



РАСПРЕДЕЛЕННЫЕ СИСТЕМЫ

ВВЕДЕНИЕ



О СЕБЕ

- Наталья Ивлева
Natalja Ivleva
- Преподаватель TalTech Virumaa kolledž
 - Веб программирование
 - Программирование мобильных устройств
 - IoT программирование
- Email: natalja.ivleva@taltech.ee



ПРОГРАММА КУРСА

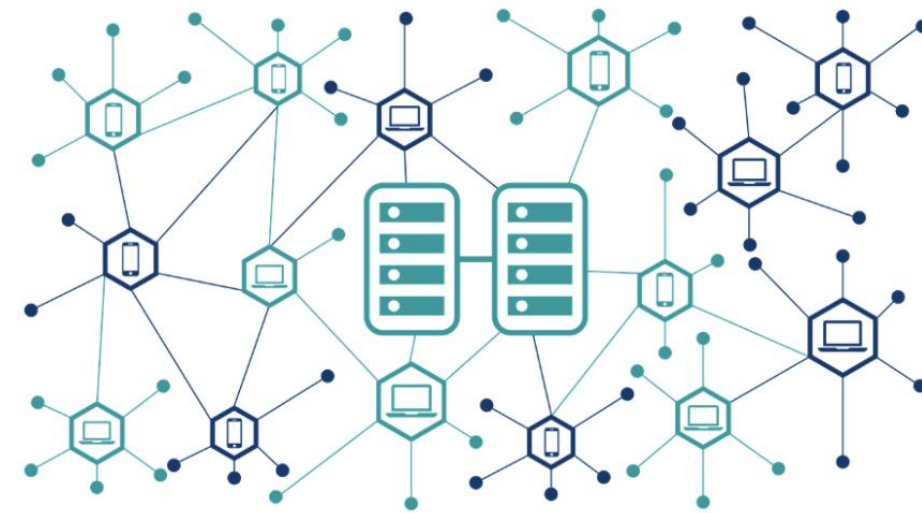
- Что такое распределенные системы
- Форматы данных
 - XML
 - JSON
 - YAML
 - ...
- SOAP

РАСПРЕДЕЛЁННЫЕ СИСТЕМЫ

- *Распределенная система* - система, в которой обработка информации сосредоточена не на одной вычислительной машине, а распределена между несколькими компьютерами.
- Распределенная система — самая распространенная — это система нескольких автономных вычислительных узлов, взаимодействующих для выполнения общей цели, или же система, состоящая из набора двух или более независимых узлов, которые координируют свою работу посредством синхронного или асинхронного обмена сообщениями. (<http://infocom.uz/2015/06/03/raspredelennye-sistemy-preimushhestva-i-ix-primenenie/>)

РАСПРЕДЕЛЁННЫЕ СИСТЕМЫ (2)

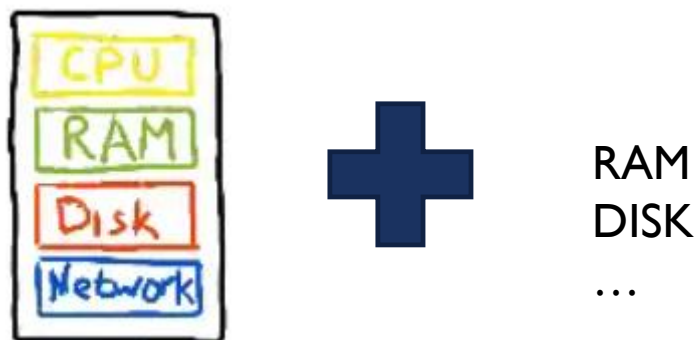
- отношения местоположений элементов играют существенную роль
 - распределение функций, ресурсов между множеством элементов
 - отсутствие единого управляющего центра
 - выход из строя одного из узлов не приводит к полной остановке всей системы
-
- Разделение ресурсов - разработка распределенной системы. Роль сервера в распределенных системах заключается в предоставлении сервисов.



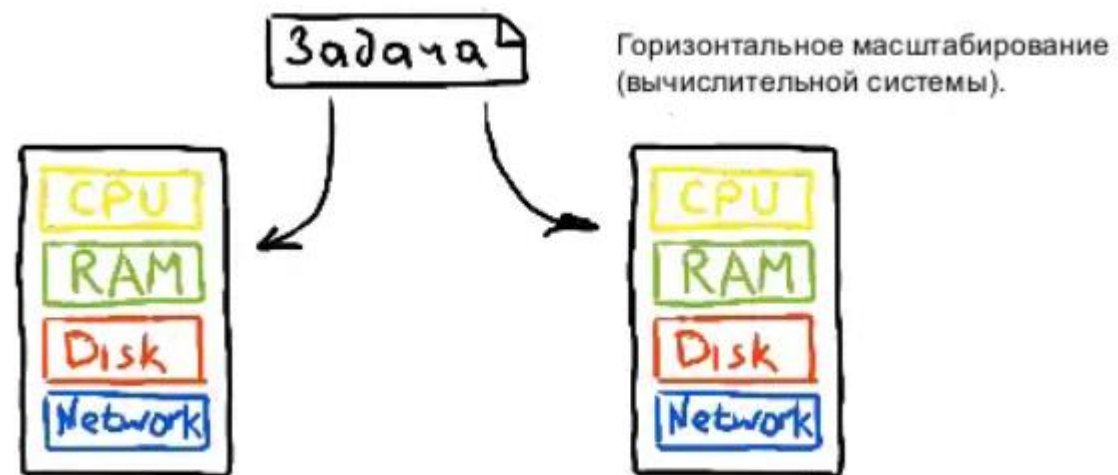
<https://medium.com/german-gorelkin/designing-distributed-systems-introduction-6086aed83bed>

МАСШТАБИРОВАНИЕ

- Вертикальное



- Горизонтальное



ОСНОВНЫЕ ХАРАКТЕРИСТИКИ

- **Совместное использование ресурсов.** Распределенные системы допускают совместное использование как аппаратных (жестких дисков, принтеров), так и программных (файлов, компиляторов) ресурсов.
- **Открытость.** Это возможность расширения системы путем добавления новых ресурсов.
- **Параллельность.** В распределенных системах несколько процессов могут одновременно выполняться на разных компьютерах в сети. Эти процессы могут взаимодействовать во время их выполнения.
- **Масштабируемость.** Под масштабируемостью понимается возможность добавления новых свойств и методов.
- **Отказоустойчивость.** Наличие нескольких компьютеров позволяет дублирование информации и устойчивость к некоторым аппаратным и программным ошибкам. Распределенные системы в случае ошибки могут поддерживать частичную функциональность. Полный сбой в работе системы происходит только при сетевых ошибках.
- **Прозрачность.** Пользователям предоставляется полный доступ к ресурсам в системе, в то же время от них скрыта информация о распределении ресурсов по системе.

НЕДОСТАТКИ

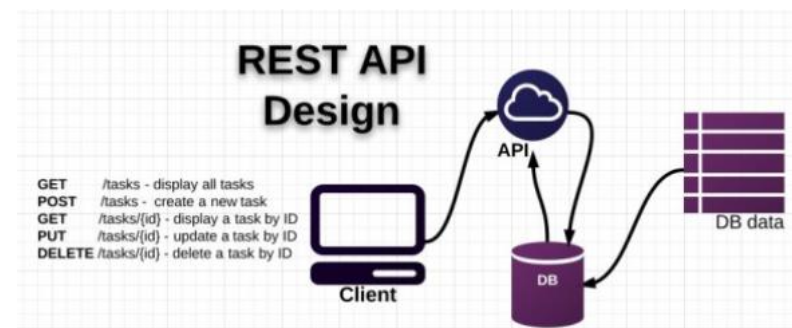
- **Идентификация ресурсов.** Ресурсы в распределенных системах располагаются на разных компьютерах, поэтому систему имен ресурсов следует продумать так, чтобы пользователи могли без труда открывать необходимые им ресурсы и ссылаться на них. Примером может служить система URL(унифицированный указатель ресурсов), которая определяет имена Web-страниц.
- **Коммуникация.** Универсальная работоспособность Internet и эффективная реализация протоколов TCP/IP в Internet для большинства распределенных систем служат примером наиболее эффективного способа организации взаимодействия между компьютерами. Однако в некоторых случаях, когда требуется особая производительность или надежность, возможно использование специализированных средств.
- **Качество системного сервиса.** Этот параметр отражает производительность, работоспособность и надежность. На качество сервиса влияет ряд факторов: распределение процессов, ресурсов, аппаратные средства и возможности адаптации системы.
- **Архитектура программного обеспечения.** Архитектура ПО описывает распределение системных функций по компонентам системы, а также распределение этих компонентов по процессорам. Если необходимо поддерживать высокое качество системного сервиса, выбор правильной архитектуры является решающим фактором.

WEB-СЕРВИСЫ

- Web-сервисы - новое слово в технологии распределенных систем.
- Web -сервис - это приложение, которое предоставляет открытый интерфейс, пригодный для использования другими приложениями в Web.
- Спецификация ONE (Спецификация *Open Net Environment*) Sun требует,
 - чтобы Web -сервисы были доступны через HTTP и другие Web -протоколы,
 - чтобы дать возможность обмениваться информацией посредством XML -сообщений
 - чтобы их можно было найти через специальные сервисы - сервисы поиска.
 - Для доступа к Web -сервисам разработан специальный протокол - *Simple Object Access Protocol (SOAP)* ,который представляет средства взаимодействия на базе XML для многих Web -сервисов.

WEB-СЕРВИСЫ (2)

- REST (сокр. от англ. Representational State Transfer-«передача состояния представления») — архитектурный стиль взаимодействия компонентов распределённого приложения в сети. REST представляет собой согласованный набор ограничений, учитываемых при проектировании распределённой гипермедиа-системы.
- Для веб-служб, построенных с учётом REST (то есть не нарушающих накладываемых им ограничений), применяют термин «RESTful».
- В отличие от веб-сервисов (веб-служб) на основе SOAP, не существует «официального» стандарта для RESTful веб-API. REST является архитектурным стилем, в то время как SOAP является протоколом. Несмотря на то, что REST не является стандартом сам по себе, большинство RESTful-реализаций используют стандарты, такие как HTTP, URL, JSON и XML.



ПРИМЕР

<http://www.omdbapi.com/> - информация о фильмах

OMDb API

The Open Movie Database

The OMDb API is a RESTful web service to obtain movie information, all content and images on the site are contributed and maintained by our users.

If you find this service useful, please consider making a [one-time donation](#) or [become a patron](#).



Poster API

The Poster API is only available to patrons.

Currently over 280,000 posters, updated daily with resolutions up to 2000x3000.

API

- Application Programming Interface — программный интерфейс приложения
 - фрагмент программного обеспечения с определённой функцией,
 - сервер целиком, приложение целиком или же просто отдельную часть приложения.
- API — это часть программного обеспечения, которое подключает одно приложение непосредственно к данным и службам другого, предоставляя ему доступ к определенным частям сервера.
- Каждый раз, когда пользователь посещает какую-либо страницу в сети, он взаимодействует с API удалённого сервера. API — это составляющая часть сервера, которая получает запросы и отправляет ответы.
- API-интерфейсы позволяют обмениваться данными между двумя приложениями, и они являются основой для большинства современных приложений. Они позволяют оптимизировать ИТ-архитектуру, упростить обмен наборами данных.

SOAP

- SOAP (Simple Object Access Protocol — простой протокол доступа к объектам) — это отдельный протокол, который немного сложнее, поскольку определяет больше стандартов, чем REST, — такие вещи, как безопасность и способ отправки сообщений.
- Причины, по которым вы можете захотеть разработать приложение с SOAP API, включают более высокий уровень безопасности (например, мобильное приложение, взаимодействующее с банком), приложения для обмена сообщениями, которым требуется надежная связь, связь с устаревшими системами или соответствие требованиям ACID.

ПРИМЕР SOAP

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:sch="http://www.soapexample.com/xml/users">
  <soapenv:Header/>
  <soapenv:Body>
    <sch:UserDetailsRequest>
      <sch:name>John</sch:name>
    </sch:UserDetailsRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Запрос



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header/>
  <soapenv:Body>
    <ns2:UserDetailsResponse xmlns:ns2="http://www.soapexample.com/xml/users">
      <ns2:User>
        <ns2:name>John</ns2:name>
        <ns2:age>5</ns2:age>
        <ns2:address>Greenville</ns2:address>
      </ns2:User>
    </ns2:UserDetailsResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Ответ

РАЗЛИЧИЯ REST И SOAP

- SOAP — это формат протокола, основанный на XML, REST — это архитектурный подход.
 - Пример реализации **RESTful**: JSON через HTTP
 - Пример реализации **SOAP**: XML поверх SOAP через HTTP
- Специфика SOAP — это формат обмена данными. С SOAP это всегда SOAP-XML, который представляет собой XML, включающий:
 - Envelope (конверт) – корневой элемент, который определяет сообщение и пространство имен, использованное в документе,
 - Header (заголовок) – содержит атрибуты сообщения, например: информация о безопасности или о сетевой маршрутизации,
 - Body (тело) – содержит сообщение, которым обмениваются приложения,
 - Fault – необязательный элемент, который предоставляет информацию об ошибках, которые произошли при обработке сообщений. И запрос, и ответ должны соответствовать структуре SOAP.
- Специфика REST — использование HTTP в качестве транспортного протокола. Он подразумевает наилучшее использование функций, предоставляемых HTTP — методы запросов, заголовки запросов, ответы, заголовки ответов и т. д.

РАЗЛИЧИЯ REST И SOAP (2)

Формат обмена сообщениями

- В SOAP вы используете формат SOAP XML для запросов и ответов.
- В REST такого фиксированного формата нет. Вы можете обмениваться сообщениями на основе XML, JSON или любого другого удобного формата. JSON является самым популярным среди используемых форматов.

Определения услуг

- SOAP использует **WSDL** (Web Services Description Language) — язык описания веб-сервисов и доступа к ним, основанный на языке XML.
- REST не имеет стандартного языка определения сервиса. Несмотря на то, что WADL был одним из первых предложенных стандартов, он не очень популярен. Более популярно использование Swagger или Open API.

РАЗЛИЧИЯ REST И SOAP (3)

Транспорт

- SOAP не накладывает никаких ограничений на тип транспортного протокола. Вы можете использовать либо Web протокол HTTP, либо MQ.
- REST подразумевает наилучшее использование транспортного протокола HTTP

Простота реализации

- RESTful веб-сервисы, как правило, гораздо проще реализовать, чем веб-сервисы на основе SOAP.

REST обычно использует JSON, который легче анализировать и обрабатывать. В дополнение к этому, REST не требует наличия определения службы для предоставления веб-службы.

- Однако в случае SOAP вам необходимо определить свой сервис с использованием WSDL, и при обработке и анализе сообщений SOAP-XML возникают большие накладные расходы.

ВОПРОСЫ?