Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Ciências de Dados e Big Data

Disciplina: Ciências de Dados e Big Data

Horário:

Quinta-Feira – 19:00 – 22:40

Professor :Eduardo Savino Gomes

Email:eduardo.Savino@fecap.br

Ciências de Dados e Big Data

EMENTA: Estudo das atividades que tem por objetivo a coleta, armazenagem, organização, administração, governança e entrega de grandes volumes de dados, tendo em vista assegurar um alto nível de qualidade e **torná-los acessíveis para as aplicações nas áreas de inteligência de negócios e análise de dados**.

Ciências de Dados e Big Data

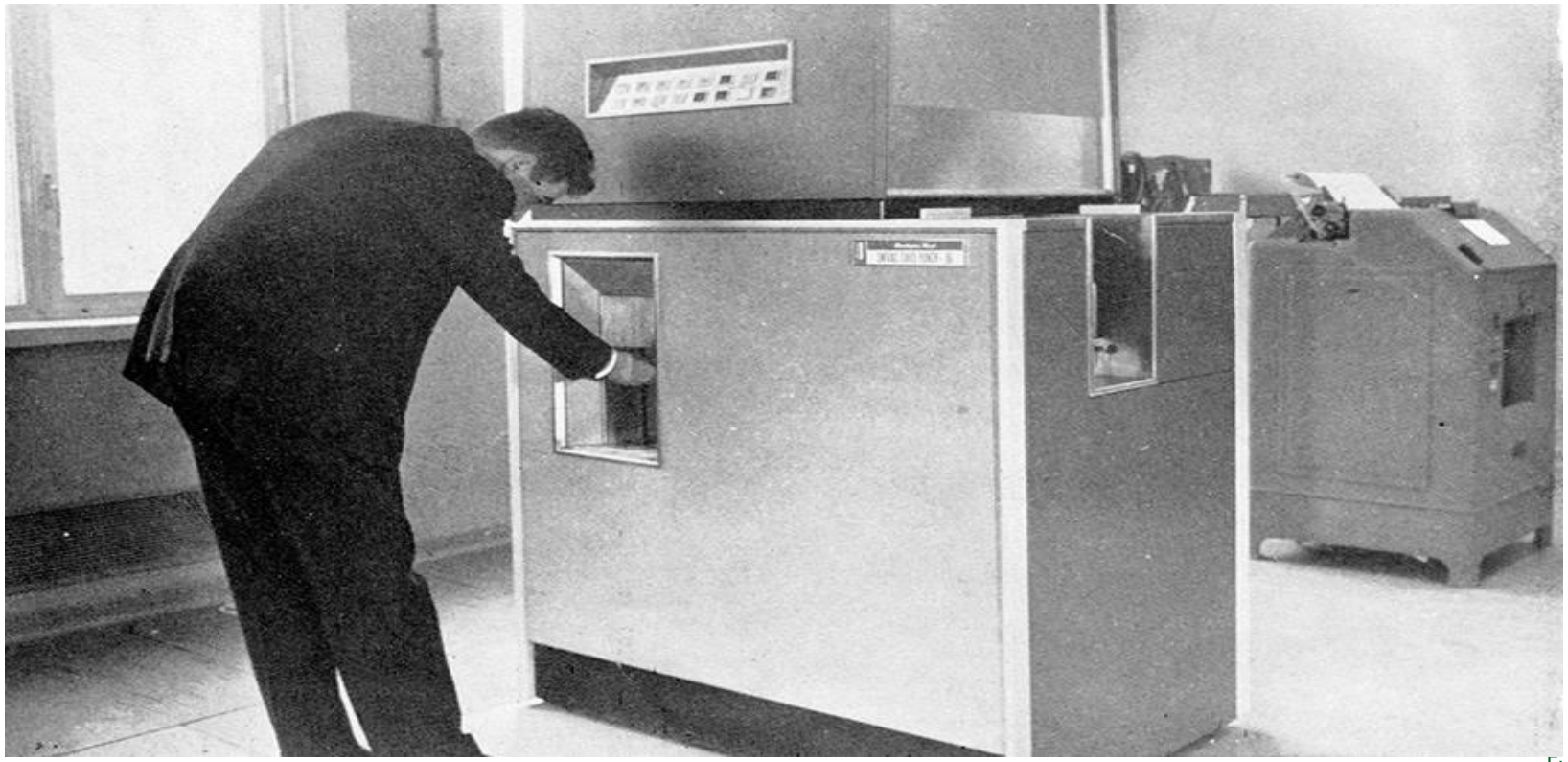
AGENDA

- SISTEMA DISTRIBUÍDO
- SISTEMA DE ARQUIVOS DISTRIBUÍDOS
- HADOOP
- MAP REDUCE

SISTEMA DISTRIBUÍDO

Evolução histórica

- Computadores iniciais: caros e grandes
  - execução por um operador: setup do job (carregar cartões), executar programa, imprimir resultado



SISTEMAS DISTRIBUÍDOS

Evolução histórica

- **Anos 50 e 60**: batching, spooling, multiprogramação
  - batching: juntar jobs semelhantes para processamento
  - spooling: sobreposição de I/O e CPU
  - multiprogramação: diversos programas sendo executados concorrentemente pela CPU
  - Objetivo: otimizar a utilização da CPU
  - Não existia a interação entre usuário e computador

SISTEMAS DISTRIBUÍDOS

Evolução histórica

- **Início dos anos 60**: sistemas de time sharing
  - utilização de diversos terminais "burros" conectados a um computador
  - impressão de um computador por usuário
  - tarefas principais/comuns são executadas pelo computador principal
    - desenvolvimento dos minicomputadores: menores e mais rápidos!
  - 1o. passo na direção dos sistemas distribuídos!
    - compartilhamento de recursos
    - acesso remote

- Terminais e computador muito próximos

SISTEMAS DISTRIBUÍDOS

Evolução histórica

- **Final dos anos 60 e início dos anos 70**: surgimento das redes de computadores e do sistema operacional UNIX
  - Ethernet – Xerox Palo Alto (1973): Local Area Network
    - permitiu interligar mais computadores a distâncias maiores usando uma velocidade maior (e.g. rede de computadores de um prédio)
  - ARPANet – DoD (1969): Wide Area Network
    - interligação entre computadores localizados dispersamente (cidades e/ou países diferentes)

SISTEMAS DISTRIBUÍDOS

Evolução histórica

- Final dos **anos 70**: protocolo TCP/IP
  - definição de padrão para comunicação entre computadores

- **Início dos anos 80**: microprocessadores e estações de trabalho
  - redução do custo (em relação aos mainframes)

- **Final dos anos 80**: estações de trabalho ligadas em rede
  - diversos serviços para comunicação entre pessoas/máquinas
    - FTP, TELNET, MAIL

SISTEMAS DISTRIBUÍDOS

Evolução histórica

SISTEMAS DISTRIBUÍDOS

Motivação

- **Avanços em microeletrônica**
  - processadores mais rápidos e baratos
- **Avanços em comunicações**
  - redes mais eficientes e confiáveis
- **Popularidade das redes de computadores**
  - redes de telefones celulares, redes corporativas, redes caseiras
  - redes de computadores de alta velocidade (Myrinet ~2Gb/s)

Inicialmente: Recursos caros, escassos e pouco eficiente
Depois: Recursos baratos, mais eficiente e (muitas vezes) ociosos

- **Compartilhamento de recursos**
  - Componentes de HW (disco, impressora)
  - SW (arquivos, bases de dados, programas)
  - Outros (vídeo, áudio)

- **Relação custo/desempenho**
  - melhor utilizar diversos processadores interconectados do que um único computador centralizado

Ciências de Dados e Big Data

SISTEMAS DISTRIBUÍDOS

- O que é um Sistema Distribuído?
  - É um sistema onde os componentes de HW e SW, localizados em **computadores interligados por uma rede**, comunicam e coordenam suas ações somente através de **troca de mensagens** [Coulouris].
  - Coleção de computadores que **não compartilham memória ou relógio físico comum**, que se comunicam por **mensagens** sobre uma **rede de comunicação**, e cada computador possui sua **própria memória** e executa seu próprio sistema operacional. Tipicamente são semi-autônomos e fracamente acoplados enquanto **cooperam** para resolver um problema coletivamente (Tanenbaum / Van Steen).

- Dois aspectos:
  - Hardware: autonomia
  - Software: sistema único

Tipos de Sistemas Distribuídos

1- Sistemas de computação distribuídos
2- Sistemas de informação distribuídos
3- Sistemas Pervasivos

1- Sistemas de Computação Distribuídos

Cluster Computing

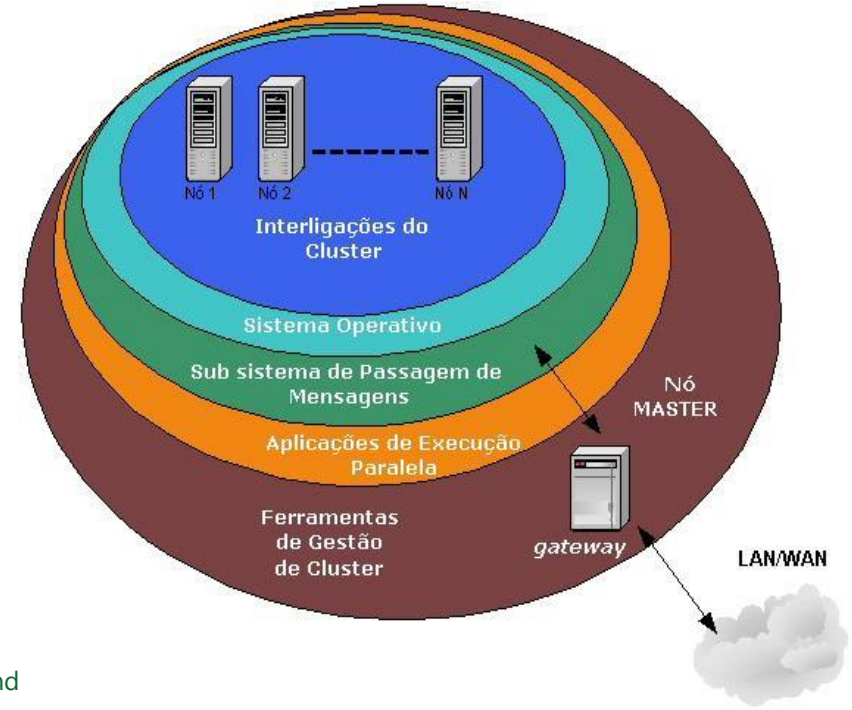
- Tornaram-se populares pela razão preço/desempenho.
- Estações mais potentes e mais baratas
- Rede melhor
- Computação intensiva paralela.

Cluster Computing

Beowulf é um projeto para aglomerados de computadores (ou Clusters) para computação paralela, usando computadores pessoais, não especializados e portanto mais baratos. O projeto foi criado por Donald Becker da NASA, e são utilizados em todo mundo, por exemplo no processamento de dados com finalidade científica e na renderização de filmes de animação 3D.

- Mestre
  - Alocação de nós / fila / escalonamento
  - Interface para usuário
  - Executa middleware para execução de programas e gerência do Cluster.

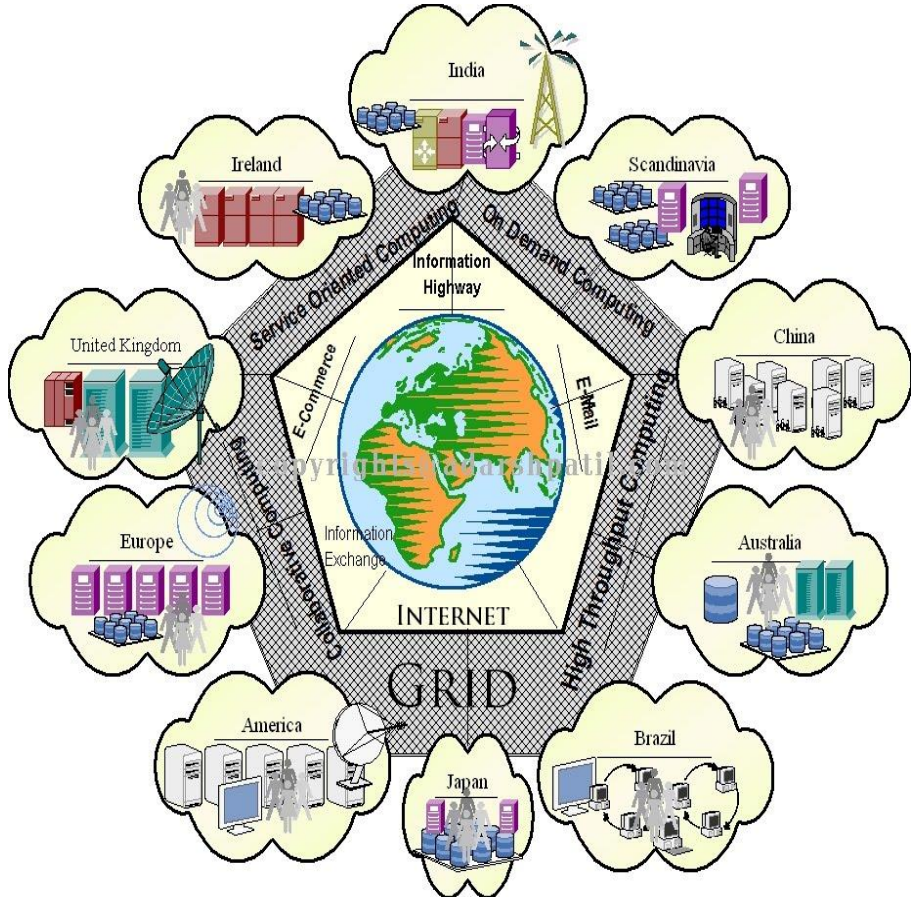
- Bibliotecas de execução em sistemas paralelos: facilidades para comunicação por troca de mensagem.
- Nós de computação: SO padrão pode ser suficiente.
- Migração de processos: movimento transparente de processos de um nó inativo para qualquer outro nó.



Computação em grade (Grid Computing)

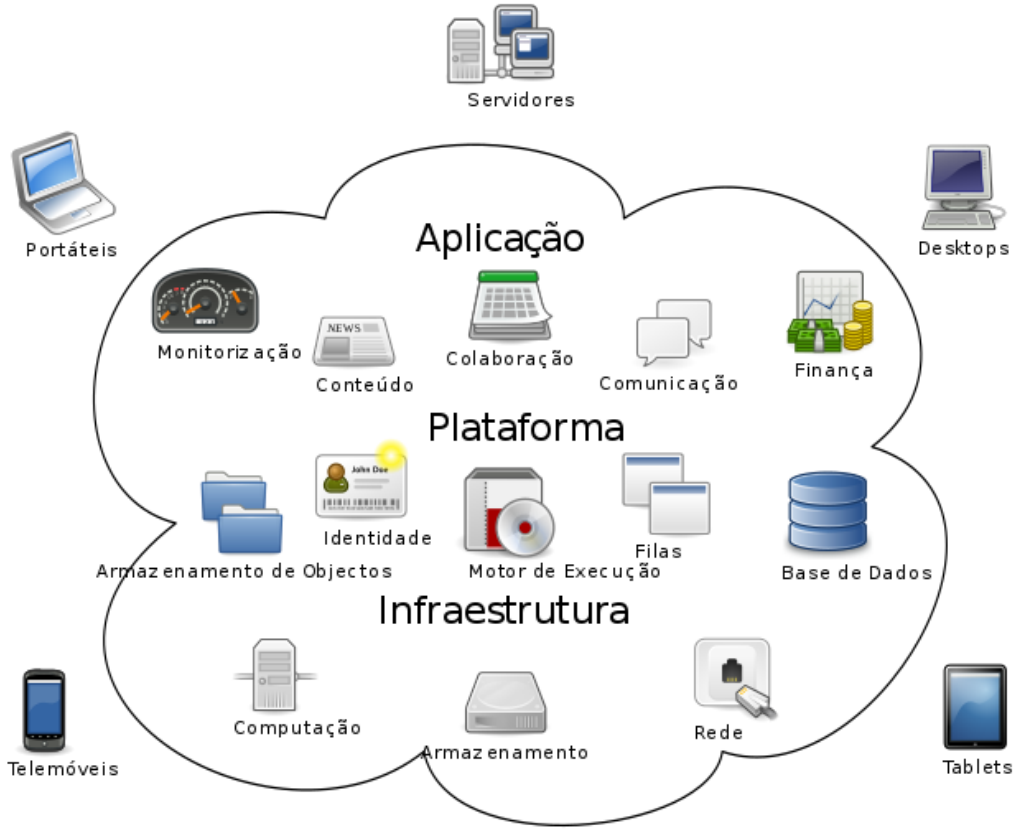
Cluster: homogêneo. X Grade: heterogênea

- Nenhuma premissa adotada em relação a hardware, S.O., rede, domínio administrativo, política de segurança, etc.
- Recursos de diferentes organizações reunidos para permitir colaboração.
- Organização virtual: pessoas em uma organização virtual têm direitos sobre recursos dessa organização.
- Servidores de computação (inclusive clusters), armazenamento, instrumentos, bancos de dados, equipamentos, sensores, telescópios, etc.



Cloud Computing

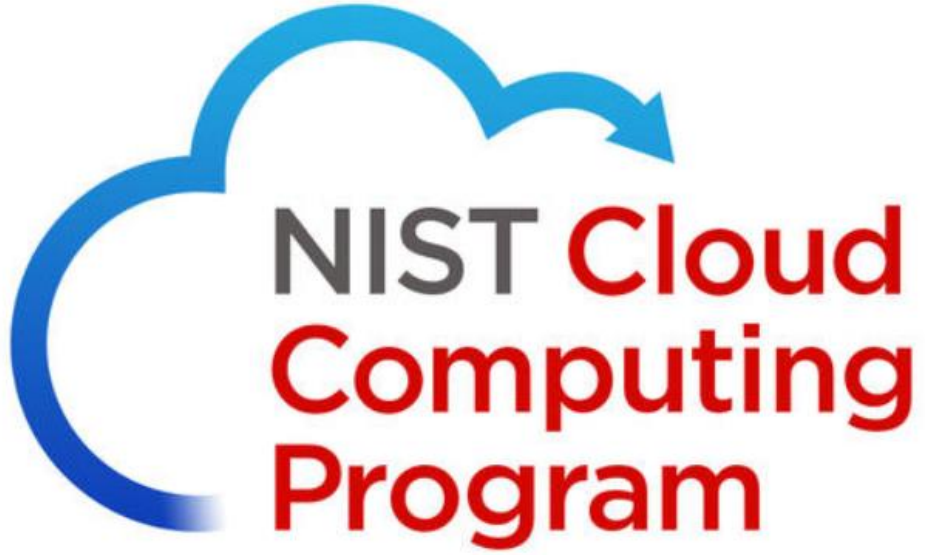
- Introduz noção de “computação como serviço”.
- Relação com *utility grids*.
- Recursos virtualizados.
- Diferentes níveis de serviço. IaaS, PaaS, SaaS.
- Pagamento pelo uso.
- Nuvem pública / privada / híbrida / comunitária.
- Evita alto investimento inicial.
- Visões: provedor e cliente.



Computação em nuvem

Definição

A **computação em nuvem** é um modelo para permitir acesso conveniente à rede sob demanda a um pool compartilhado de recursos de computação configuráveis (por exemplo, redes, servidores, armazenamento, aplicativos e serviços) que podem ser rapidamente provisionados e liberados com o mínimo esforço de gerenciamento ou interação do provedor de serviços.



NIST Cloud Computing Program - NCCP | NIST (National Institute of Standards and Technology-USA)

Definição

Este modelo de nuvem promove a disponibilidade e é composto por cinco características essenciais:

- autoatendimento sob demanda,
- acesso à rede ampla,
- pooling de recursos,
- elasticidade rápida,
- Serviço Medido

Três modelos de serviço:

- Software em Nuvem como Serviço (SaaS),
- Cloud Platform as a Service (PaaS),
- Cloud Infrastructure as a Service (IaaS)); e,

Quatro modelos de implantação:

-Nuvem privada,
-Nuvem comunitária,
-Nuvem pública,
-Nuvem híbrida.

As principais tecnologias de habilitação incluem:

- (1) redes rápidas de ampla área,
- (2) computadores de servidor poderosos e baratos e
- (3) virtualização de alto desempenho para hardware de *commodities.*

NIST Cloud Computing Program - NCCP | NIST (National Institute of Standards and Technology-USA)

Ciências de Dados e Big Data

Características de Computação em Nuvem

Ciências de Dados e Big Data

Características de Computação em Nuvem

No SP 800-145, o NIST especifica que uma infraestrutura em nuvem deve ter as cinco características principais :

- *Autoatendimento sob demanda: "Um consumidor pode provisionar unilateralmente capacidades de computação, como tempo do servidor ou armazenamento em rede, automaticamente conforme a necessidade, sem exigir interações humanas com cada provedor de serviços." – NIST*
- *Acesso amplo via rede: "Capacidades que estão disponíveis na rede e são acessadas através de mecanismos padrão que promovem o uso por plataforma client thin ou thick client heterogênea (por exemplo, telefones móveis, tablets, laptops e estações de trabalho)." – NIST*
- *Pool de recursos: "Os recursos de computação do provedor são colocados em um pool para servir vários consumidores usando um modelo multi-tenant (multi-inquilino), com recursos físicos e virtuais diferentes atribuídos e reatribuídos dinamicamente de acordo com a demanda do consumidor. Há um sentido de independência de localização em que o consumidor geralmente não tem controle ou conhecimento sobre o local exato dos recursos fornecidos, mas pode especificar o local em um nível de abstração mais alto (por exemplo, país, estado ou datacenter). Exemplos de recursos que incluem armazenamento, processamento, memória e largura de banda de rede." – NIST*

Ciências de Dados e Big Data

Características de Computação em Nuvem

• *Elasticidade rápida: "As capacidades podem ser provisionadas rápida e elasticamente, em alguns casos automaticamente, para escalar rápida e proporcionalmente para fora e para dentro com a demanda. Para o consumidor, as capacidades disponíveis para provisionamento geralmente parecem ser ilimitadas e podem ser apropriadas em qualquer quantidade e a qualquer momento." – NIST*

• *Serviço medido: "Os sistemas de nuvem automaticamente controlam e otimizam o uso de recursos aproveitando uma capacidade de medição em algum nível de abstração apropriado para o tipo de serviço (por exemplo, armazenamento, processamento, largura de banda e contas de usuário ativo). O uso de recursos pode ser monitorado, controlado e reportado, oferecendo transparência para o provedor e para o consumidor do serviço utilizado." – NIS*

Modelos de serviço em nuvem

Modelos de serviço em nuvem

IaaS (Infraestrutura como serviço) "A capacidade fornecida ao consumidor para provisionar recursos de processamento, armazenamento, redes e outros recursos de computação fundamentais em que o consumidor pode implementar e executar software arbitrário, o que inclui sistemas operacionais e aplicativos. O consumidor não gerencia nem controla a infraestrutura de base da nuvem, mas tem controle sobre os sistemas operacionais, armazenamento e aplicativos implementados; e possivelmente controle limitado de componentes do sistema de rede selecionado (por exemplo, host firewalls)." – NIST

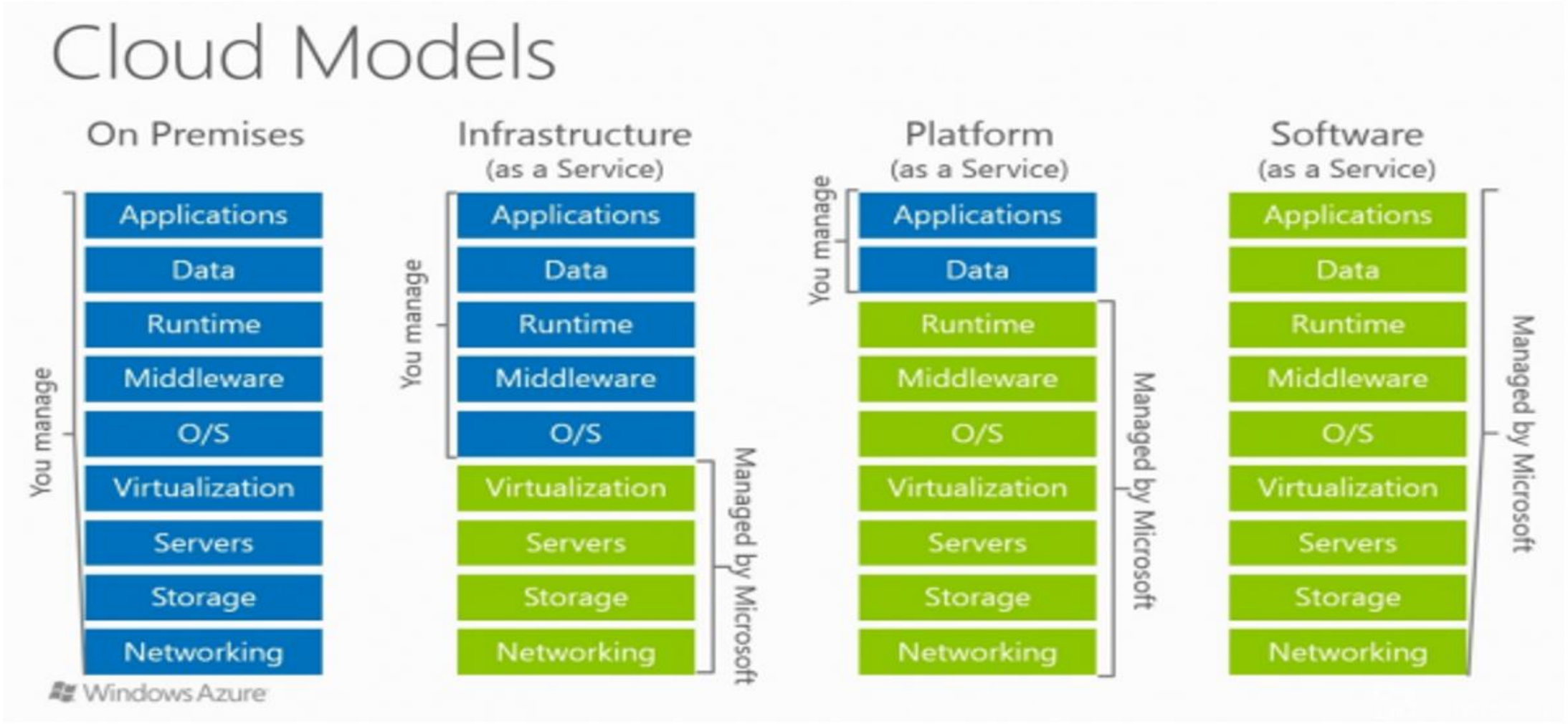
Modelos de serviço em nuvem

PaaS(Plataforma como serviço): "Capacidade fornecida ao consumidor para implementar na infraestrutura de nuvem aplicativos criados ou adquiridos pelo consumidor usando linguagens de programação, bibliotecas, serviços e ferramentas suportadas pelo provedor. O consumidor não gerencia nem controla a infraestrutura de nuvem subjacente que inclui rede, servidores, sistemas operacionais ou armazenamento, mas controla os aplicativos implementados e possivelmente as configurações do ambiente de hospedagem do aplicativo." – NIST

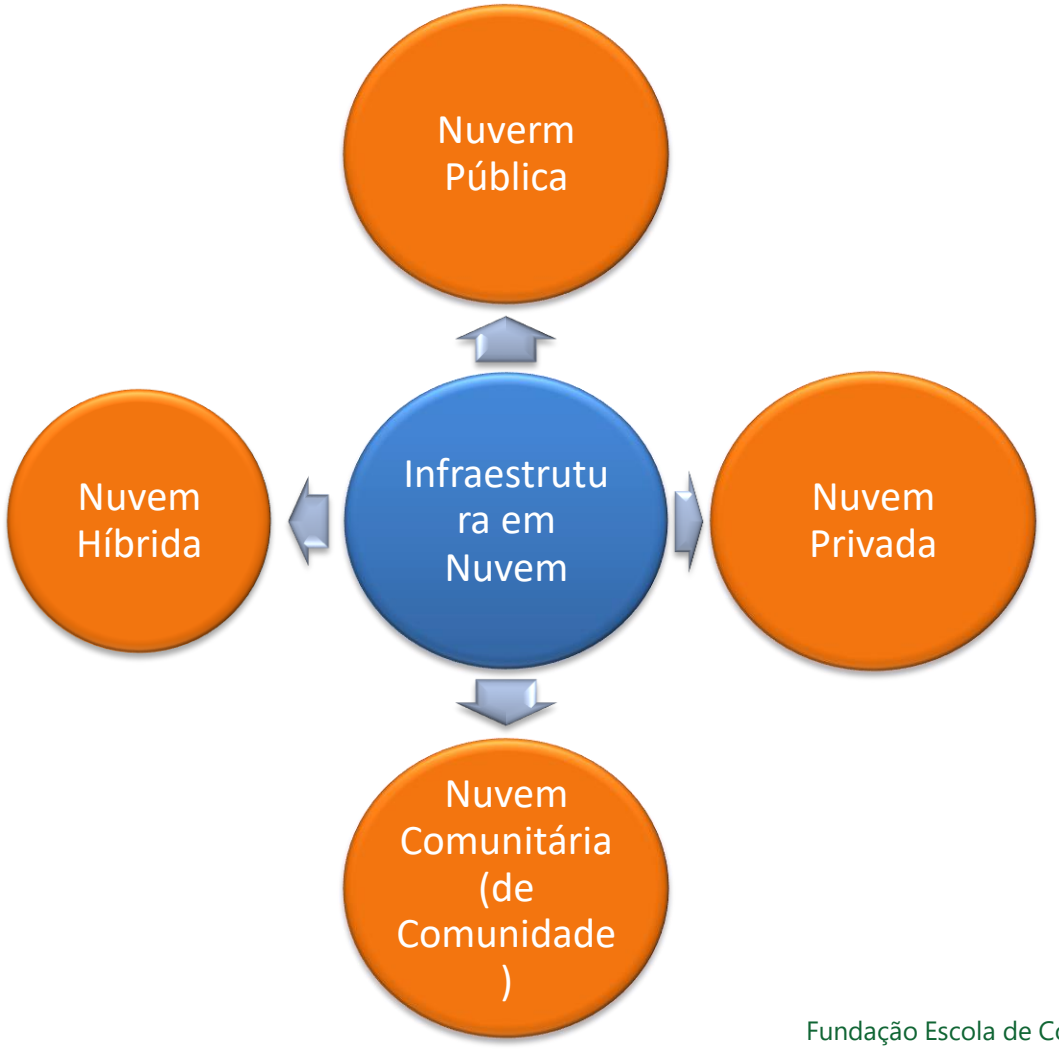
Modelos de serviço em nuvem

SaaS(Software como serviço): "Capacidade fornecida ao consumidor para utilizar os aplicativos do provedor que são executados em uma infraestrutura de nuvem. Os aplicativos são acessíveis por vários dispositivos cliente através de uma interface de cliente thin, como um navegador da Web, (por exemplo, e-mail baseado na Web), ou uma interface de programa. O consumidor não gerencia nem controla a infraestrutura subjacente de nuvem que inclui a rede, servidores, sistemas operacionais, armazenamento ou capacidades de aplicativos individuais, com a possível exceção de configurações de aplicativo específicas de usuário limitadas." – NIST

Modelos de serviço em nuvem



Modos de Implementação



Modos de Implementação

Nuvem pública: "A infraestrutura em nuvem tem uso aberto pelo público em geral. Ela pode ser de propriedade, gerenciada e operada por um negócio, academia ou organização governamental ou alguma combinação delas. Ela existe nas instalações do provedor de nuvem." – NIST

Nuvem privada: "A infraestrutura em nuvem é para uso exclusivo por uma organização que compreenda vários clientes (por exemplo, unidades de negócios). Ela pode ser de propriedade, gerenciada e operada pela organização, por um terceiro ou por uma combinação desses e pode existir dentro e fora do local. – NIST

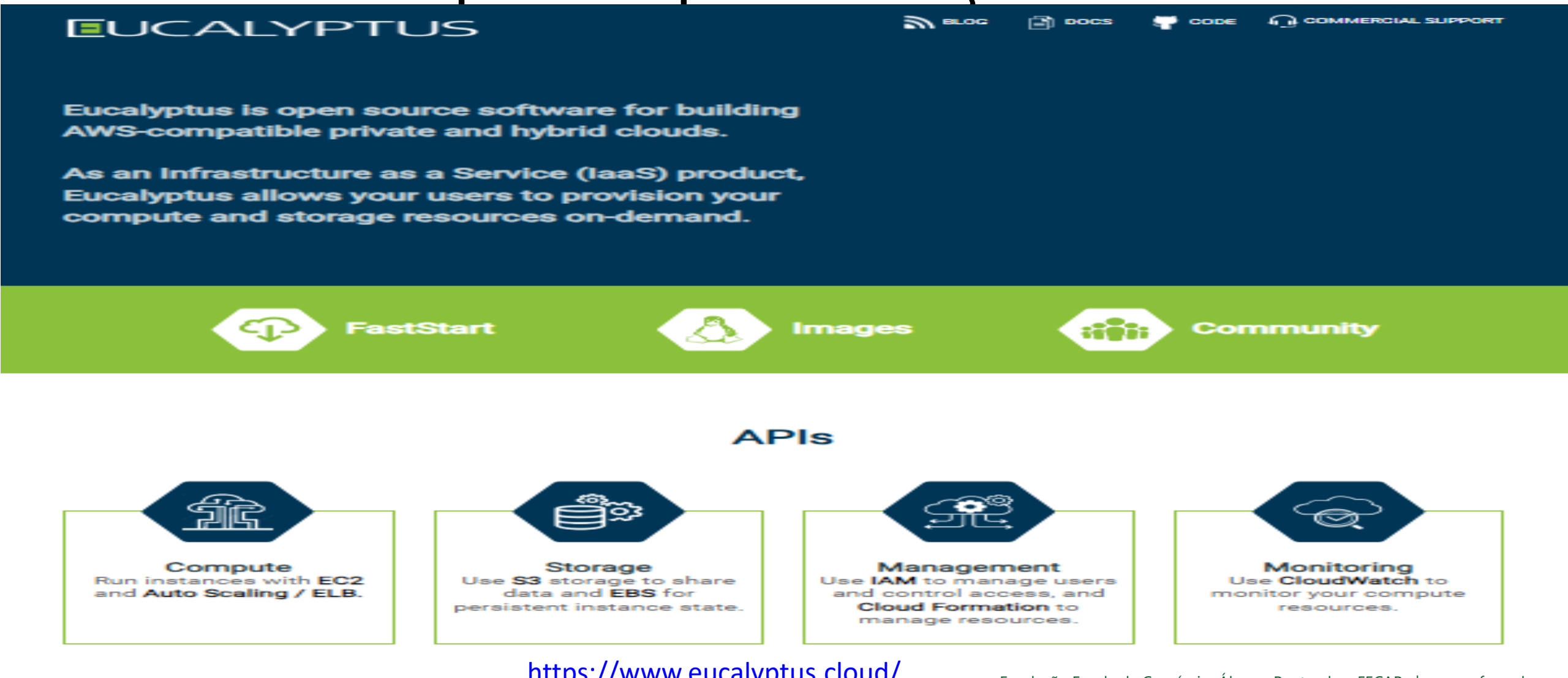
Modos de Implementação

Nuvem de comunidade: "A infraestrutura de nuvem é de uso exclusivo de uma comunidade específica de consumidores de organizações que compartilham interesses (por exemplo, missão, exigências de segurança, política e considerações de conformidade). Ela pode ser de propriedade, gerenciada e operada por uma ou mais das organizações na comunidade, por um terceiro ou por uma combinação desses e pode existir dentro e fora do local." – NIS

Nuvem híbrida: "A infraestrutura em nuvem é um composição de duas ou mais infraestruturas em nuvem diferentes (privada, comunidade ou pública) que permanecem entidades exclusivas, mas são conectadas por tecnologia padronizada ou patenteada que permite que a portabilidade de dados e aplicativos (por exemplo, pico de nuvem para balanceamento de carga entre nuvens)." – NIS

Ciências de Dados e Big Data

Software para Implementação de Nuvem



https://www.eucalyptus.cloud/

Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Ciências de Dados e Big Data

Software para Implementação de Nuvem

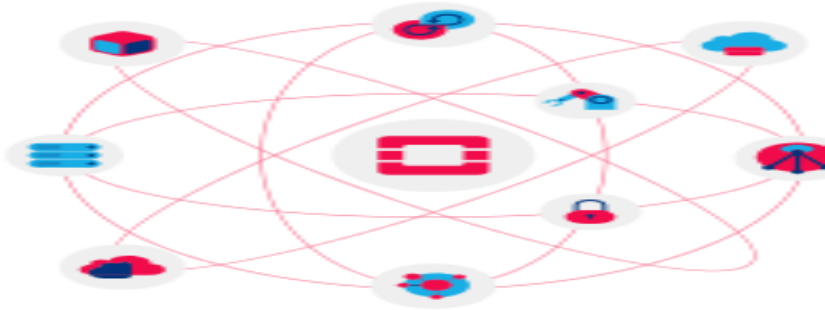
openstack. | SEARCH | SOFTWARE | USE CASES | EVENTS | COMMUNITY | MARKETPLACE | BLOG | DOCS | JOIN | LOG IN

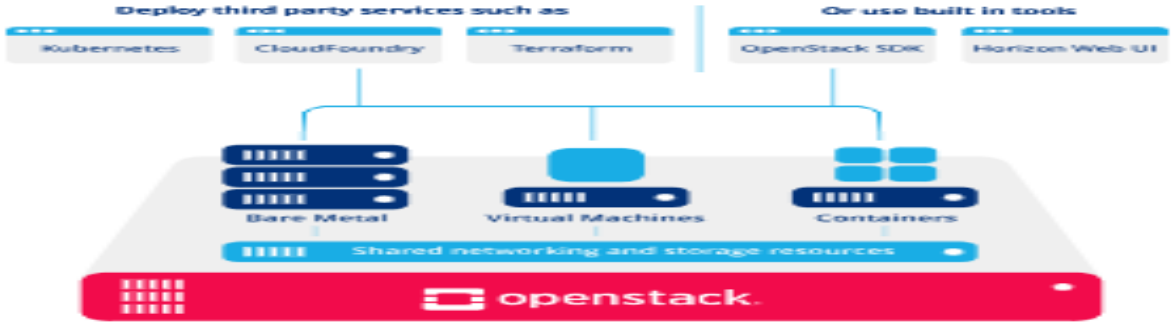
The Most Widely Deployed Open Source Cloud Software in the World

Deployed by thousands. Proven production at scale. OpenStack is a set of software components that provide common services for cloud infrastructure.

BROWSE OPENSTACK COMPONENTS

OpenStack is developed by the community. For the community. **Learn how to contribute →**





Cloud Infrastructure for Virtual Machines, Bare Metal, and Containers

Openstack controls large pools of compute, storage, and networking resources, all managed through APIs or a dashboard.

Beyond standard infrastructure-as-a-service functionality, additional components provide orchestration, fault management and service management amongst other services to ensure high availability of user applications.

READ MORE

On-Premises

Host your cloud infrastructure internally or find an OpenStack partner in the Marketplace

Public Cloud

Leverage one of the 180+ OpenStack powered public cloud data centers

At the Edge

Telecoms and retailers rely on OpenStack for their distributed systems

https://www.openstack.org/

Software para Implementação de Nuvem

Canonical | We are hiring | Products

ubuntu® | Enterprise | Developer | Community | Download | Search | Sign in

OpenStack | What is OpenStack | Features | Managed | Consulting | Install | Support

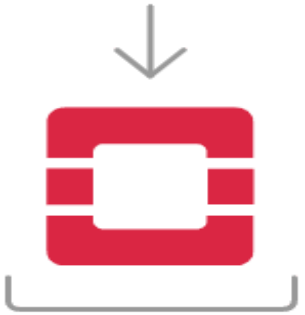
Install OpenStack yourself

Try OpenStack on your workstation or a VM. Set up a small-scale cloud or deploy a large cluster across hundreds of physical machines. Use MicroStack or Charmed OpenStack depending on your use case.

If you need OpenStack consulting or a fully managed service, talk to an expert.

Install OpenStack

Download guide to OpenStack for Beginners ›



https://ubuntu.com/openstack/install

2-Sistemas de informação distribuídos

- Organizações se defrontaram com uma profusão de aplicações em rede.
  - Interoperabilidade complicada.
- Algumas soluções de middleware existentes são resultado da integração de tais aplicações em um sistema empresarial
- Mais fácil que desenvolver todas novamente.
- Integração em vários níveis.
- Aplicação (servidor + banco de dados) disponibilizada a clientes remotos.
- Cliente envia requisição, recebe resposta.
- Integração em nível mais baixo: clientes poderiam empacotar várias requisições para diferentes servidores em uma única requisição maior.
  - Envio para execução em forma de ***transação distribuída***.
    - Idéia fundamental: ou todas ou nenhuma seria executada.
- Ex.: reserva passagem, hotel, aluguel de automóvel, restaurante (groupon), passagem de trem.

3-Sistemas distribuídos pervasivos

- SDs até agora: estáveis / nós fixos e conexões, até certo ponto, permanentes.
- Alcançado através de diversas técnicas (p.ex. mascarar falhas, esconder localização, etc)
- Dispositivos móveis e embarcados: instabilidade é o comportamento esperado.
- Sistemas distribuídos pervasivos (espalhado; difuso...)
- Características: equipamentos pequenos, bateria, mobilidade.

- Inerentemente distribuído.
- Configurados pelos proprietários.
  - Descobrir ambiente automaticamente é desejável.
  - Encaixar-se no ambiente onde está.

- Requisitos:
  - Aceitar/adotar mudanças de contexto.
  - Encorajar composição ad hoc. (Sistema se organiza de acordo com a finalidade)
  - Reconhecer compartilhamento como comportamento padrão.

3-Sistemas distribuídos pervasivos

Adotar mudanças de contexto:

- Dispositivo ciente de que seu ambiente pode mudar continuamente.
  - Ex.: descobrir que rede não está disponível porque usuário está se movimentando. Conectar-se a outra rede ou tomar providências adequadas.

- Incentivar composição ad-hoc
  - Dispositivos utilizados de formas diferentes por usuários diferentes
  - Configuração, automática ou não, de aplicações tem que ser fácil
  - Compartilhamento como padrão
  - Dispositivos se comunicam para trocar informações.
  - Meios para ler, armazenar, gerenciar e compartilhar informações.
  - Conectividade intermitente e dinamicidade de dispositivos conectados: espaço de informações muda o tempo todo.

- Mobilidade necessita de suporte a adaptação fácil e dependente de aplicação a seu ambiente local.
- Descobrir serviços eficientemente e agir de acordo.
- Existe transparência de distribuição em sistemas pervasivos?
  - Distribuição de dados e controle é inerente a tais sistemas.

3-Sistemas distribuídos pervasivos

Sistemas domésticos

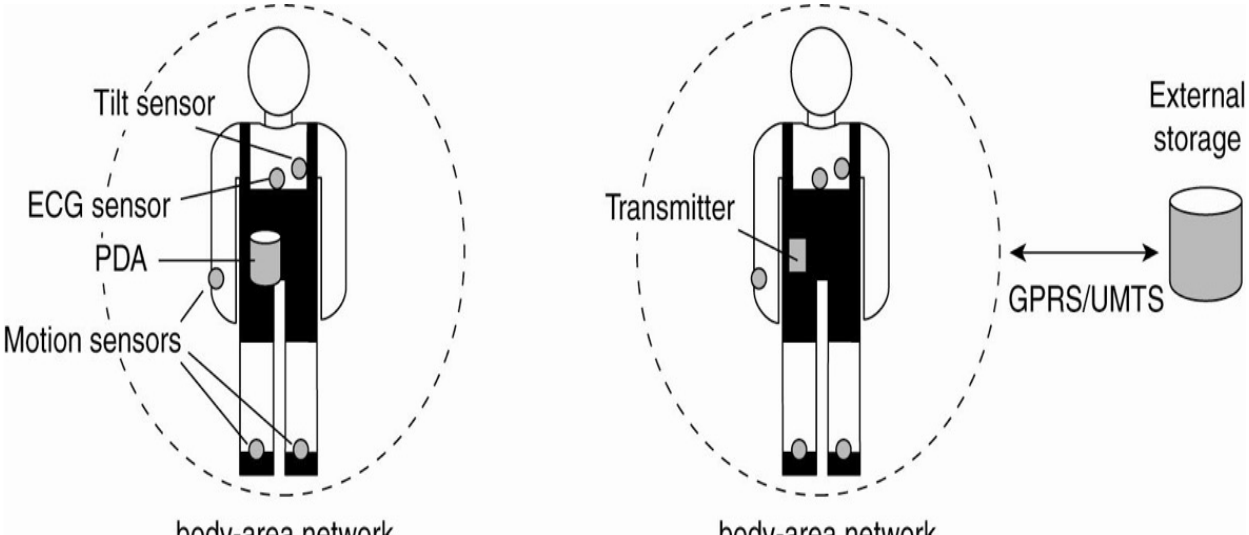
- 1 ou mais computadores pessoais, televisão, backup, celulares, geladeira, câmeras de vigilância, relógios, iluminação...
- Necessidade de autoconfiguração e autogerência
- Não assumir que usuários finais são capazes ou têm disposição para configurar e manter em funcionamento, além de corrigir falhas
- Primeiro passo: universal plug and play
- Gerenciamento do espaço pessoal
  - O que compartilhar, com qual dispositivo, sob quais circunstâncias, por quanto tempo, etc

- Haverá uma máquina que gerencia sistema doméstico?
  - Outros dispositivos só fornecem interface.
- Sistemas de recomendação.
  - Dedução do que deve ser colocado no espaço pessoal de alguém.
  - Metadados para sistemas de recomendação.

3-Sistemas distribuídos pervasivos

Sistemas de saúde distribuídos

- Dispositivos de monitoramento de saúde
- Contato automático com o médico
- Evitar hospitalização
- Vários sensores em uma rede de área corporal (body-area network – BAN)
- Preferencialmente sem fio



- Monitoramento de pessoas em um serviço de saúde pervasivo.
  - (a) concentrador local; ou
  - (b) conexão sem fio contínua.

Àrea crescente em Engenharia Biomédica

Requisitos

Transparência

Transparência do acesso: os programas clientes não devem conhecer a distribuição de arquivos.

Transparência de localização: os programas clientes devem ver um espaço de nomes de arquivos uniforme.

Transparência de mobilidade: nem os programas clientes, nem as tabelas de administração de sistema nos computadores clientes precisam ser alterados quando os arquivos são movidos.

Transparência de desempenho: os programas clientes devem continuar a funcionar satisfatoriamente,

Transparência de mudança de escala: o serviço pode ser expandido de forma paulatina, para lidar com uma ampla variedade de cargas e tamanhos de rede.

Requisitos

Atualizações concorrentes de arquivos: As alterações feitas em um arquivo por um único cliente não devem interferir na operação de outros clientes que estejam acessando (controle de concorrência).

Replicação de arquivos: Em um serviço de arquivos que suporta replicação, um arquivo pode ser representado por várias cópias de seu conteúdo em diferentes locais.

Heterogeneidade do hardware e do sistema operacional : As interfaces de serviço devem ser definidas de modo que o software cliente e servidor possa ser implementado para diferentes sistemas operacionais e computadores.

Tolerância a falhas: Essencial nos sistemas distribuídos, também no serviço de arquivo distribuído.

Requisitos

***Consistência*:** Quando os arquivos são replicados, ou armazenados em cache, em diferentes sites (servidores), há um atraso inevitável na propagação das modificações e isso pode resultar em certo desvio da semântica de cópia única.

Segurança :Praticamente todos os sistemas de arquivos fornecem mecanismos de controle baseados no uso de listas de controle de acesso. Nos sistemas de arquivos distribuídos, há necessidade de autenticar as requisições dos clientes para que o controle de acesso no servidor seja baseado nas identidades corretas de usuário e para proteger o conteúdo das mensagens de requisição-resposta com assinaturas digitais e (opcionalmente) criptografia de dados secretos.

Eficiência :Um serviço de arquivo distribuído deve oferecer recursos que tenham pelo menos o mesmo poder e generalidade daqueles encontrados nos sistemas de arquivos

Ciências de Dados e Big Data

HDFS – Hadoop Distributed File System

O **HDFS é um projeto da Apache Software Foundation** e um subprojeto do projeto Apache Hadoop (https://hadoop.apache.org/). O Hadoop é ideal para armazenar grandes quantidades de dados, do porte de terabytes e pentabytes, e usa o HDFS como sistema de armazenamento. O **HDFS permite a conexão de *nós*** (computadores pessoais padrão) contidos nos clusters **por meio dos quais os arquivos de dados são distribuídos**. É possível **acessar e armazenar os arquivos de dados como um sistema de arquivos contínuo**. O acesso aos arquivos de dados é gerenciado de um modo em *fluxo*, o que significa que aplicativos ou comandos são executados diretamente por meio do modelo de processamento **MapReduce** .

O HDFS é tolerante a falhas e disponibiliza acesso de alto rendimento a grandes conjuntos de dados.

Ciências de Dados e Big Data

HDFS – Hadoop Distributed File System

Um pouco de História

Fevereiro de 2003: Jeffrey Dean e Sanjay Ghemawat, engenheiros do Google, desenvolvem a tecnologia **MapReduce**, que possibilitou otimizar a indexação dos dados sobre as páginas Web e suas ligações.

Outubro de 2003: **Google File System** (GFS) é desenvolvido. O GFS é um sistema de arquivos distribuído criado para dar suporte ao armazenamento e processamento do grande volume de dados da tecnologia **MapReduce**

Dezembro de 2004: o Google publica o artigo **Simplified Data Processing on Large Clusters**, de autoria dos engenheiros Dean e Ghemawat, apresentando os principais conceitos e características da tecnologia MapReduce, porém, sem detalhes sobre a implementação; (https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.324.78&rep=rep1&type=pdf)

Ciências de Dados e Big Data

HDFS – Hadoop Distributed File System

Um pouco de História

Dezembro de 2005: Douglas Cutting divulga uma implementação do MapReduce e do sistema de arquivos distribuídos com base nos artigos do GFS e do MapReduce como parte do subprojeto Nutch, adotado pela comunidade de software livre para criar um motor de busca na Web. Posteriormente o Nutch seria hospedado como projeto Lucene, na **Apache Software Foundation**, tendo como principal função fornecer um poderoso mecanismo de busca e indexação de documentos.

Fevereiro de 2006: Yahoo! decide investir no projeto Nutch, mantendo o código aberto. Nesse mesmo ano, o projeto recebe o nome de **Hadoop**, passando a ser um projeto independente pertecente Apache Software Foundation

Dezembro de 2011: O Apache Hadoop disponibiliza a versão estável (a 1.0.0).

Maio de 2012: a Apache faz o lançamento da versão da 2.0 do Hadoop, incluindo alta disponibilidade no sistema de arquivos (HDFS) e melhorias no código.

HDFS – Objetivos

O HDFS tem muitos objetivos. Estes são alguns dos mais notáveis:

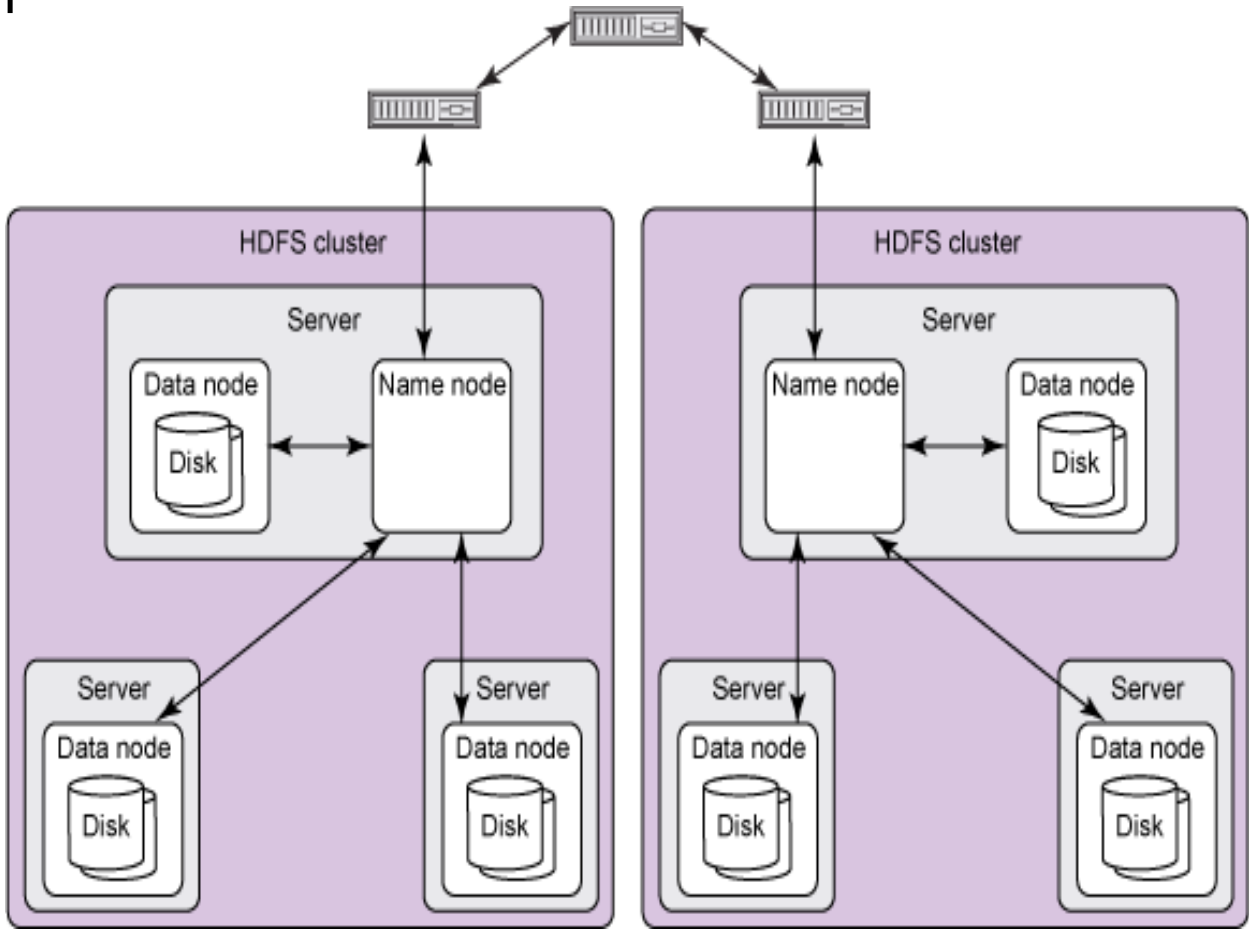
- Tolerância a falhas pela detecção de falhas e aplicação de recuperação rápida, automática
- Acesso a dados por meio do fluxo MapReduce
- Modelo de simultaneidade simples e robusto
- Lógica de processamento próxima aos dados, ao invés dos dados estarem próximos à lógica de processamento
- Portabilidade entre sistemas operacionais e hardware padrão heterogêneos

HDFS – Objetivos

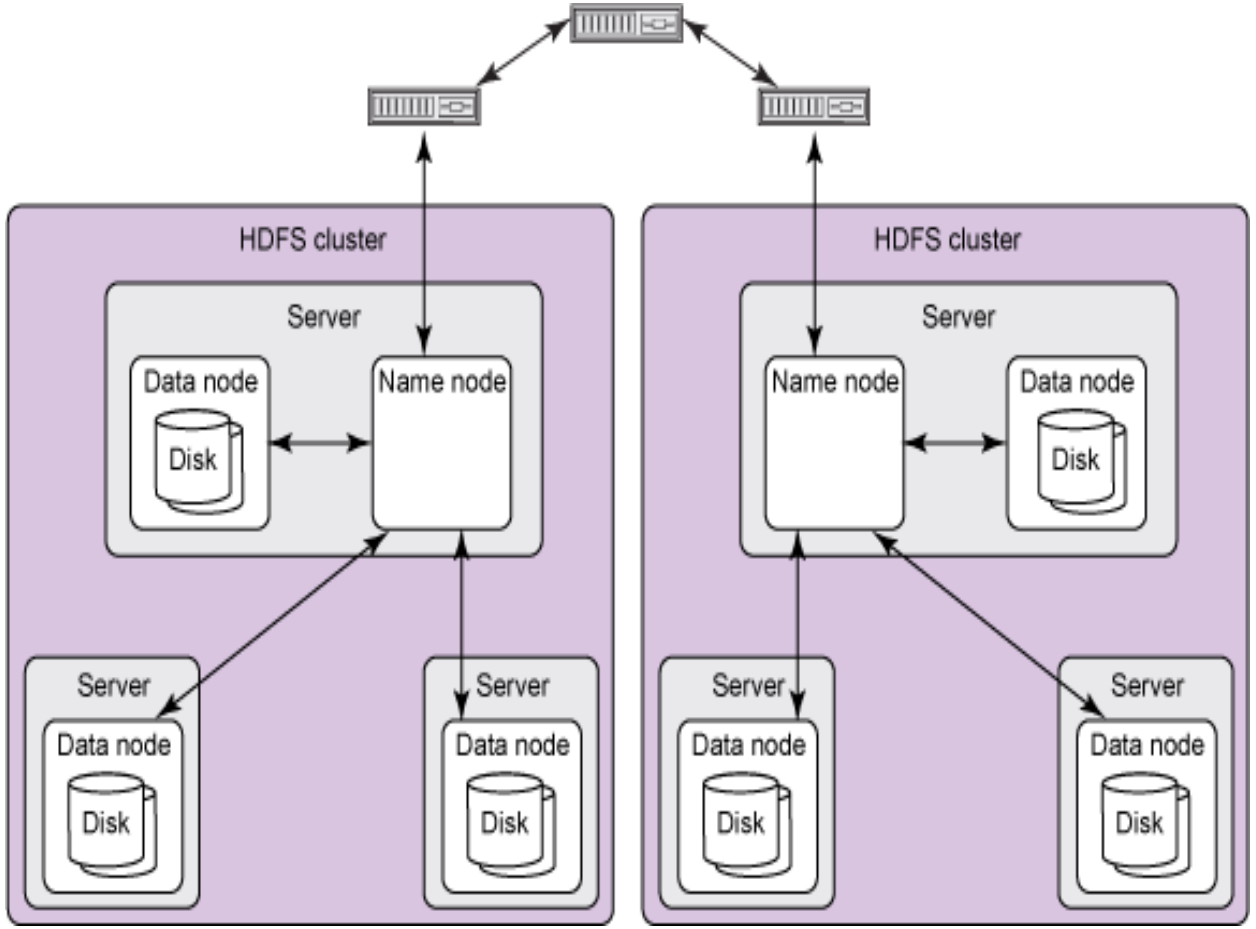
- Escalabilidade para armazenar e processar de modo confiável grandes quantidades de dados
- Economia pela distribuição de dados e pelo processamento entre clusters de computadores pessoais padrão
- Eficiência pela distribuição de dados e pela lógica para processá-los em paralelo nos nós em que os dados estão localizados
- Confiabilidade pela manutenção automática de várias cópias dos dados e pela reimplementação automática da lógica de processamento no caso de falhas

HDFS – Arquitetura

O HDFS é composto por clusters de nós interconectados no local onde os arquivos e diretórios residem. Um cluster HDFS consiste em um único nó, conhecido como um NameNode, que gerencia o namespace do sistema de arquivos e regula o acesso do cliente aos arquivos. Além disso, os nós de dados (DataNodes) armazenam dados como blocos dentro dos arquivos.



HDFS – Arquitetura



Nós de nome e nós de dados

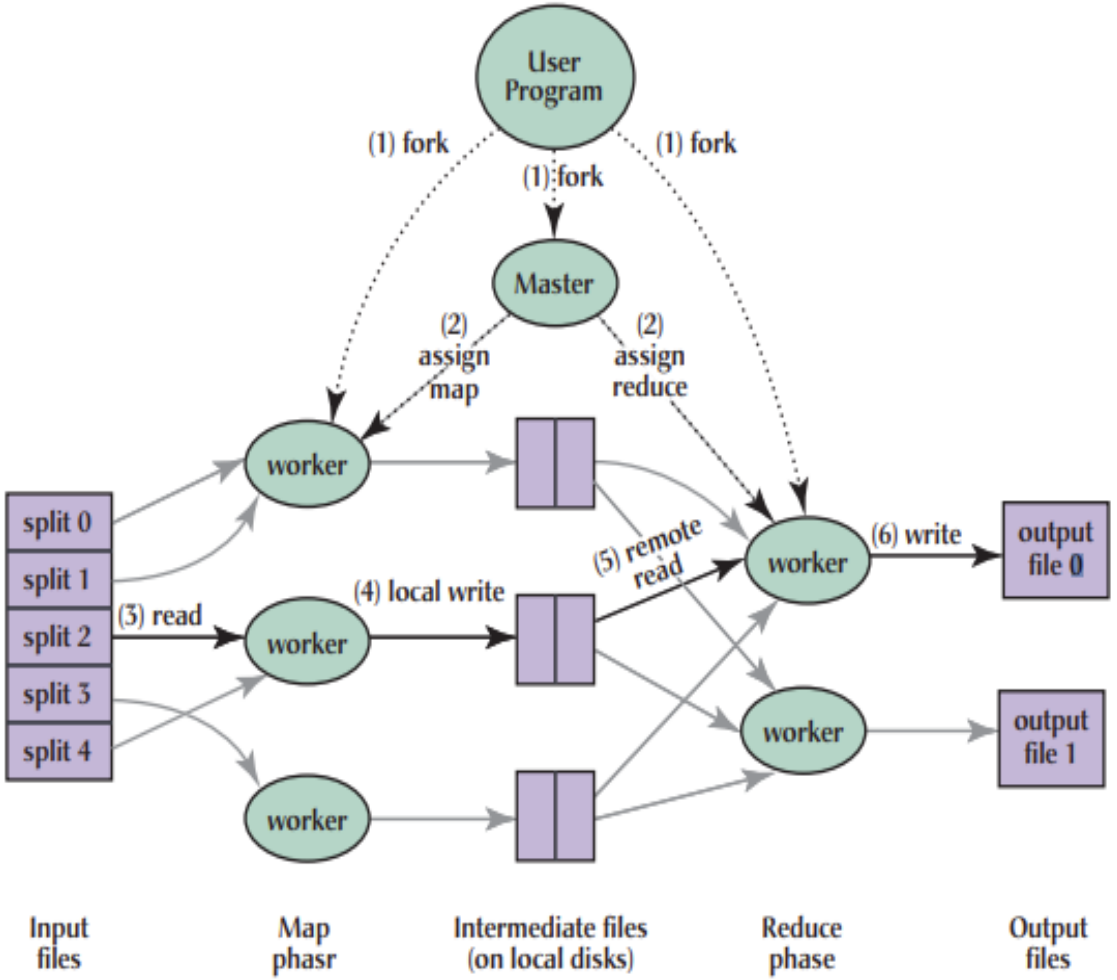
Dentro do HDFS, um **nó de nome** gerencia operações de namespace do sistema de arquivos do tipo abrir, fechar e renomear arquivos e diretórios. Um **nó de nome** também mapeia blocos de dados a nós de dados, os quais gerenciam as solicitações de leitura e gravação dos clientes HDFS. Os **nós de dados** também criam, excluem e replicam blocos de dados de acordo com as instruções do nó de nome dominante.

Ciências de Dados e Big Data

HDFS – MapReduce

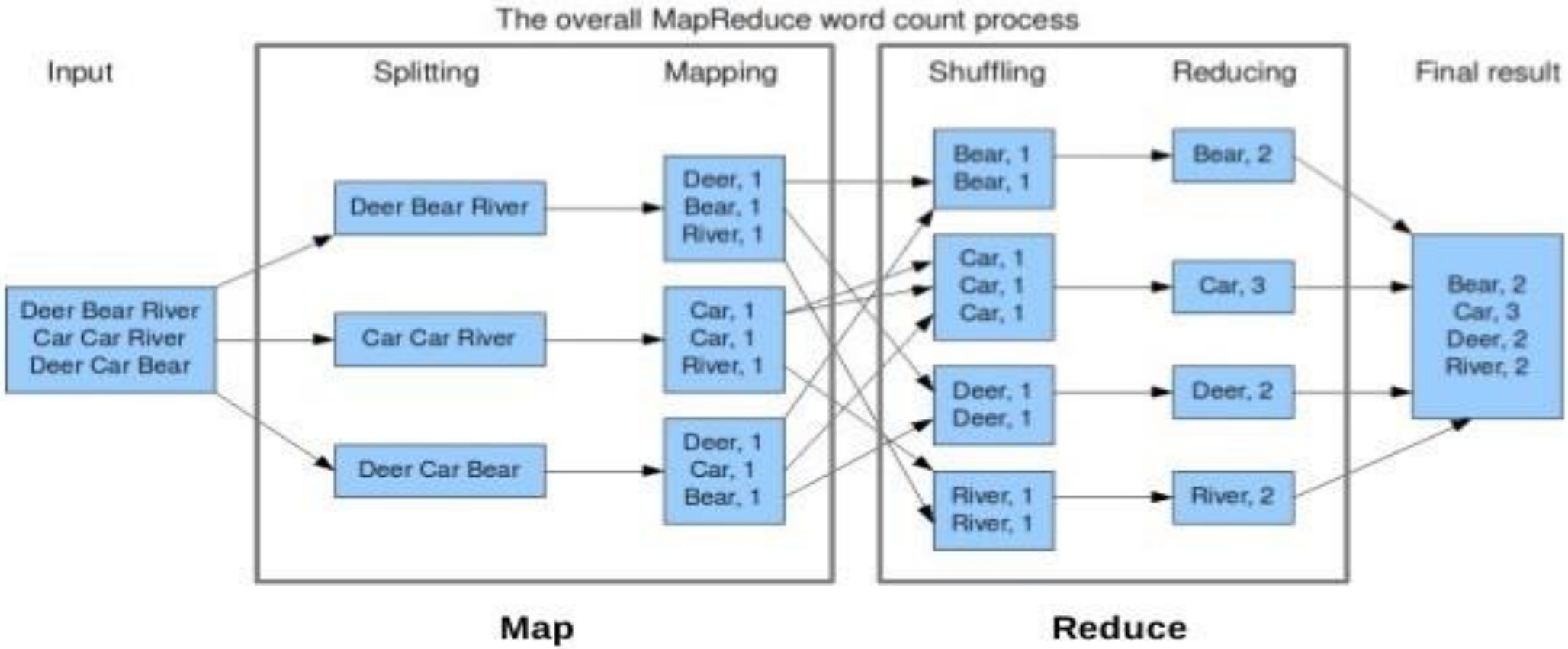
MapReduce é um modelo de programação desenhado para processar grandes volumes de dados em paralelo, dividindo o trabalho em um conjunto de tarefas independentes. Programas MapReduce são escritos em um determinado estilo influenciado por construções de programação funcionais, especificamente expressões idiomáticas para listas de processamento de dados. Este módulo explica a natureza do presente modelo de programação e como ela pode ser usada para escrever programas que são executados no ambiente Hadoop.

https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html



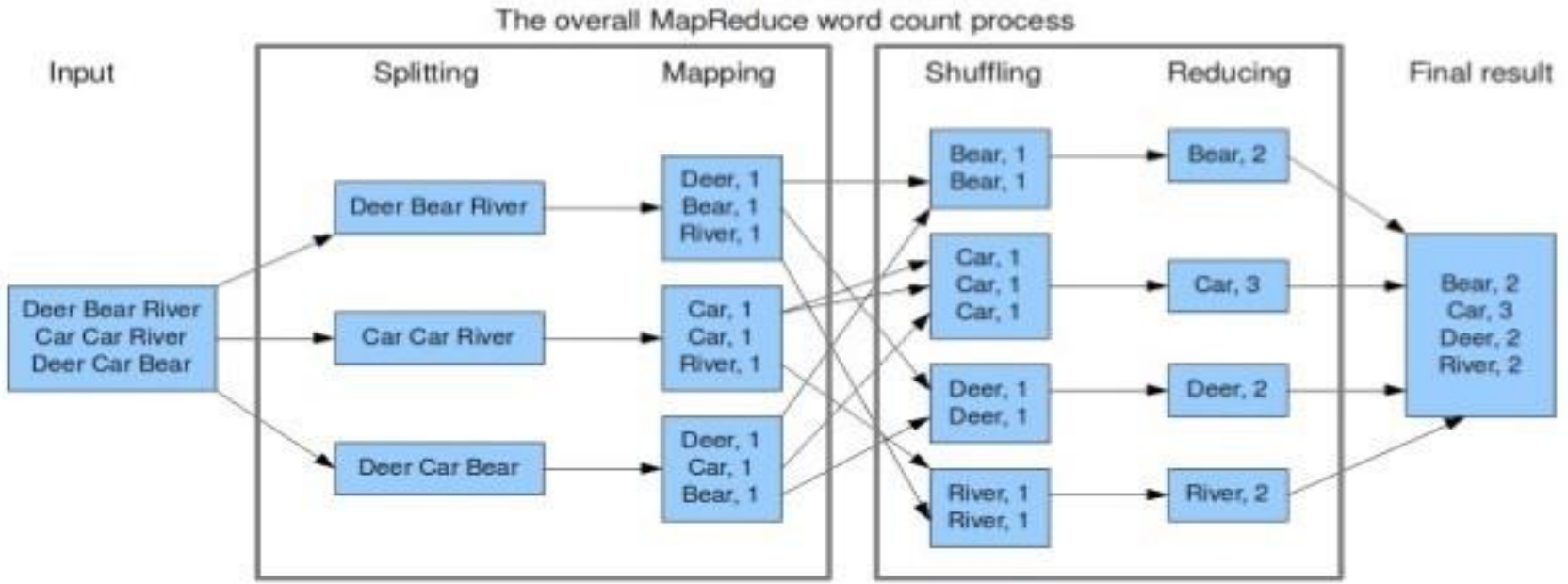
Ciências de Dados e Big Data

HDFS – MapReduce



Ciências de Dados e Big Data

HDFS – MapReduce



Map

```
map(String key, String value):
   // key: document name
   // value: document contents
   for each word w in value:
      EmitIntermediate(w, "1");
```

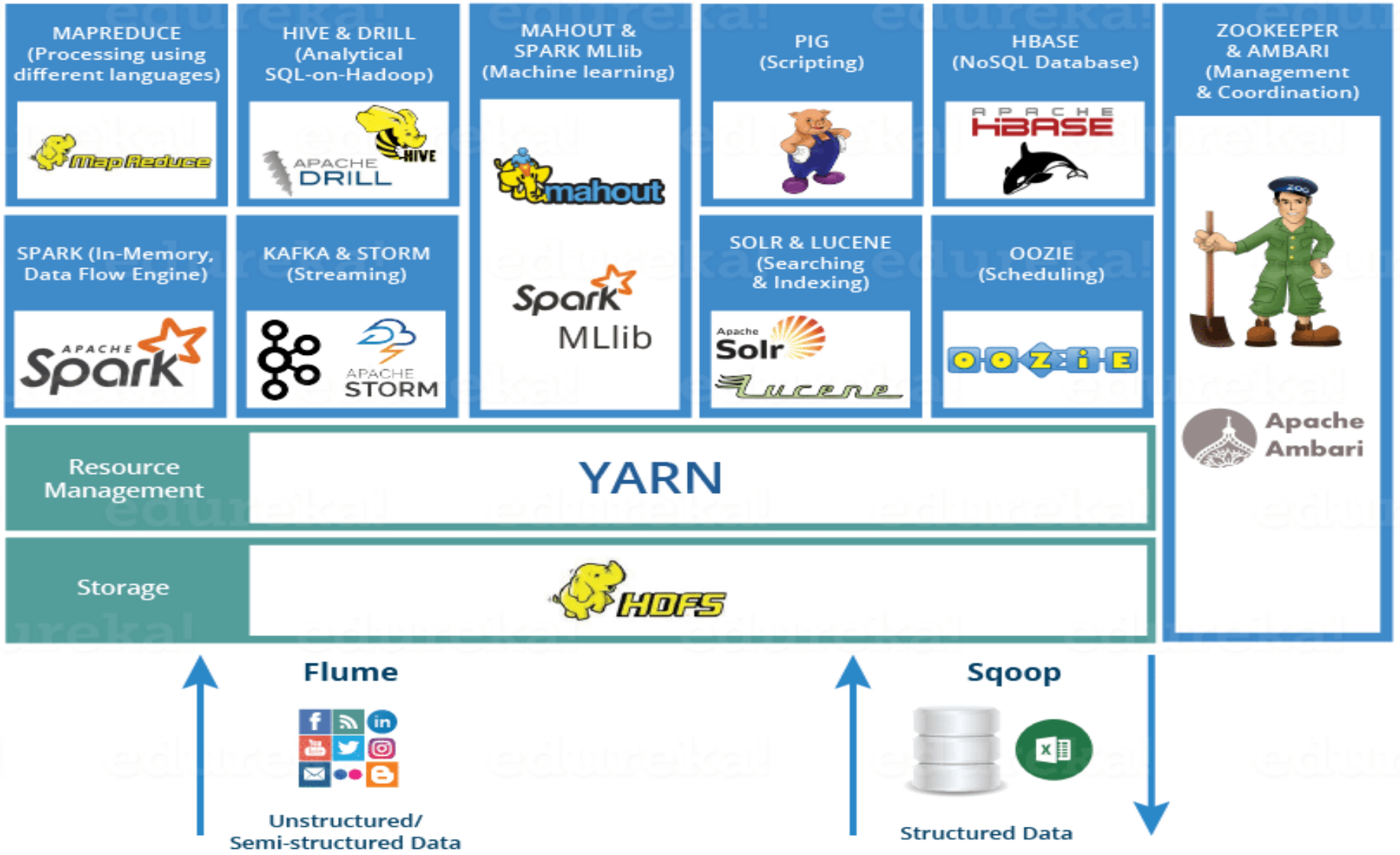
Reduce

```
reduce(String key, Iterator values):
   // key: a word
   // values: a list of counts
   int result = 0;
   for each v in values:
      result += ParseInt(v);
   Emit(AsString(result));
```

Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Ciências de Dados e Big Data

Ecossistema Hadoop



OBRIGADO!!!!