введение 3

1 Типы распределенных систем 4

2 Распределенная система автоматизации 8

Заключение 11

Список литературы 12

Введение

В настоящее время подавляющее большинство компьютерных программ является распределенными. Распределенными системами называются системы, в которых информация сосредоточена на нескольких физических носителях. Распределенные системы различаются по сложности реализации. И, хотя реализация систем является нетривиальной, распределенные системы обладают всеми преимуществами распределенных вычислений, и, что очень важно, лишены присущих им недостатков. В частности, сложных режимов отказа и недетерминированности. Для проектирования распределенных систем необходимы строгие методы описания. Распределенная система существенно смягчает требования к ОС. Поэтому, дальнейшая разработка новых распределенных систем является актуальной научно-технической задачей. Объектом данного исследования являются распределенные системы.

Предметом данного исследования являются распределенные системы в мире.

Целью данного исследования является описание распределенных систем в мире.

Для того, чтобы достичь поставленной цели, необходимо решить следующие задачи:

- описать типы распределенных систем;

- проанализировать самые распространенные мировые системы.

Работа состоит из введения, двух глав, заключения и списка литературы.

В первой главе рассматриваются основные типы распределенных систем и их общие характеристики.

Представлены основные модели распределенной системы, модели физического устройства и модели ресурса.

Во второй главе рассматриваются наиболее популярные мировые системы. Подробно рассматривается система промышленной автоматизации на основе общей шины и принципы функционирования web-сервисов.

1 Типы распределенных систем

Все распределительные системы можно подразделить на три типа.

К первому типу относятся автономные распределительные системы. К этому типу относятся системы пакетной обработки, кластеры анализа больших данных, фермы рендеринга сцен кино, кластеры сворачивания белка и т. п. Подобные системы обладают всеми преимуществами распределенных вычислений: масштабированием, отказоустойчивостью, и т.д. А также лишены присущих остальным системам недостатков, таких как сложных режимов отказа, недетерминированности, и т.д.

Далее находятся распределенные системы, функционирующие в режиме мягкого реального времени.

Основная функция подобных систем заключается в непрерывном создании и обновлении результатов. К таким системам относятся некоторые построители поисковых индексов, системы поиска неисправных серверов, роли для Amazon Elastic Compute Cloud (Amazon EC2) и т. д. Индексатор поиска способен работать в автономном режиме от 10 минут до нескольких часов, не вызывая ухудшения работы пользователя.

К третьему типу относятся системы жесткого реального времени. Их также называют сервисами «запрос-отклик». Для систем подобного типа сложно организовать правильную реализацию. Это обусловлено тем, что запросы поступают неожиданно, и отклики необходимо отправить быстро. К ним относятся интерфейсные веб-серверы, конвейер заказов, транзакции по кредитным картам, и т.д.

Далее будем рассматривать системы жесткого отклика. Сложность распределенных систем жесткого отклика состоит в том, что система система может опрашивать сообщение из одной области сбоя в другую.

Эта операция, на первый взгляд, представляется легкой задачей, но подобные отправки много сложнее.

В качестве примера рассмотрим реализацию фрагмента кода из игры игры Pac-Man [1]:

```
board.move(pacman, user.joystickDirection())
ghosts = board.findAll(":ghost")
for (ghost in ghosts)
if board.overlaps(pacman, ghost)
user.slayBy(":ghost")
board.remove(pacman)
return
```

Представим сетевую версию этого кода, в котором объект board хранится в отдельном сервере. В этом случае каждый вызов объекта board приводит к отправке и получению сообщений между двумя серверами. Так, когда с одного сервера на другой отправляется сообщение request / reply, то должно, как минимум, выполняться восемь действий. Рассмотрим подробнее передачу сообщений «запрос-отклик». Цикл «запрос-отклик» всегда состоит из одних и тех же этапов (Рис. 1).

Рис. 1. Схема цикла сообщений «запрос-отклик» [1].

Как показано на Рис. 1, клиентская машина CLIENT отправляет запрос MESSAGE по сети NETWORK на клиентскую машину SERVER, которая отправляет ответ. Таким образом, в благоприятном случае выполняются следующие действия:

1. Отправка запроса: CLIENT отправляет запрос MESSAGE по сети NETWORK;
2. доставка запроса: сеть NETWORK доставляет сообщение на клиентскую машину SERVER;
3. проверка запроса: SERVER проверяет MESSAGE;
4. обновление SERVER: SERVER при необходимости обновляет свое состояние;
5. отправка ответа: SERVER отправляет сообщение REPLY в сеть NETWORK;
6. доставка ответа: сеть NETWORK доставляет сообщение REPLY в машину CLIENT;
7. проверка ответа: CLIENT проверяет REPLY;
8. обновление CLIENT: CLIENT при необходимости обновляет свое состояние.

С физической точки зрения первая операция означает отправку пакетов через сетевой адаптер, в результате которой электрические сигналы передаются по маршрутизаторам, образующих сеть между компьютерами CLIENT и SERVER. Данная операция реализуется отдельно от операции 2, поскольку она может закончиться неудачно по некоторым причинам. Например, неожиданная потеря питания SERVER и невозможность принять входящие пакеты. Аналогично происходят и основные этапы. Следовательно, цикл «запрос-отклик

. Габриэльсон Дж. Сложности, связанные с распределенными системами [Электронный ресурс]. – Режим доступа: <https://aws.amazon.com/ru/builders-library/challenges-with-distributed-systems> (дата обращения 29.04.2021).

2. Распределенные системы автоматизации [Электронный ресурс]. – Режим доступа: https://www.bookasutp.ru/chapter1_1_3.aspx (дата обращения 29.04.2021).

3. Соммервилл И. Инженерия программного обеспечения / И. Соммервилл. – М.: Вильямс. – 2002. – 624 с.

4. Драница А. Java против .NET [Электронный ресурс]. – Режим доступа: <https://old.computerra.ru/2003/516/203739/> (дата обращения 29.04.2021).

5. Бернс Брендан. Распределенные системы. Паттерны проектирования / Брендан Бернс. – СПб: Питер. – 2019. – 224 с.

Эта часть работы выложена в ознакомительных целях. Если вы хотите получить работу полностью, то приобретите ее воспользовавшись формой заказа на странице с готовой работой:

<https://stuservis.ru/referat/175832>