Лекция 13: Виртуализация, CAP-теорема

Виртуализация

Причины виртуализации

- средний уровень загрузки Windows серверов ~ 5%
- Unix-серверы ~ 10-20%
- Подход "одно приложение — один сервер":
  - быстрое увеличение серверного парка
  - возрастание затрат на администрирование, энергопотребление и охлаждение
  - потребность в дополнительных помещениях

Виртуализация

- В основе виртуализации лежит возможность одного компьютера выполнять работу нескольких компьютеров благодаря распределению его ресурсов по нескольким средам.
- Применение виртуализации позволяет распределять вычислительные ресурсы между приложениями, каждое из которых при этом "видит" только предназначенные ему ресурсы и "считает", что ему выделен отдельный сервер.
- Кроме того, решения виртуализации дают возможность запускать в разделах разные ОС с помощью эмуляции их системных вызовов к аппаратным ресурсам сервера

История Виртуализации

- Компания IBM была первой, кто задумался о создании виртуальных сред для различных пользовательских задач, тогда еще в мэйнфреймах (1967 г. – платформа IBM CP-40)
- После появления персональных компьютеров интерес к виртуализации несколько ослаб ввиду бурного развития операционных систем, которые предъявляли адекватные требования к аппаратному обеспечению того времени.
- Однако бурный рост аппаратных мощностей компьютеров в конце девяностых годов прошлого века заставил ИТ-сообщество вновь вспомнить о технологиях виртуализации программных платформ.

История Виртуализации

- В 1999 г. компания VMware представила технологию виртуализации систем на базе x86.
- Позднее в "битву" за место в этом модном направлении развития информационных технологий включились такие компании как
  - Parallels (ранее SWsoft),
  - Oracle (Sun Microsystems),
  - Citrix Systems (XenSourse),
  - Microsoft (в 2003 г., купив Connectix и выпустив свой первый продукт Virtual PC).

Преимущества виртуализации

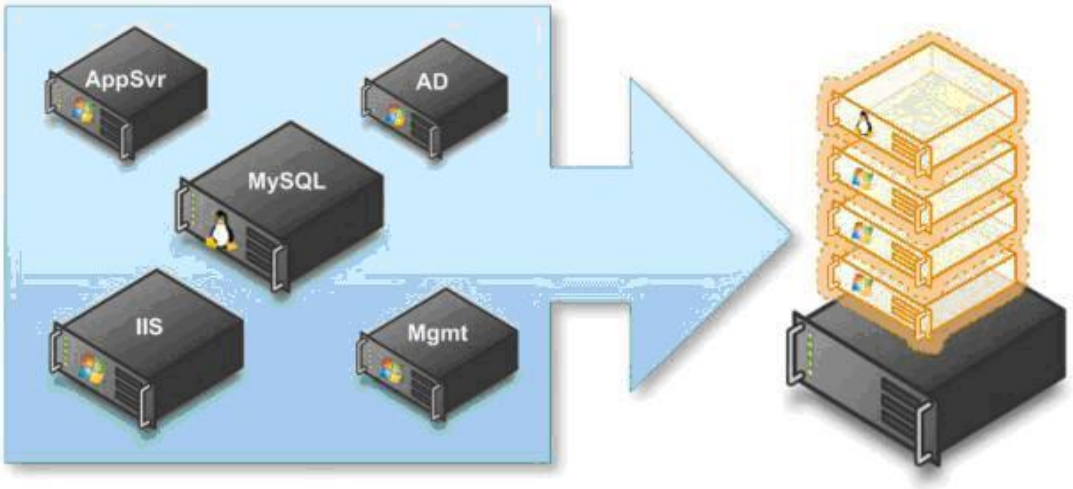
- Эффективное использование вычислительных ресурсов.
- Сокращение расходов на инфраструктуру.
- Снижение затрат на программное обеспечение.
- Повышение гибкости и скорости реагирования системы.
- Несовместимые приложения могут работать на одном компьютере.
- Повышение доступности приложений и обеспечение непрерывности работы предприятия.
- Возможности легкой архивации.
- Повышение управляемости инфраструктуры.

Виртуальная машина

- **Виртуальной машиной** будем называть программную или аппаратную среду, которая скрывает настоящую реализацию какого-либо процесса или объекта от его видимого представления.
- **Виртуальная машина** — это полностью изолированный программный контейнер, который работает с собственной ОС и приложениями, подобно физическому компьютеру. Виртуальная машина действует так же, как физический компьютер, и содержит собственные виртуальные (т.е. программные) ОЗУ, жесткий диск и сетевой адаптер.

Виртуализация серверов

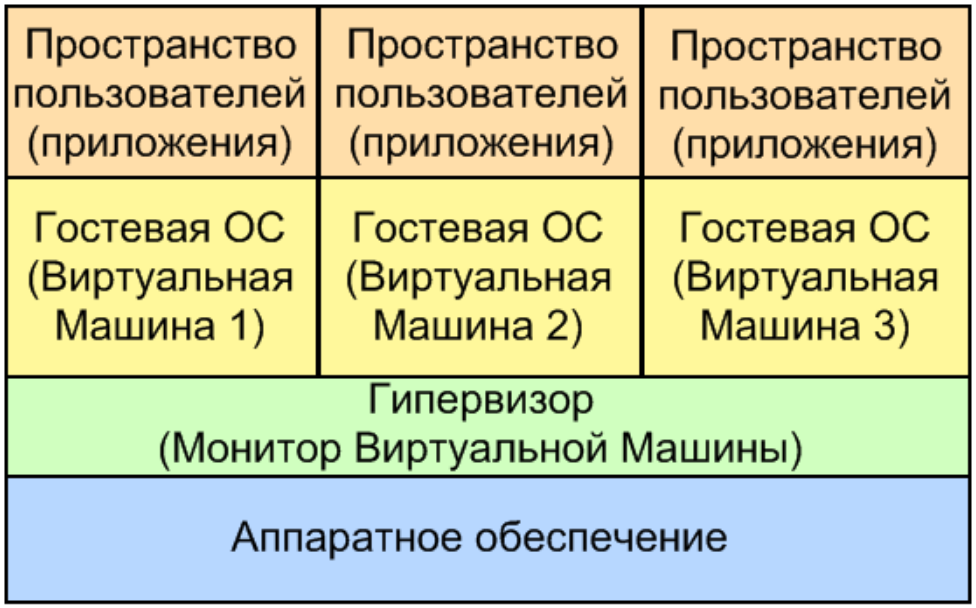
- Виртуализация серверов подразумевает запуск на одном физическом сервере нескольких виртуальных серверов.
- На каждой виртуальной машине может быть установлена операционная система, на которую могут быть установлены приложения и службы.
- Типичные представители это продукты VmWare (ESX, Server, Workstation) и Microsoft (Hyper-V, Virtual Server, Virtual PC).



Гипервизор

- **Монитор Виртуальных Машин (Гипервизор, Virtual Machine Monitor, VMM)** берет на себя управление гостевыми системами. Его можно рассматривать как абстракцию между аппаратной платформой и виртуальными машинами.

- В некоторых случаях гипервизор является операционной системой; в этом случае он называется базовой операционной системой.



Виды виртуализации

Эмуляция оборудования

- Гостевая ОС запускается на виртуальной машине, которая перехватывает каждую команду и моделирует ее на реальном аппаратном обеспечении.
- Нет привязки к архитектуре host-машины (можно запускать неизмененные операционные системы, предназначенные для машин с другой архитектурой).
- Очень низкая скорость работы.

Полная виртуализация

- Используются не модифицированные экземпляры гостевых операционных систем, а для поддержки работы этих ОС служит общий слой эмуляции их исполнения поверх хостовой ОС
- Гипервизор перехватывает только некоторые опасные команды, все остальные напрямую выполняются на аппаратном обеспечении.
- VMware Workstation, VMware Server (бывший GSX Server), Parallels Desktop, Parallels Server, MS Virtual PC, MS Virtual Server, Virtual Iron.

Паравиртуализация

- Модификация ядра гостевой ОС выполняется таким образом, что в нее включается новый набор API, через который она может напрямую работать с аппаратурой, не конфликтуя с другими виртуальными машинами.

- VMware ESX Server, Xen (и решениях других поставщиков на базе этой технологии), Microsoft Hyper-V.

Виртуализация на уровне ядра ОС

- Использование одного ядра хостовой ОС для создания независимых параллельно работающих операционных сред. Для гостевого ПО создается только собственное сетевое и аппаратное окружение.
- Virtuozzo (для Linux и Windows), OpenVZ (бесплатный вариант Virtuozzo) и Solaris Containers.

Виртуализация приложений

- Сильная изоляция прикладных программ с управляемым взаимодействием с ОС, при которой виртуализируется каждый экземпляр приложений, все его основные компоненты: файлы (включая системные), реестр, шрифты, INI-файлы, COM-объекты, службы.

- Sun Java Virtual Machine, Microsoft Application Virtualization (ранее называлось Softgrid), Thinstall (в начале 2008 г. вошла в состав VMware), Symantec/Altiris

Платформы виртуализации

VMware

- Компания VMware – один из первых игроков на рынке платформ виртуализации.
- В 1998 году VMware запатентовала свои программные техники виртуализации и с тех пор выпустила немало эффективных и профессиональных продуктов для виртуализации различного уровня:
  - VMware Workstation, предназначен для настольных ПК
  - VMware ESX Server, позволяет консолидировать физические серверы предприятия в виртуальной инфраструктуре.

Citrix (Xen)

- Некоммерческий гипервизор - Xen исследовательский проект компьютерной лаборатории Кембриджского университета.
- Технология позволяет гипервизору в хостовой системе управлять гостевой ОС посредством гипервызовов VMI (Virtual Machine Interface), что требует модификации ядра гостевой системы.
- Бесплатная версия Xen включена в дистрибутивы нескольких ОС, таких как Red Hat, Novell SUSE, Debian, Fedora Core, Sun Solaris

Microsoft

- Microsoft в 2003 году она приобрела компанию Connectix (система Virtual PC). Virtual PC по своим возможностям сопоставим с VMware Player
- 2006 год: Microsoft Virtual Server 2005 – позволяет обеспечить запуск нескольких виртуальных машин на сервере.
- 2008 год: Microsoft Hyper-V обеспечивает виртуализацию на аппаратном уровне, с использованием технологий виртуализации, встроенных в процессоры.

CAP-ТЕОРЕМА

Управление Данными в PBC

В любой сетевой системе, *обеспечивающей хранение совместно доступных данных*, разработчики хотят поддерживать следующие свойства:

- **Согласованность данных (Consistency)** – в системе существует единственная версия данных, соответствующая последней по времени операции обновления.
- **Доступность данных (Availability)** – запрос к PBC в любой момент времени должен завершиться корректным откликом, не зависимо от того, к какому серверу производится подключение.
- **Устойчивость к разделению (Partition tolerance)** - расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций.

CAP-теорема

В 2000-м году Эрик Брюер (Eric Brewer – проф. Калифорнийского университета) предложил следующую теорему:

В любой сетевой системе, *обеспечивающей хранение совместно доступных данных*, одновременно могут поддерживаться только **два** из следующих трех свойств:

- **Согласованность данных (Consistency)**
- **Доступность данных (Availability)**
- **Устойчивость к разделению (Partition tolerance)**

В 2002-м году теорема была математически доказана в условиях отсутствия синхронизации (общих часов).


Доступность
Согласо-
ванность
Устой-
чивость

Что же такое Partition?

Разделение PBC это не только длительный разрыв между серверами, при котором один из них не может связаться с другим

» *Латентность сети также является разделением PBC.*

- Предположим, что у нас есть 2 сервера баз данных: один в России, другой в США.
- Они настроены на полное реплицированные
- Данные обновились на сервере в России. Через какое время сервер в США узнает об этом?
- **200 миллисекунд – лучший возможный результат** (худший возможный результат – никогда не узнает)
- В это «окно» они разделены – таким образом разделение системы происходит постоянно, даже в нормальных условиях работы PBC

Выбираем ли мы устойчивость?

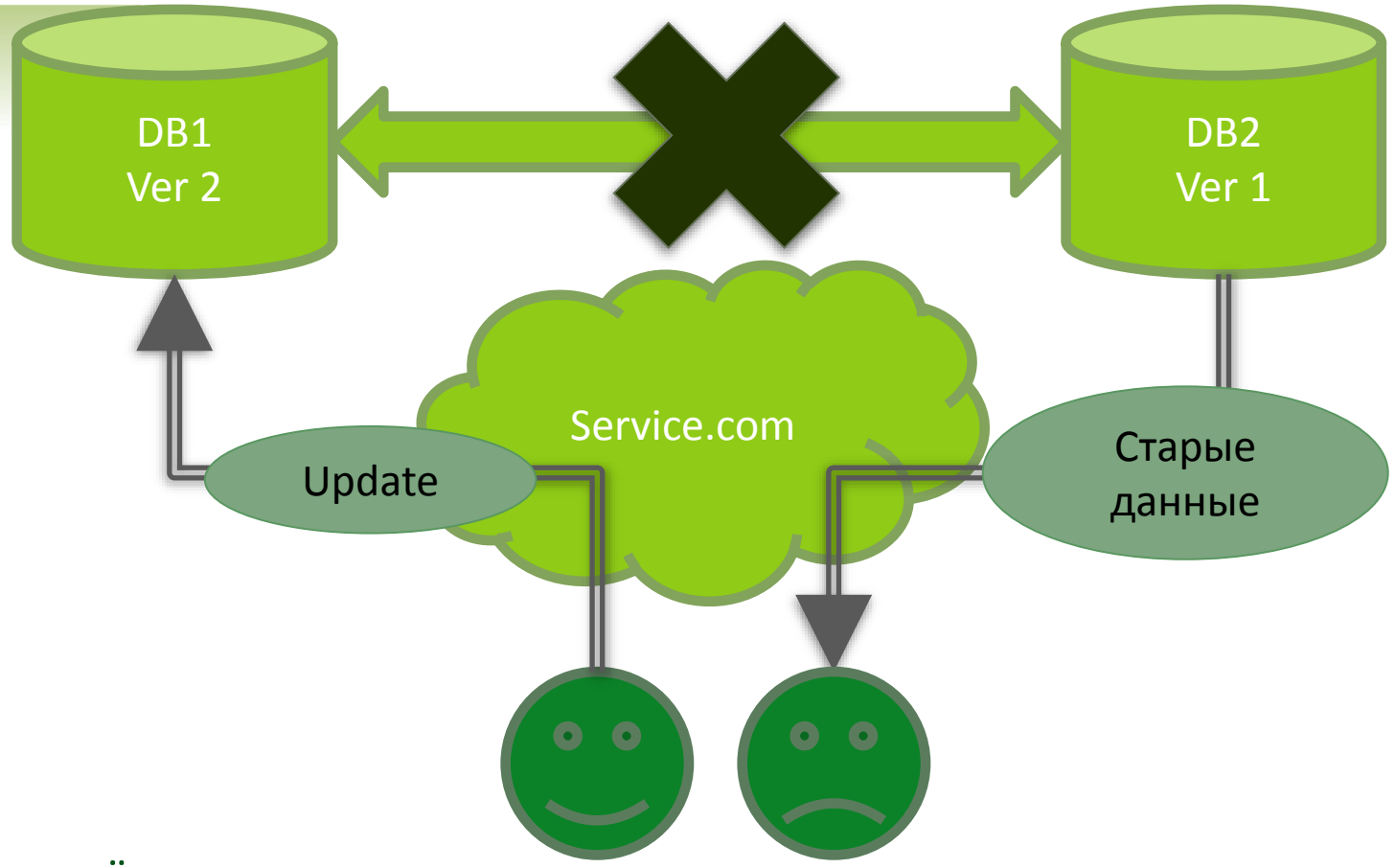
В реальном мире мы не можем выбирать, будут у нас сбои или нет. В РВС всегда будут возникать неполадки, зависит это от нас или нет:

- сетевые сбои,
- сбои работы оборудования,
- ошибки администрирования.

Плюс, любая латентность также вызывает разделение.

Поэтому приходится выбирать из двух вариантов:

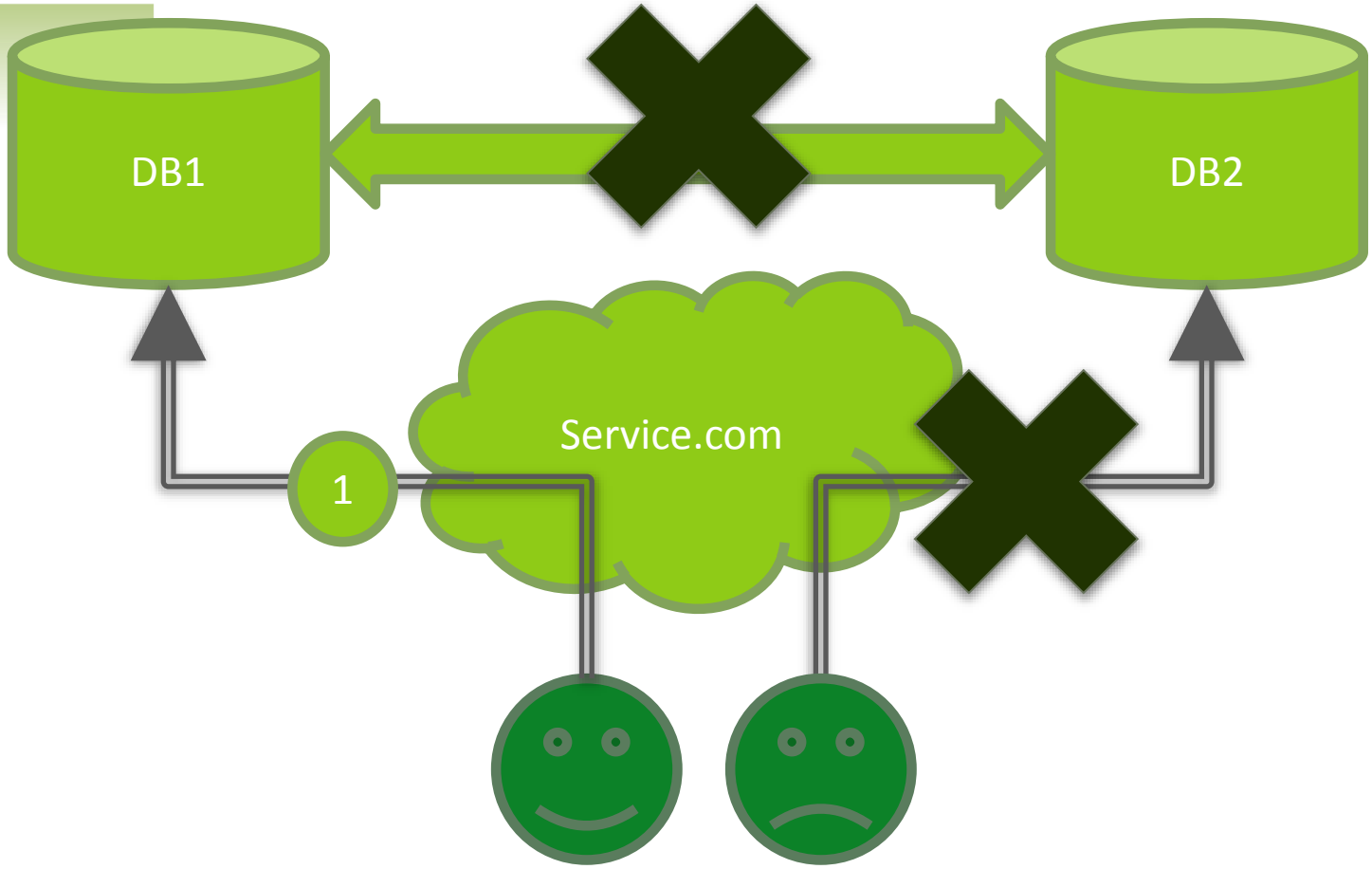
AP: 100% доступность, но несогласованность данных:



Распределённая система, отказывающаяся от целостности результата. Большинство NoSQL-систем принципиально не гарантируют целостности данных («целостные в конечном итоге» - eventually consistent).

CAP

CP: 100% согласованность, но недоступность данных при распаде:

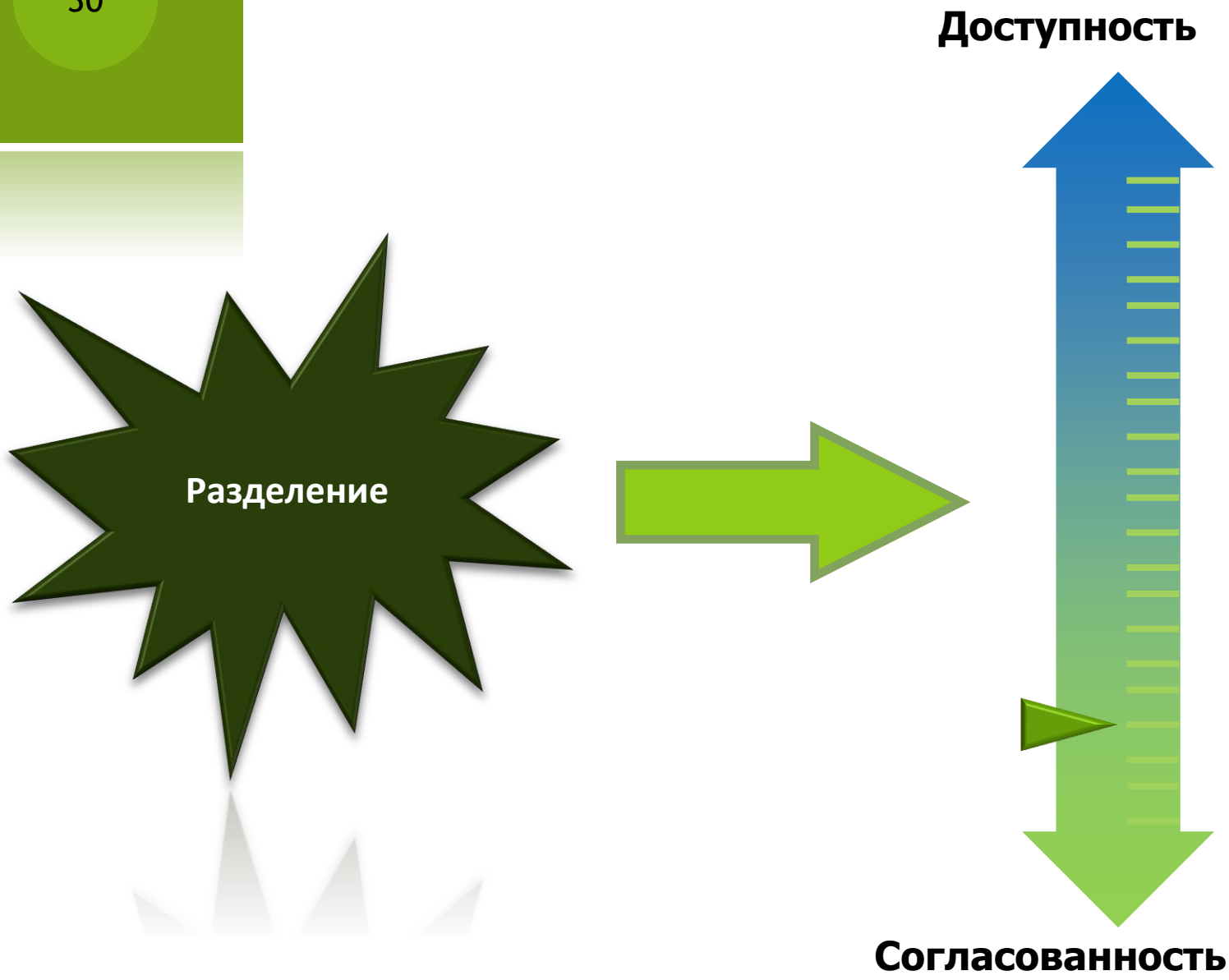


Распределённая система, в каждый момент обеспечивающая целостный результат и способная функционировать в условиях распада, в ущерб доступности может не выдавать отклик. Пессимистические блокировки для сохранения целостности.

Выбор между C и A

- Нет 100% работающего универсального решения, каким путем решать задачу обработки запросов при возникновении расщепления распределенной системы – выбирать доступность или же выбирать
- Этот выбор может быть сделан только разработчиком, на основе анализа **бизнес-модели** приложения.
- Более того, этот выбор может быть сделан не для всего приложения в целом, а отдельно для каждой его части:
  - *Согласованность* для процедуры покупки и оплаты в интернет-магазине;
  - *Доступность* для процедуры просмотра каталога товаров, чтения отзывов и т.п.
- Или же смириться с несогласованностью, в случае коротких периодов (1-2 минуты) разделения системы

Доступность и согласованность



Исключение из CAP- теоремы?

Системы без внутреннего состояния (STATELESS DISTRIBUTED SYSTEMS)

Eventual Consistency

В связи с невозможностью 100% времени поддерживать согласованность и доступность системы, пришлось искать компромисс.

- *Согласованность в конечном счете (eventual consistency) или слабая согласованность (weak consistency)* - означает, что если в течение достаточно долгого периода времени в систему не поступают новые операции обновления данных, то можно ожидать, что результаты всех предыдущих операций обновления данных в конце концов распространятся по всем узлам системы, и все реплики данных **согласуются**.
- При отсутствии сбоев, максимальный размер окна несогласованности может быть определен на основании таких факторов, как задержка связи, загруженность системы и количество реплик в соответствии со схемой репликации.
- Самая популярная система, реализующая «согласованность в конечном счете» – DNS. Обновленная запись распространяется в соответствии с параметрами конфигурации и настройками интервалов кэшированя. В конечном счете, все клиенты увидят обновление.

Более подробно о согласованности в конечном итоге: http://habrahabr.ru/post/100891/

Заблуждения

- Сетевые соединения надежны
- Латентность связи всегда нулевая
- Пропускная способность сети бесконечна
- Сетевое соединение безопасно
- Топология сети не меняется
- У вашей системы всегда 1 администратор
- Стоимость передачи данных нулевая
- Сеть гомогенна

Реальность

- Сетевые соединения не надежны и часто падают
- Латентность всегда оказывает существенное влияние на работу приложения
- Пропускная способность сети очень даже конечна
- Сетевое соединение не безопасно
- Топология сети меняется очень часто, и чем дальше узлы друг от друга, тем чаще меняется топология
- У вашей системы множество администраторов, которые делают одни и те же вещи по-разному
- Стоимость передачи данных бывает очень высока
- Сеть гетерогенна – пакеты могут разделяться, их атрибуты могут быть затерты, часть из них может быть отфильтрована