Hadoop 3.0 及下一步发展

Intel Big Data Technologies(BDT)

陈怡 sammi.chen@intel.com
龚奇源 qiyuan.gong@intel.com

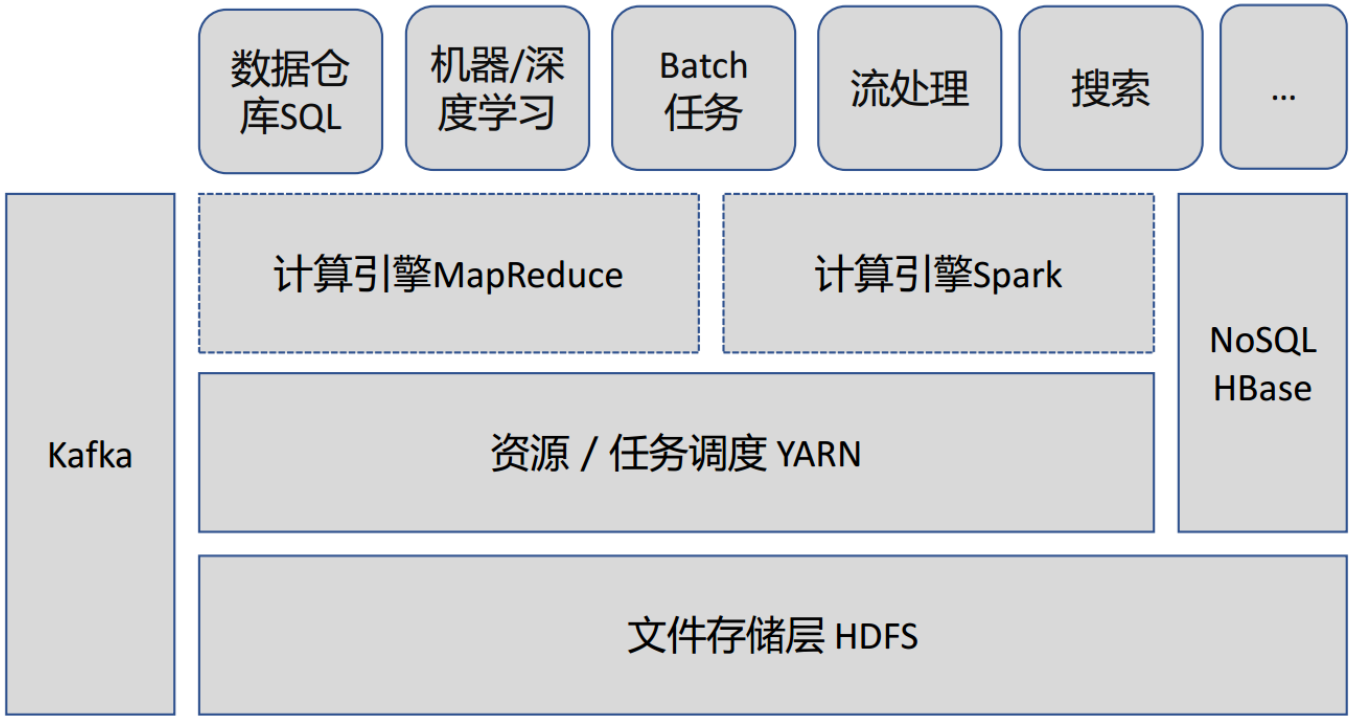
概要

- Hadoop 3.0 简介
- HDFS 纠删码
- HDFS 进一步发展
- Hadoop和深度学习

Apache Hadoop 3.0 发布时间表

- Alpha1 - 2016/09/03
- Alpha2 - 2017/01/25
- Alpha3 - 2017/05/15
- Beta - 2017/07(估计)
- GA - 2017/08(估计)

Hadoop 生态系统

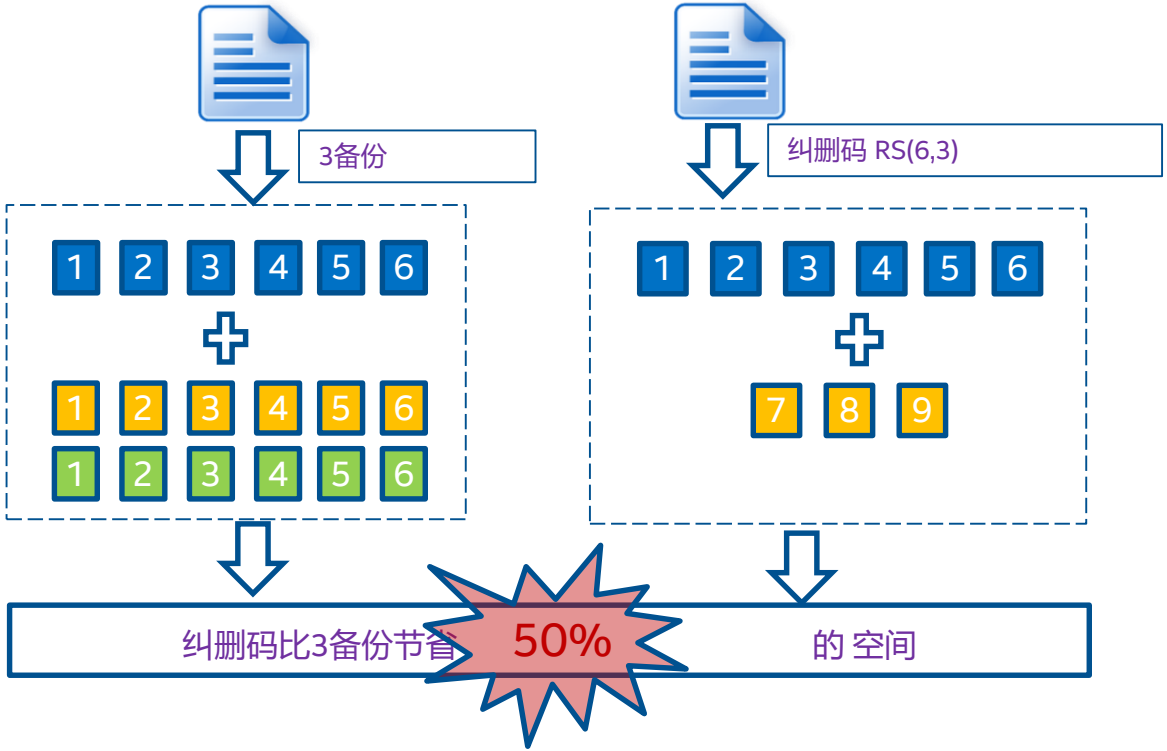

数据仓库SQL
机器/深度学习
Batch任务
流处理
搜索
...
Kafka
计算引擎MapReduce
计算引擎Spark
NoSQL HBase
资源 / 任务调度 YARN
文件存储层 HDFS

Hadoop 3.0 新功能介绍

- Common
  - JDK 8+ 升级
  - Classpath 隔离
  - Shell 脚本的重构
  - Hadoop服务默认端口变化
- HDFS
  - 纠删码
  - 多个Standby Namenode
  - Datanode内部balancer工具
  - 云存储的支持
- YARN
  - 新Timeline Service
- MapReduce
  - Task层次的Native优化

HDFS纠删码

纠删码举例 - RS(6,3)


3备份
纠删码 RS(6,3)
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
7 8 9
纠删码比3备份节省
50%
的 空间

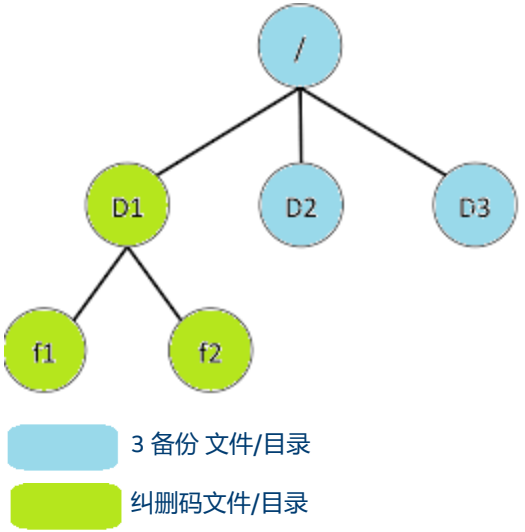
数据可靠性和存储效率

- 数据可靠性：可以同时容忍最多几个节点故障
- 存储效率：数据块/(数据块+校验块)

	可靠性 (越高越好)	存储效率 (越高越好)
1-replica	0	100%
3-replica	2	33%
EC RS(6,3)	3	67%
EC RS(10, 4)	4	71%
EC RS(3,2)	2	60%

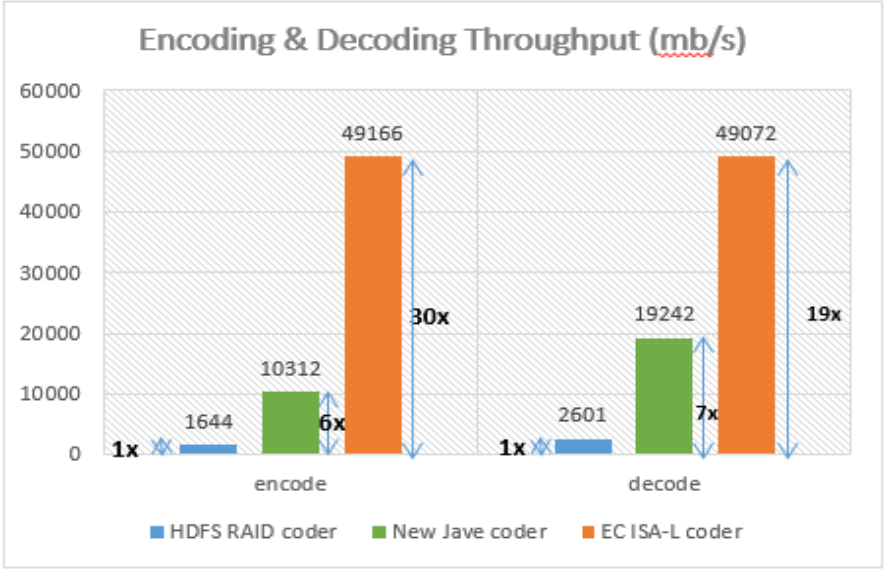
选择多样性

- 3备份数据和纠删码数据共存
- 数据互换的系统工具
- 系统内置多个纠删码策略
- 用户可以定制自己的纠删码策略



纠删码Coder性能

- 基本Intel ISA-L 库实现了高性能的纠删码 ISA-L Coder (https://github.com/01org/isa-l)
- Intel ISA-L 库利用 CPU 的高级指令集 (SSE, SSE2, AVX, AVX2, AVX512 etc.) 来优化纠删码计算
- ISA-L coder 比之前的HDFS RAID coder 和 New Java coder 性能有很大的提升



端到端性能

(3 备份作为基准，以执行时间来衡量，执行时间越短越好)

HDFS进一步发展

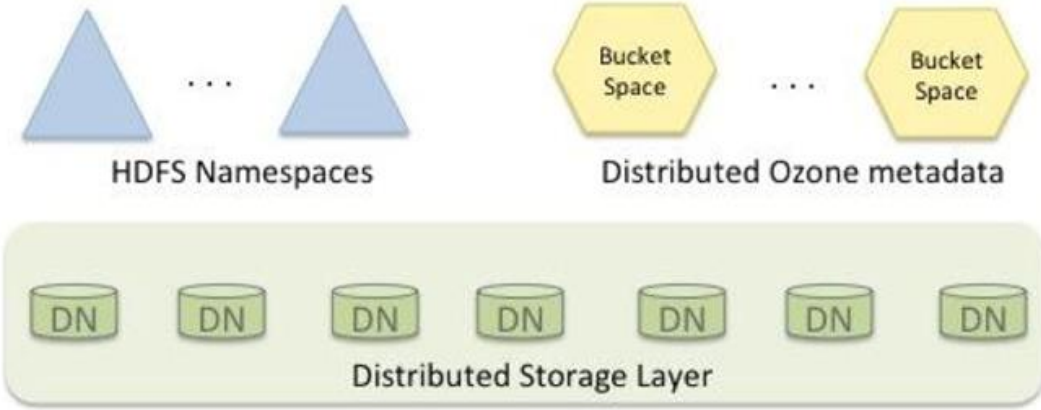
存储在云端

统一的Hadoop文件系统和API

Hadoop兼容文件系统抽象层:统一的存储API接口 `hadoop fs -ls s3a://job/`

对象存储：Ozone(Object Store inside HDFS)

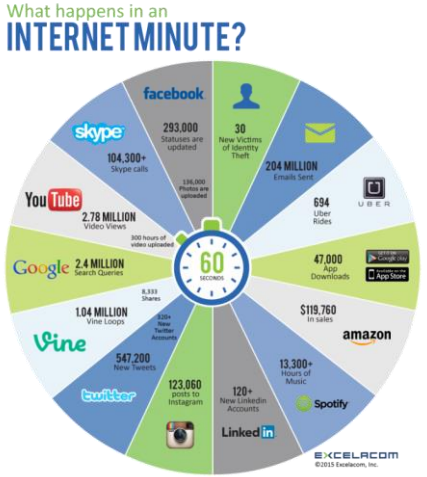
- 对象存储是一个流行的趋势：S3，Azure，Aliyun，.....
- Hadoop正在演化成为一个更为通用的平台，支持对象存储能使Hadoop适用于更多的应用领域
- 对象（Object）更为轻量，没有file metadata 和 ACL，基于K/V的API 在一些场景下更为友好
- 目标：
  - ❑ 支持T级别的数据对象
  - ❑ 支持任意大小的对象，从几K到几十MB
  - ❑ 保证一致性、可靠性和可用性



智能存储管理（SSM）

数据趋势

要存储和处理的数据量越来越庞大

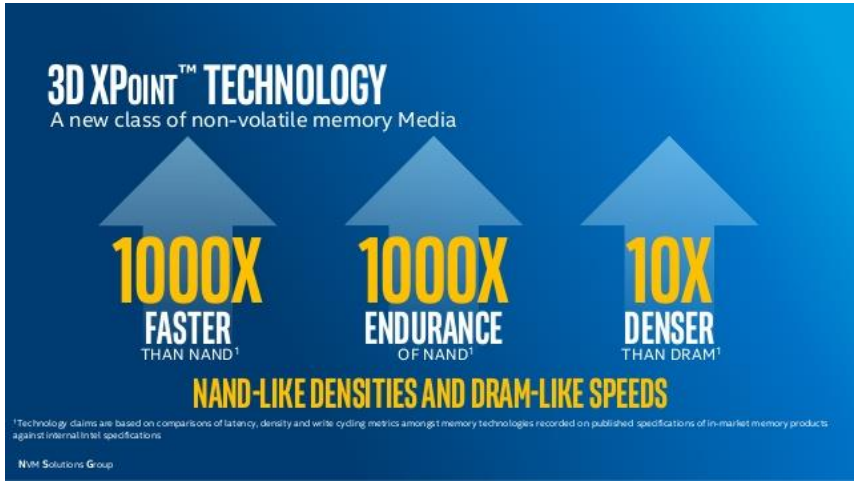


- 物联网的发展使得接入设备越来越多
- 实时流处理技术的发展使数据导入速度越来越快
- 数据分析（OLAP）日趋成熟
- 人工智能（AI）新时代，人们希望聚集更多的数据进行深度学习

硬件趋势

存储设备的两极：越来越廉价和越来越快

- 要么更廉价，更多更老的数据促进更廉价的设备
- 要么更快，SSD步伐越来越快
- 3XD Point技术和NVM设备，存储和内存统一起来



存储管理挑战

- 一个集群，同时支持好：
  - 大文件、小文件
  - 热数据、冷数据
  - 在线处理、离线分析
- 如何及时感知数据的温度
  - 将经常读的数据转入到SSD
  - 将变冷的数据移入到廉价设备
- 如何评估和提高存储设备的存储和读取效率

智能存储管理功能 目标

- 端到端的全面的智能存储解决方案
- 完整地收集集群的存储和数据访问统计
- 简化地、智能地感知集群存储状态变化并作出存储策略调整
- 提升整个存储系统的效率和利用率
- HDFS-7343：正在开发当中，欢迎反馈和参与！

实现原理

为了实现目标，我们需要:

- **收集历史信息**

  数据存取信息, HDFS 数据统计信息

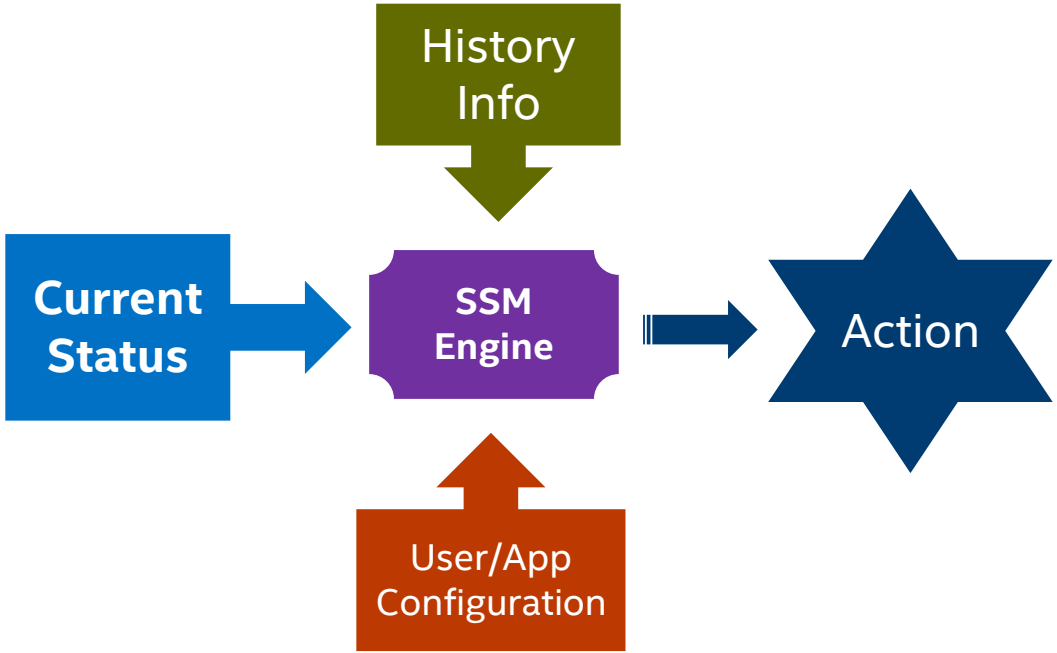
- **感知当前状态**

  存储资源状态

  数据状态

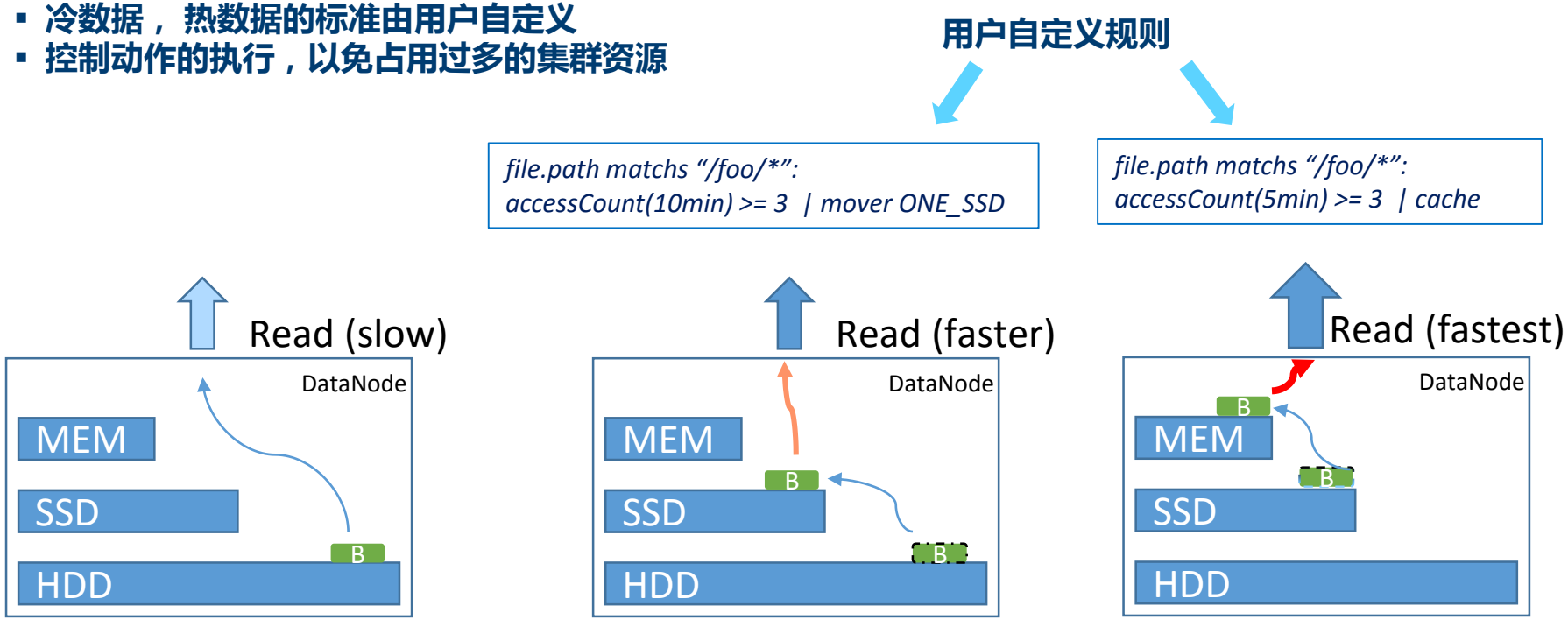
- **给用户权利**

  用户自定义规则，来表达偏好

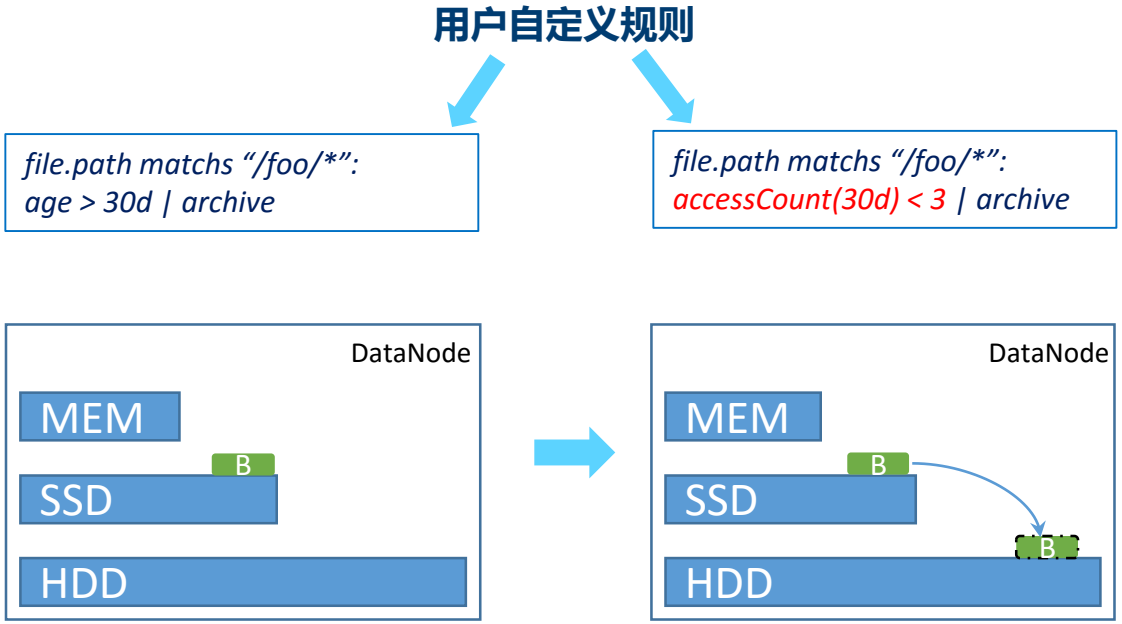


典型应用场景 – 将热数据迁移到快速设备

- 冷数据， 热数据的标准由用户自定义
- 控制动作的执行，以免占用过多的集群资源



典型应用场景 – 将冷数据迁移到慢速设备，释放快速设备空间给热数据



Hadoop和深度学习

概要

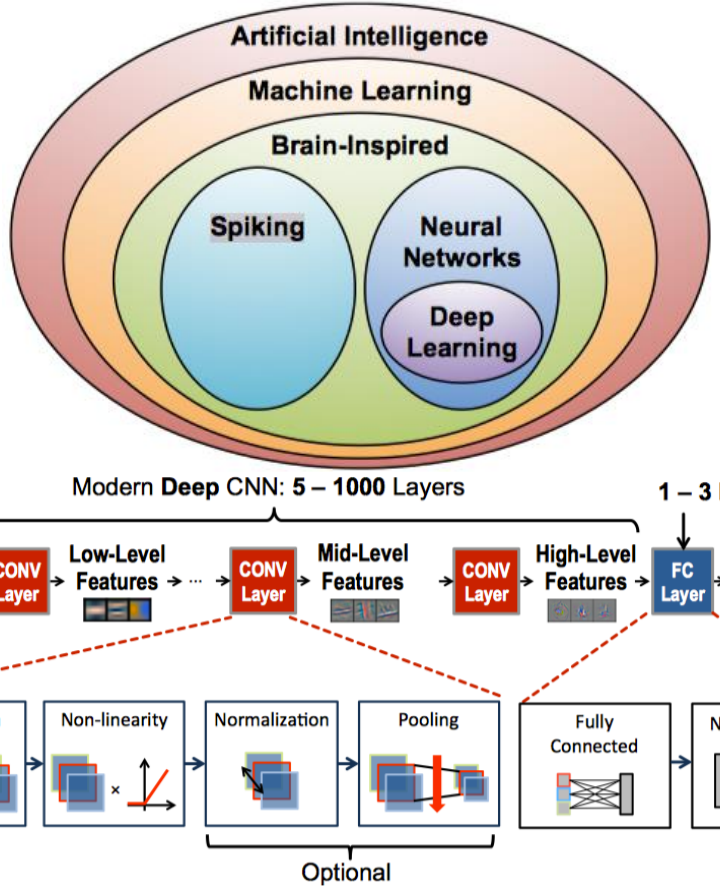
1. 深度学习发展趋势
2. 与Hadoop相关的问题、挑战和机遇
3. 更完备的资源管理(YARN)
4. 更强大的分布式存储(HDFS)
5. 依赖和环境问题(经验分享)

深度学习发展趋势(1)

“老”技术的“第三波浪潮”

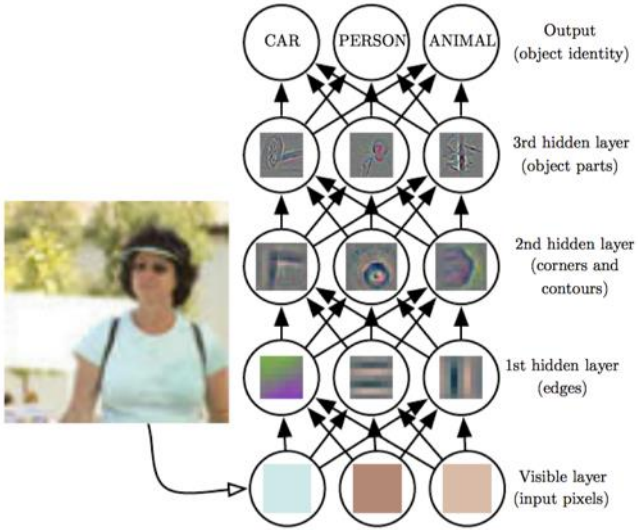
DNN Timeline

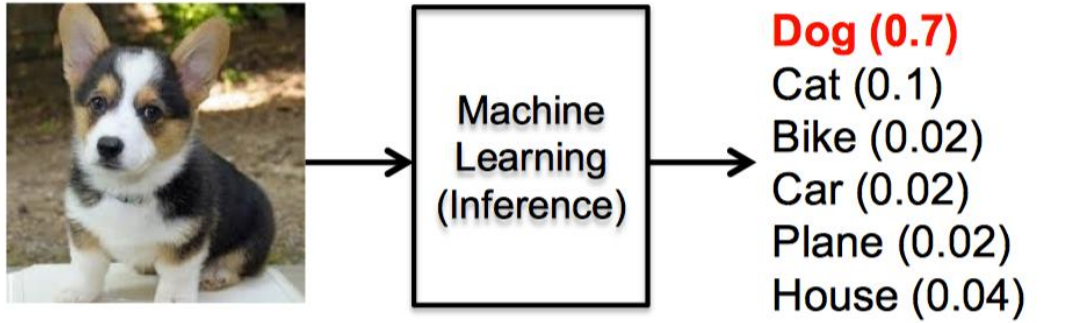
- 1940s - Neural networks were proposed
- 1960s - Deep neural networks were proposed
- 1989 - Neural net for recognizing digits (LeNet)
- 1990s - Hardware for shallow neural nets (Intel ETANN)
- 2011 - Breakthrough DNN-based speech recognition (Microsoft)
- 2012 - DNNs for vision start supplanting hand-crafted approaches (AlexNet)
- 2014+ - Rise of DNN accelerator research (Neuflow, DianNao...)

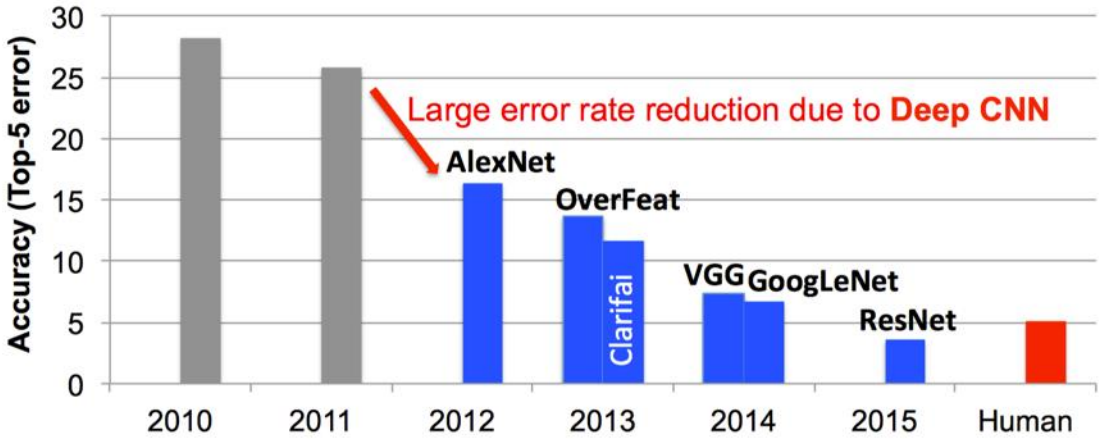


深度学习发展趋势(2)

图像分类：极高的准确率

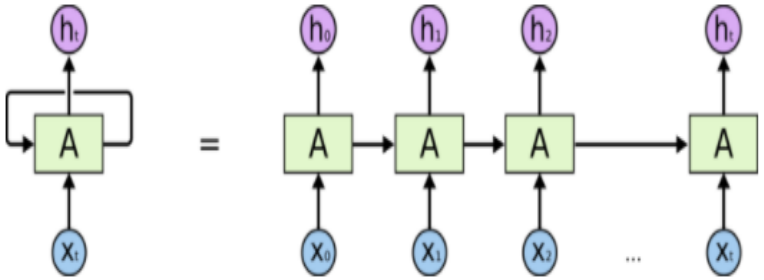




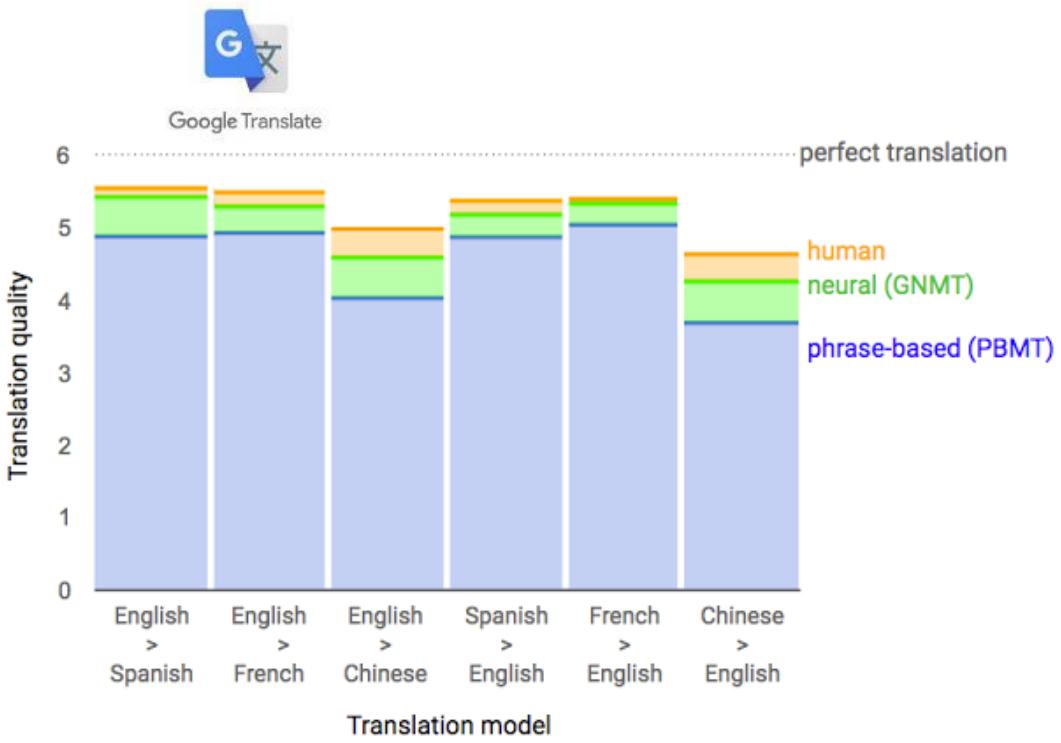


深度学习发展趋势(3)

自然语言处理：极高的翻译质量



An unrolled recurrent neural network.



深度学习发展趋势(4)

深度学习框架“**大混战**”

- **更易用的API**
- **更好的加速器支持**
- **更多的模型支持**
- 更快的算法支持
- ....

Caffe

Microsoft CNTK

torch

K

TensorFlow

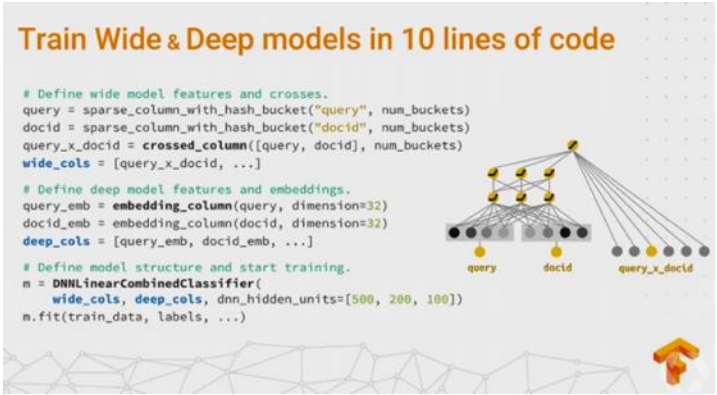
dmlc mxnet

theano

DL4J DEEPLEARNING4J

Lasagne

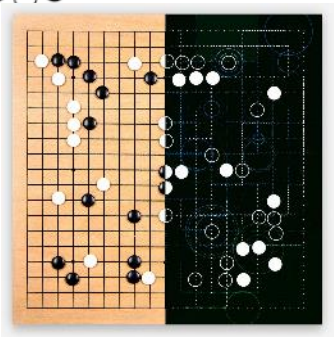
dmlc XGBoost

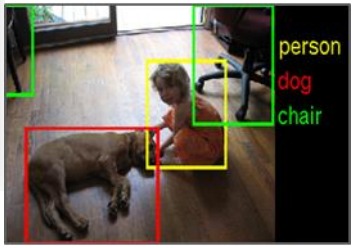


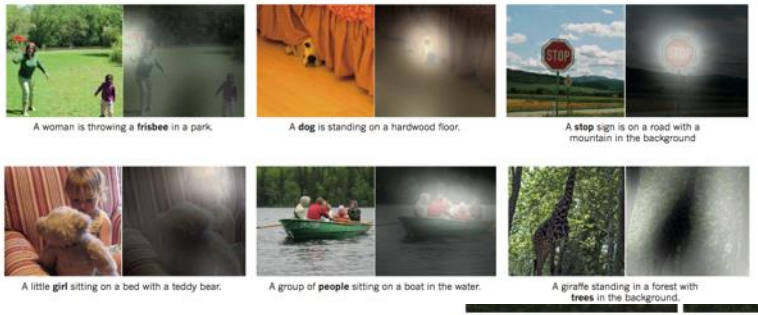
深度学习发展趋势(5)

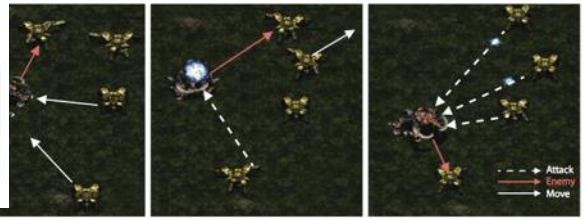
AlphaGo

- 深度学习应用“大爆炸”
  - 下围棋
  - 打游戏
  - 学毕加索作画
  - 写诗
  - 写代码
  - 检测皮肤癌
  - ....









(a) time step 1 (b) time step 2 (c) time step 3 (d) time step 4

深度学习发展趋势(6)

Intel BigDL

TensorFlow/Caffe on Spark

TensorFlow on K8S

HDL

TensorFlow on Yarn

TensorFlow on Yarn

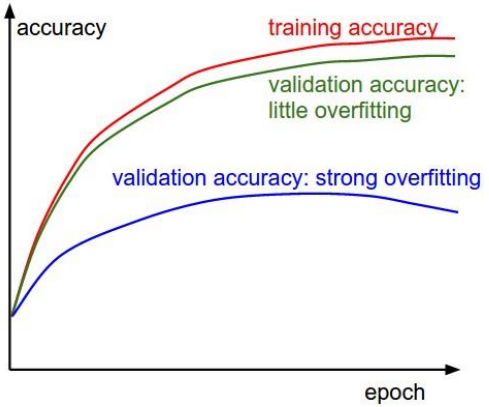
TensorFlow on Yarn

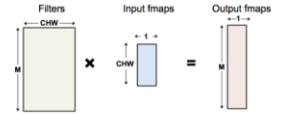
TensorFlow on Mesos

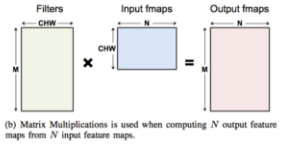
深度学习发展趋势(7)

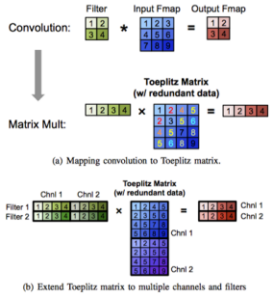
存在问题：

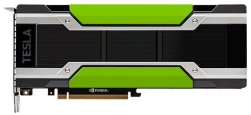
- 更完备的数据集
- 过拟合
- 参数调整
- 需要大量计算资源
- 需要大量存储资源
- …

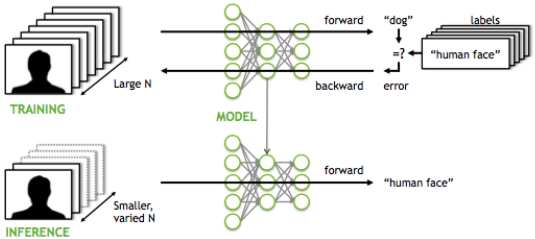










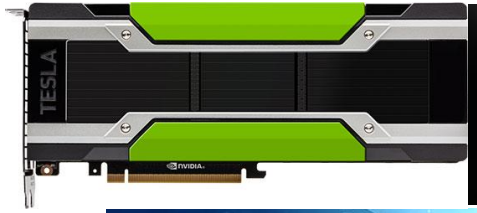


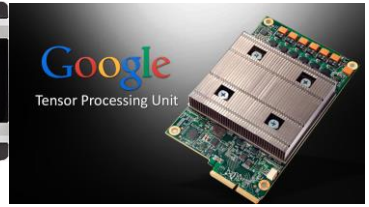
与Hadoop相关的问题、挑战和机遇

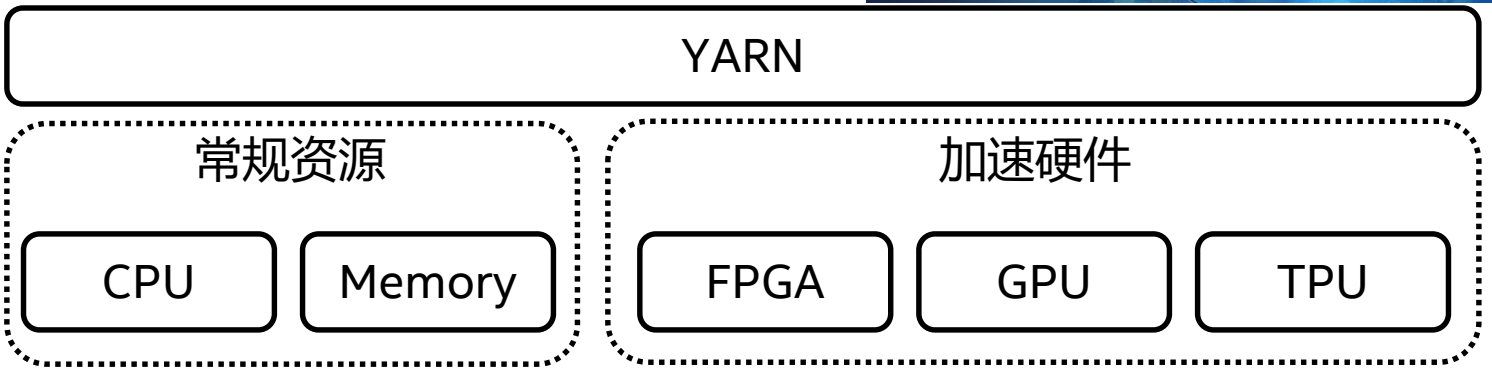
1. 资源管理(YARN)
   - 深度学习加速器管理(GPU、TPU和FPGA)
   - Long Running和自动调整 (Inference Serving)
2. 分布式存储(HDFS)
   - 快速/低成本的顺序读取 (原始文件和模型存储)

更完备的资源管理(YARN)(1)

- 深度学习加速器管理(GPU、TPU和FPGA)
  - GPU on YARN**(YARN-6223)**
  - FPGA on YARN**(YARN-5983)**
  - Customer Defined Resource**(YARN-3926)**







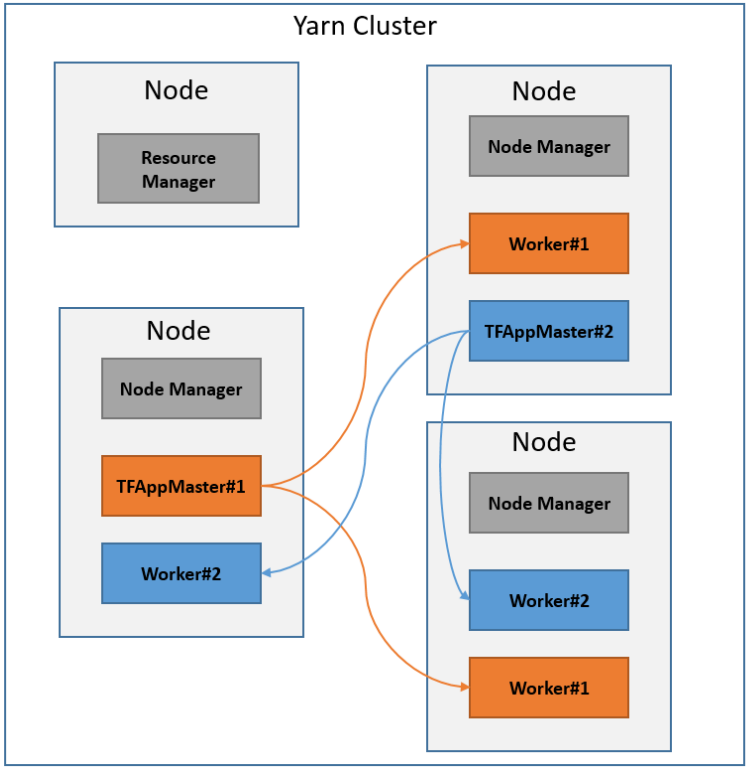
更完备的资源管理(YARN) (2)

Task 1: train googlenet with tensorflow on 1 node with 4 GPUs

TFClient#1

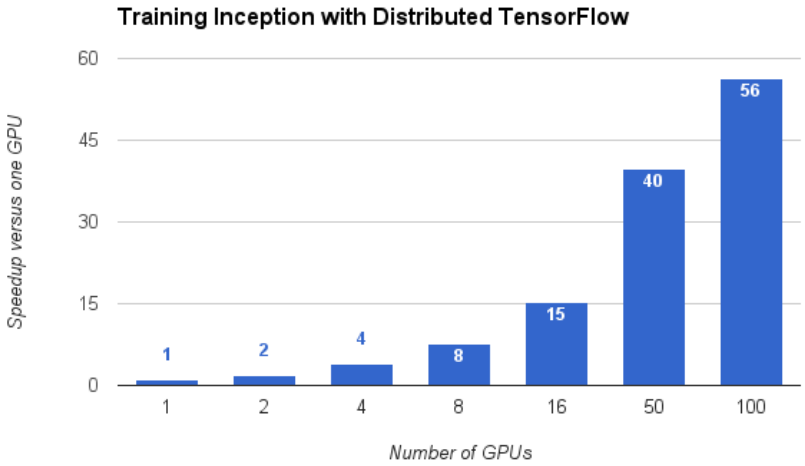
TFClient#2

Task 2: train XXX model with tensorflow/mxnet on 8 nodes with FPGA



更完备的资源管理(YARN) (3)

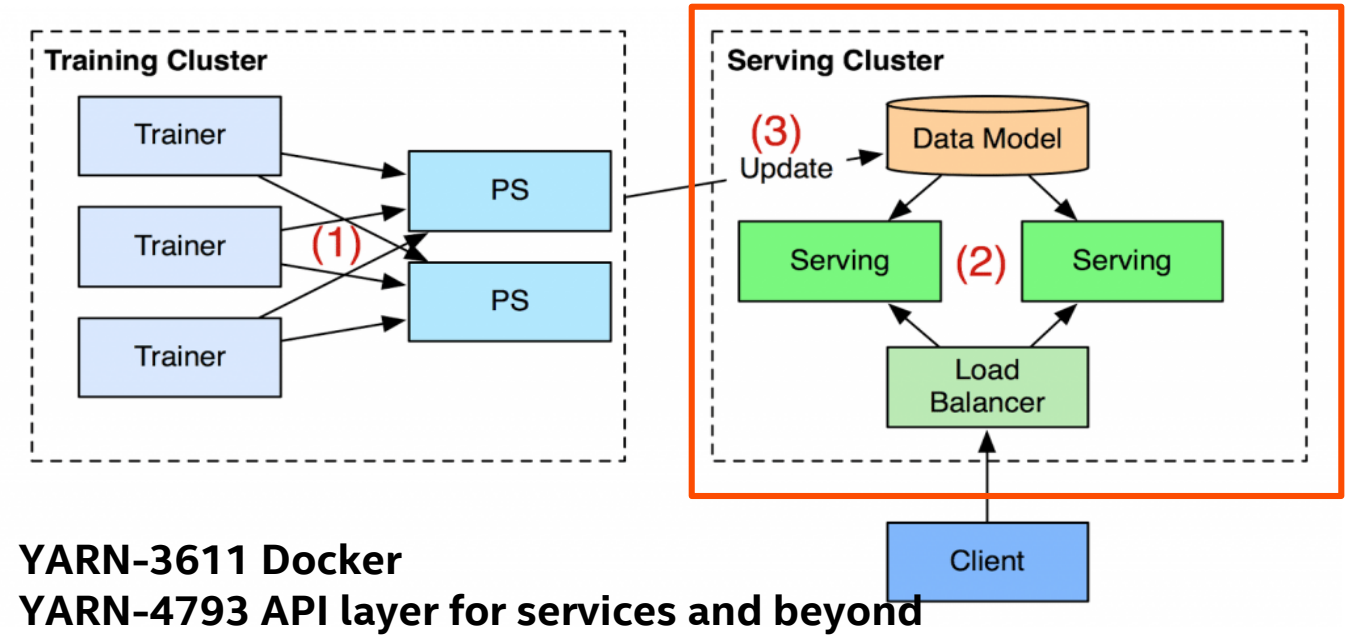
- Distributed DL (Data/Model Parallelism)
  - **Speed Up Training**
    - 1 month ->1 day ->Even faster
    - **Real time** application/analysis
    - **Deeper** Neural Network
  - **Large scale DL Applications**
    - Large datasets
    - Large parameters: graphs, variables and weights etc.



Natural Language Processing
Train with **10 billion parameters**
1day/64 k40s Google

更完备的资源管理(YARN)(4)

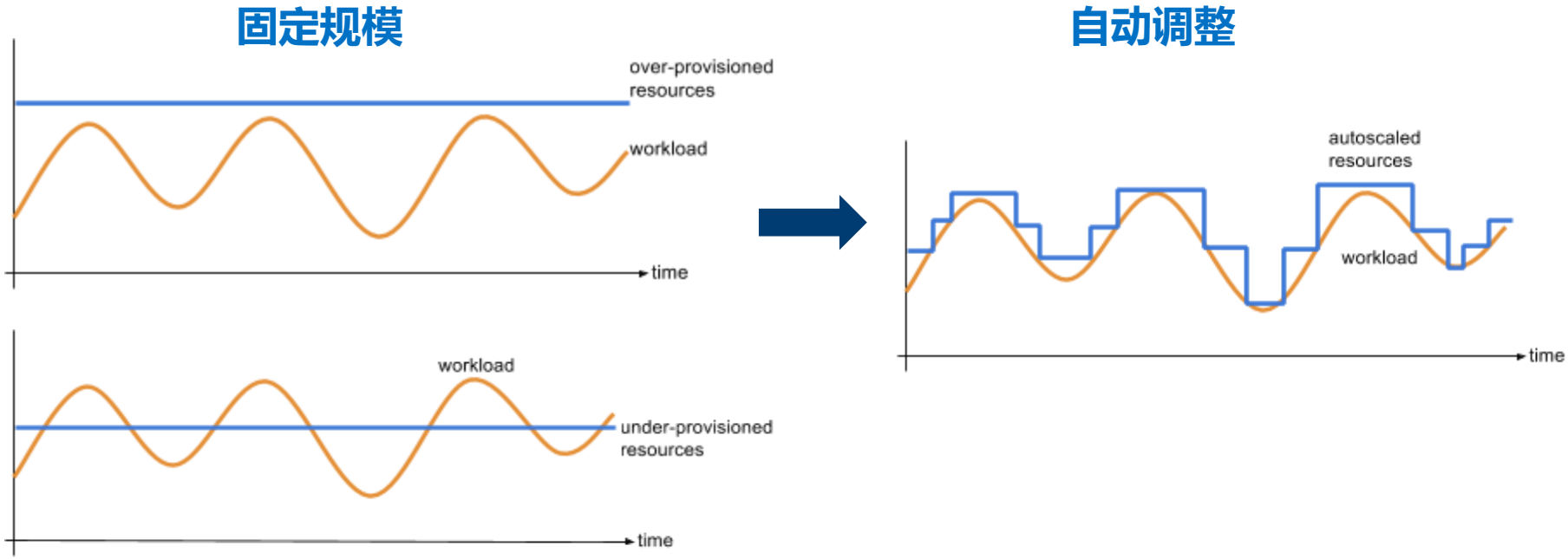
- Long Running和自动调整 (Inference Serving)



YARN-3611 Docker
YARN-4793 API layer for services and beyond

更完备的资源管理(YARN)(5)

更高的可用性和资源利用率



更强大的分布式存储(HDFS)

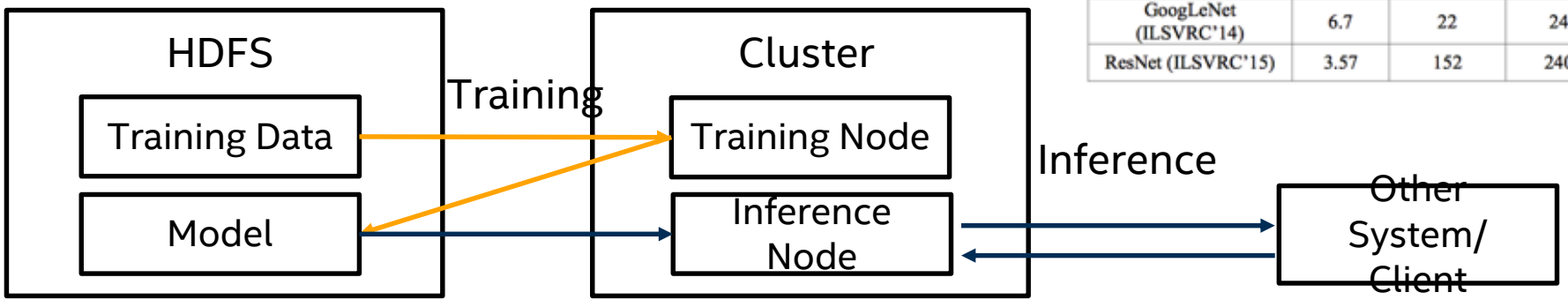
MNIST

ImageNet

快速/低成本的顺序读写需求

- 数据集的存储和读取
- 模型、日志和片段的存储与读取

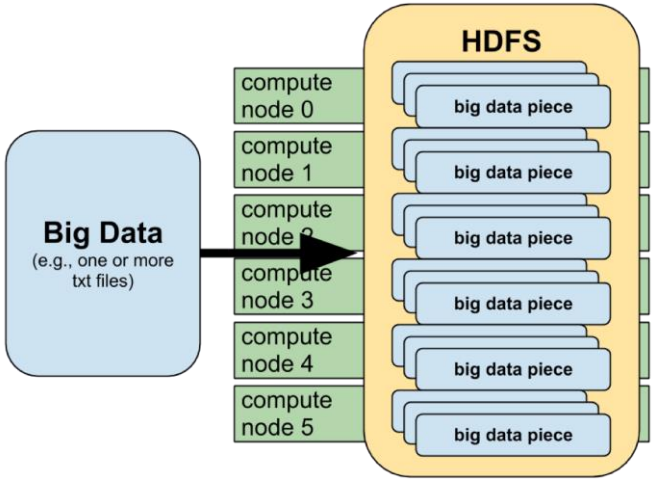
Network	Top-5 Error %	Depth (#layers)	Model Size (MB)
AlexNet (ILSVRC'12)	15.3	8	240
VGG (ILSVRC'14)	7.3	19	500
GoogLeNet (ILSVRC'14)	6.7	22	24
ResNet (ILSVRC'15)	3.57	152	240

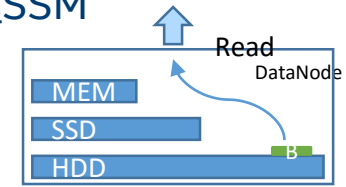


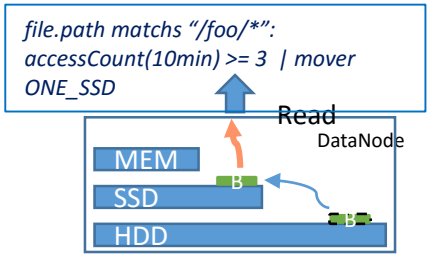
更强大的分布式存储(HDFS)

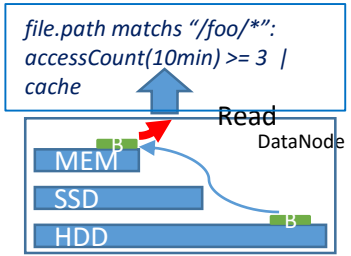
快速的顺序读写方案

- 硬件升级(成本高)
  - 节点增加(增加并行)
  - 介质升级(提高IO等)
- **软件和架构升级***
  - HDFS Cache和优化
  - 智能(混合式)存储管理SSM



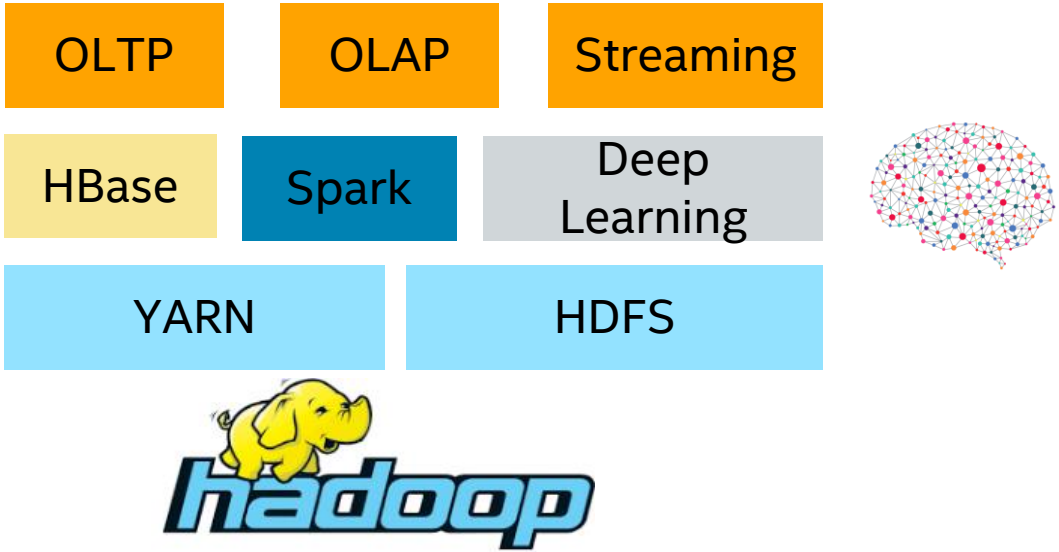






Hadoop的机遇

Deep Learning & Hadoop

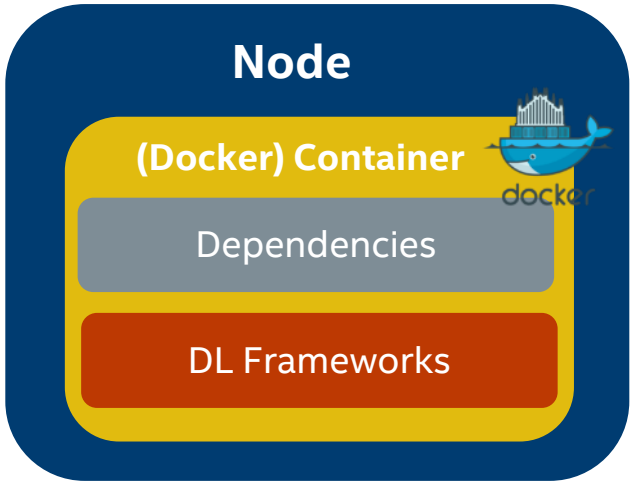


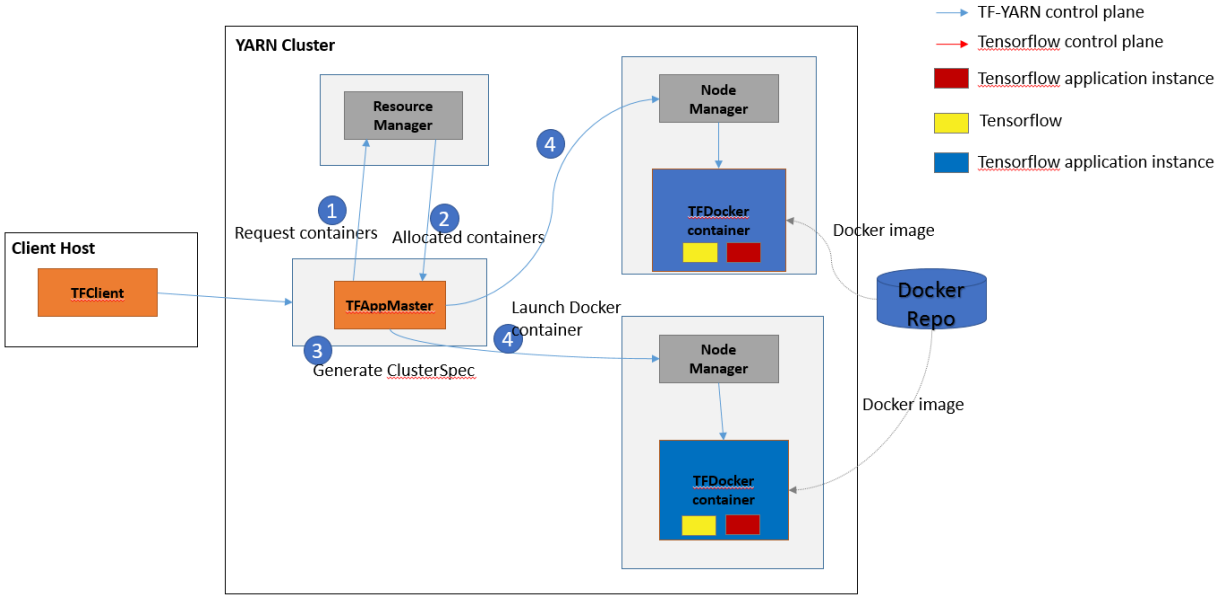
依赖和环境问题(经验分享)

- 多个深度学习框架的依赖问题
  - 第三方包类
    - 软件加速库：Openblas、MKL、CUDA和cuDNN等
    - 图像处理库：OpenCV
    - ....
  - **Python第三方包类**
    - **Numpy，Six, matplotlib和Jinja等**

依赖和环境问题(经验分享)(1)

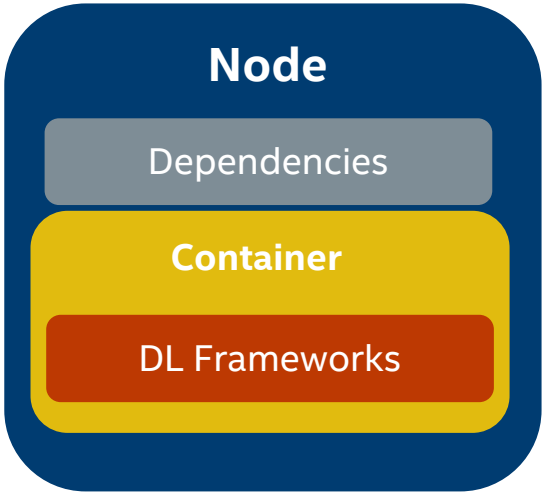
Docker方案(YARN-3611)

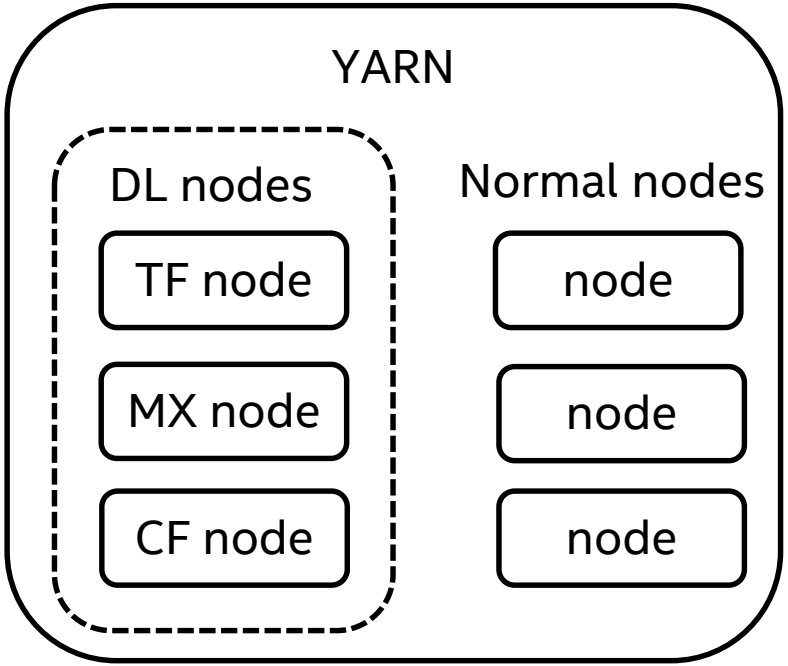




依赖和环境问题(经验分享)(2)

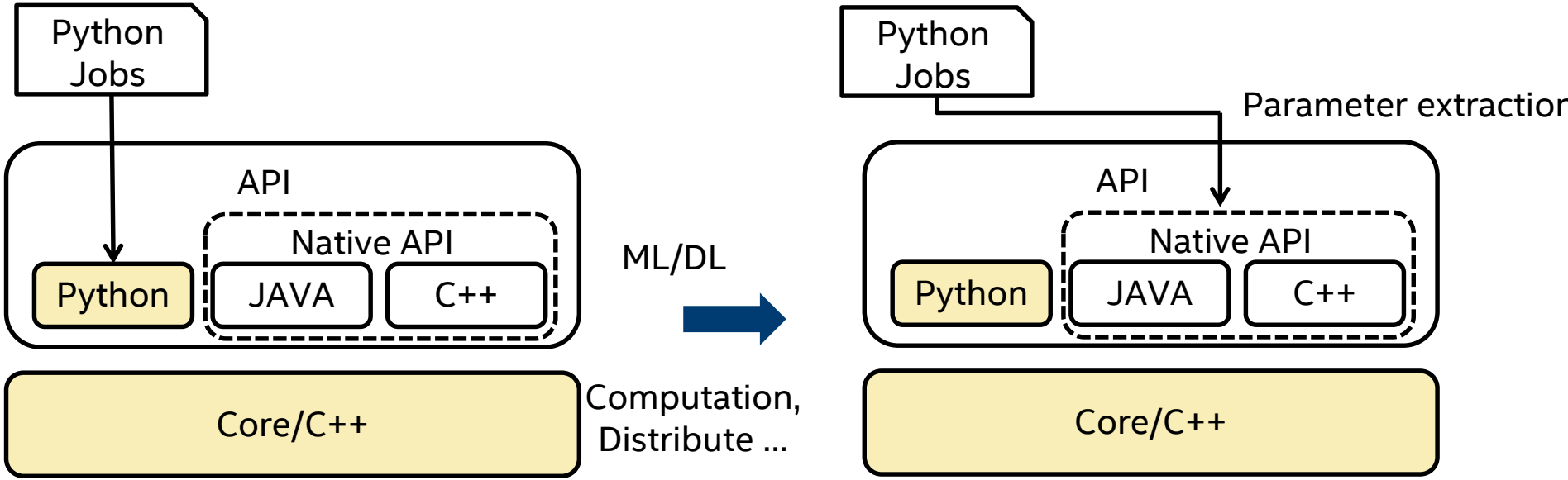
Label深度学习节点





依赖和环境问题(经验分享)(3)

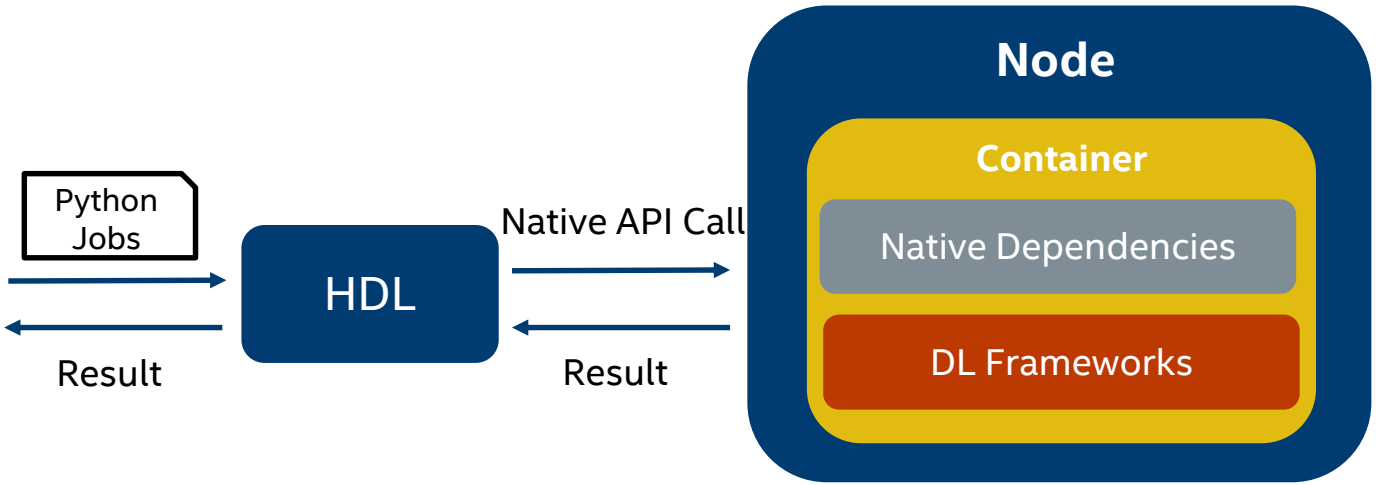
Native方案(我们的经验)



依赖和环境问题(经验分享)(5)

Native 方案的优势

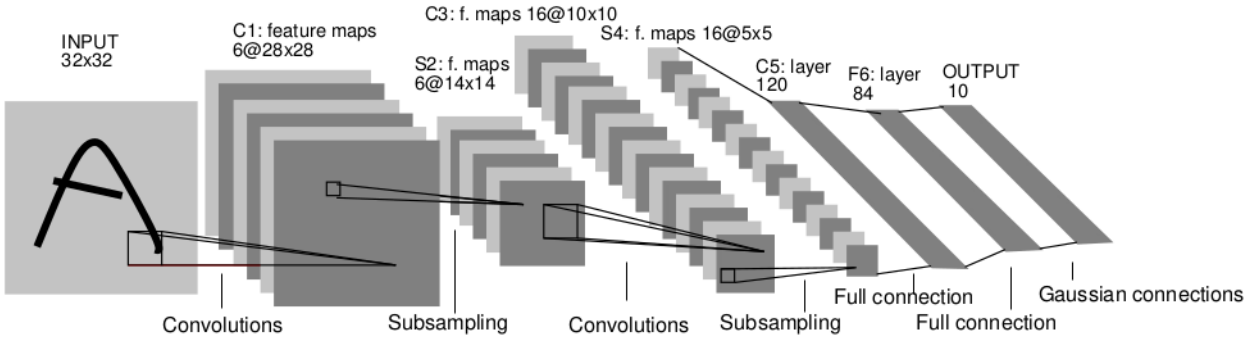
- 对现有Hadoop环境无影响
- 对用户无影响
- 性能更好



依赖和环境问题(经验分享)(6)

Native方案Demo

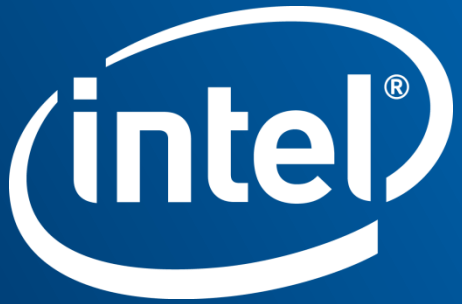
- 集群无Python环境
- 可以同时运行Caffe、 TensorFlow和MXNet (mnist) job



依赖和环境问题(经验分享)(7)

Thanks!
Q & A

Any suggestions or participations will be appreciated.


intel®