PaddlePaddle on Kubernetes

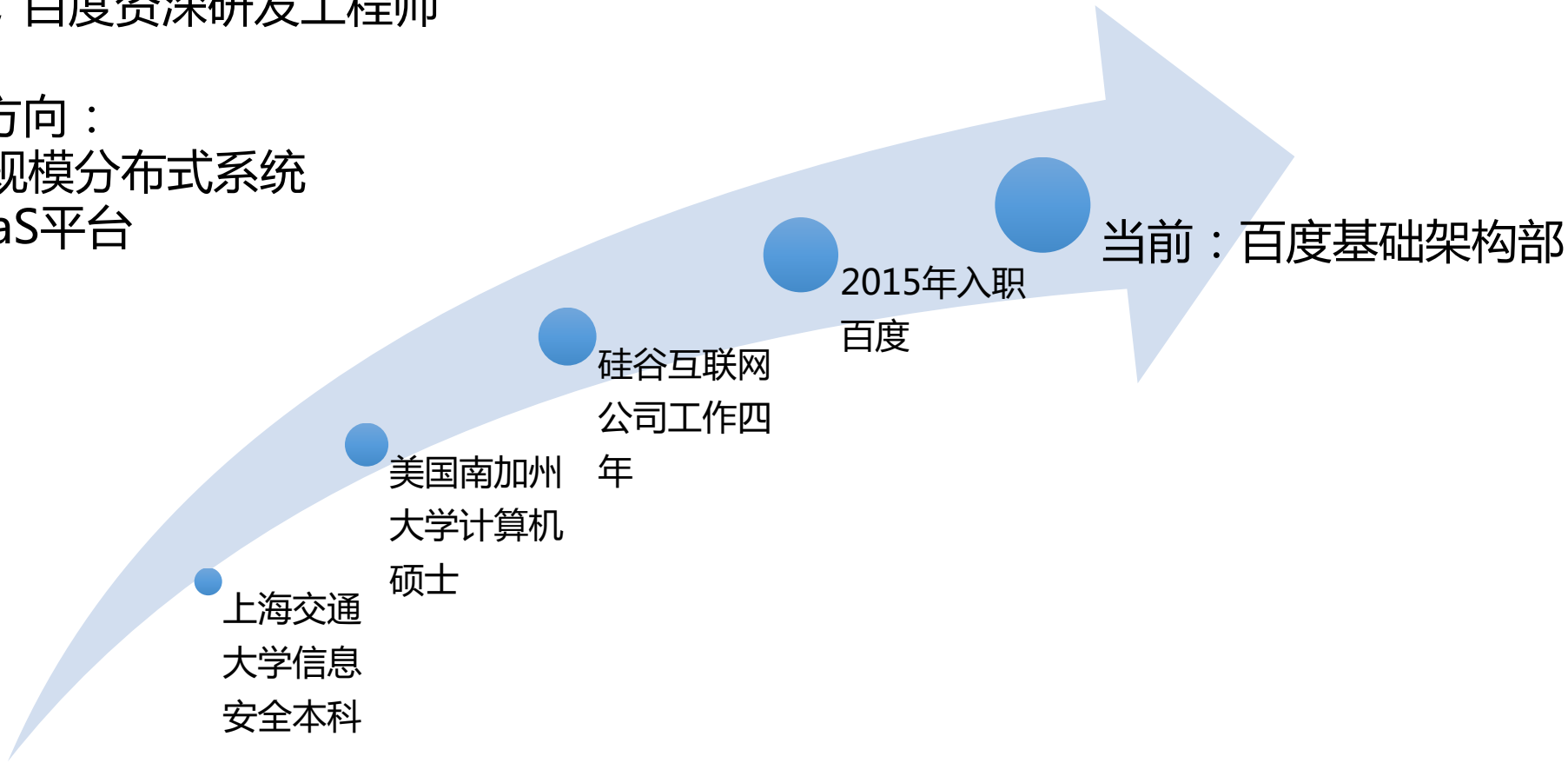
百度深度学习框架k8s实践

百度开放云　周倜

个人简介

周倜，百度资深研发工程师

主要方向：
- 大规模分布式系统
- PaaS平台



演讲内容

Why Kubernetes & AI

PaddlePaddle技术细节

- 分布式
- 在离线混布
- GPU支持

PaddlePaddle介绍

深度学习未来展望

百度云

Why Kubernetes & AI

Why Kubernetes – AI集群的痛点

AI的发展得益于(移动)互联网产生的大量的数据

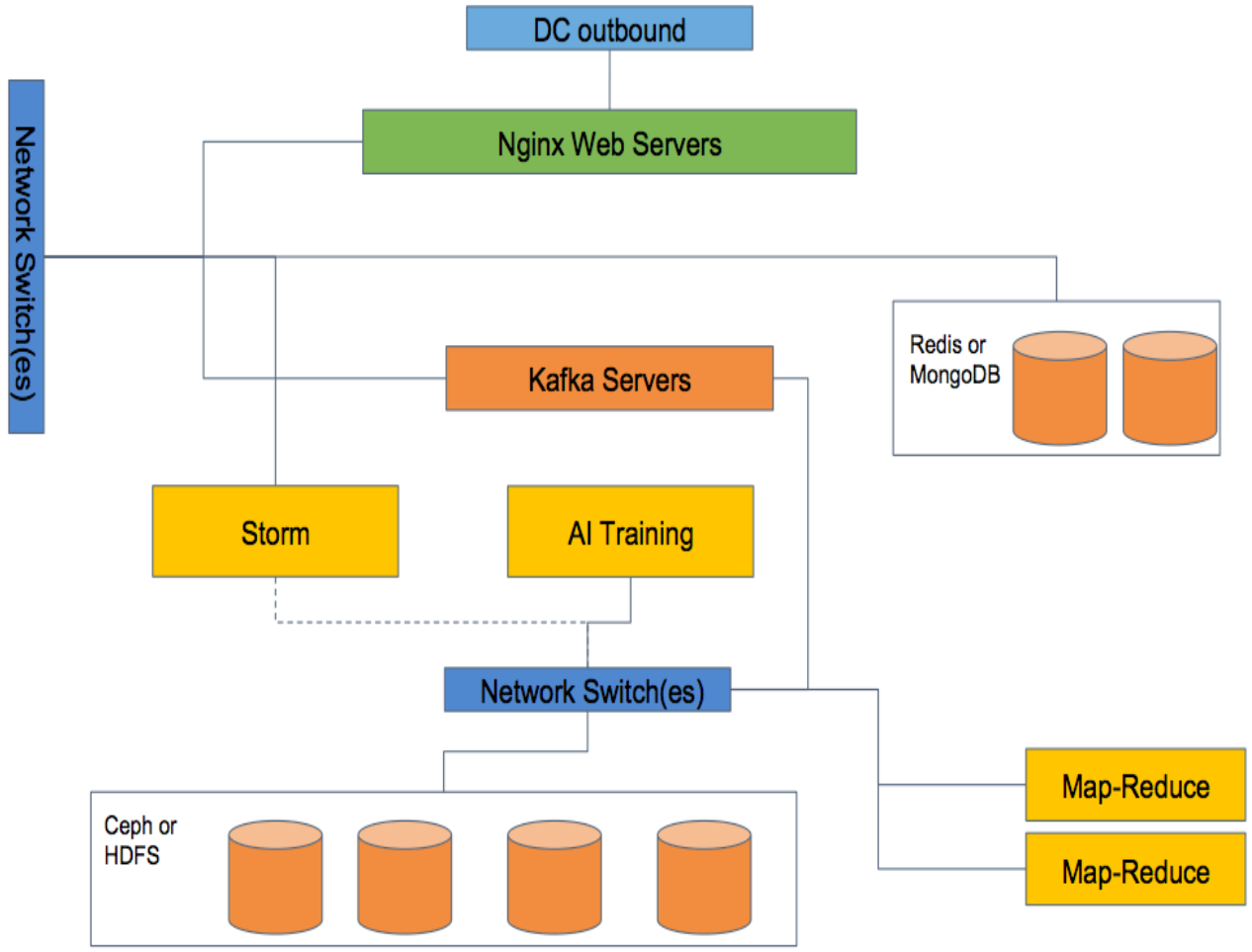
- 大量互联网开放的数据
- 大量Web Server的日志
- 大量在线传感器采集的数据

大量数据的处理，依赖于分布式存储和不同的分布式计算框架

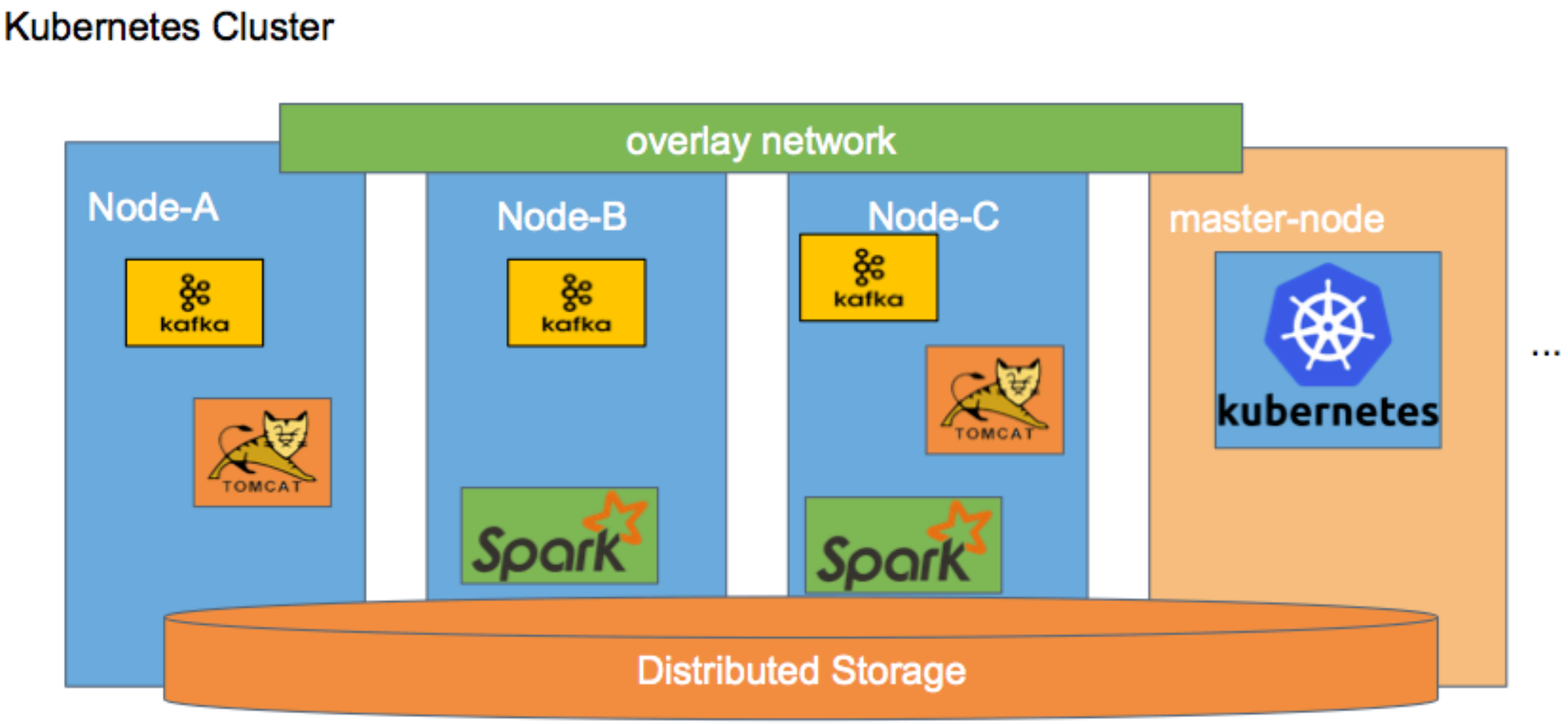
- HDFS/Ceph/GlusterFS
- Map-Reduce
- Storm -> Beam
- PaddlePaddle/Tensorflow

AI的应用须依赖于生产业务的数据流

- 专用集群会创建多个不同的集群，成本高昂
- 专用集群的每个集群利用率都难以提高
- 专用集群的在线伸缩能力弱

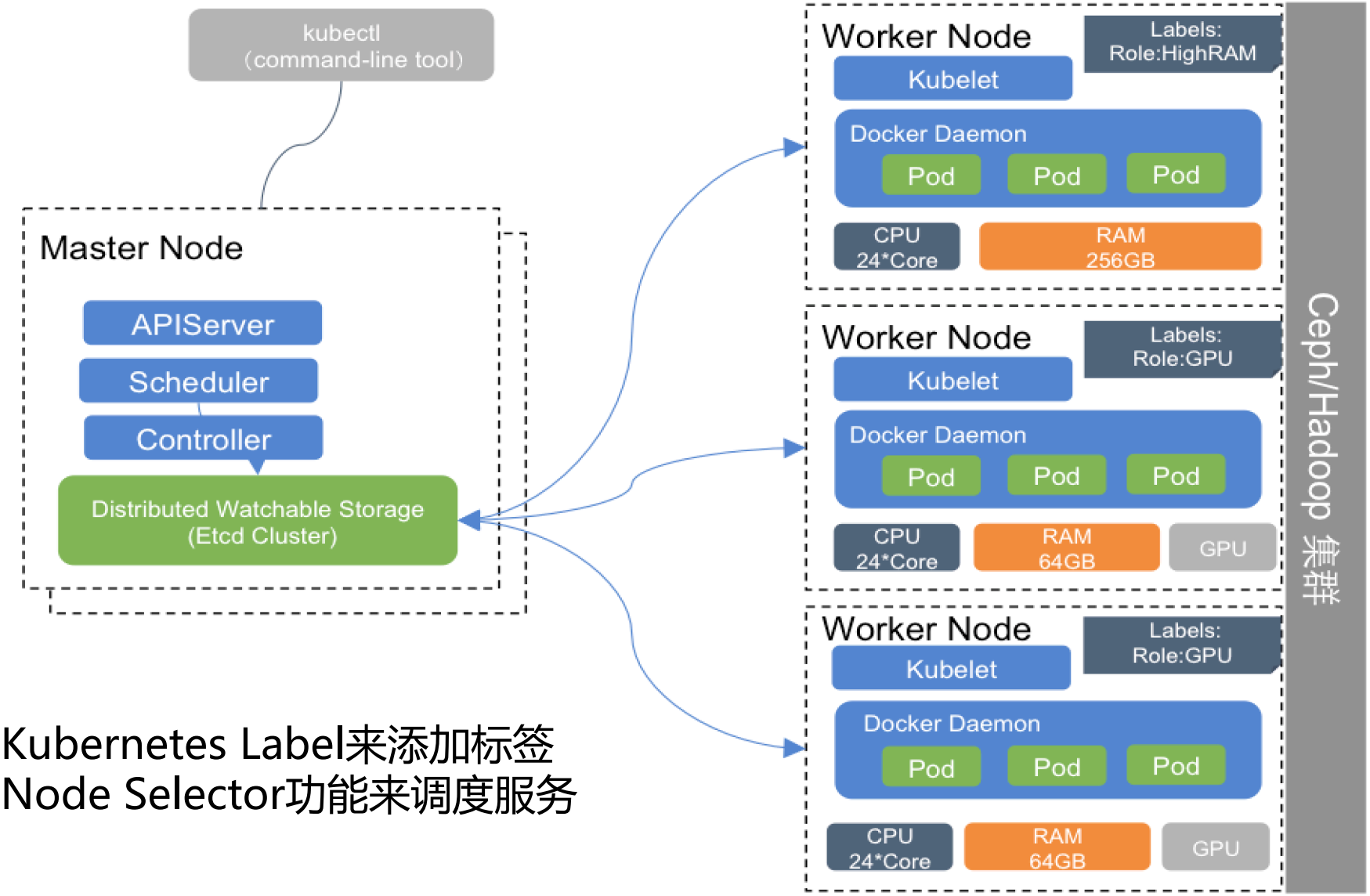


Why Kubernetes – AI通用集群



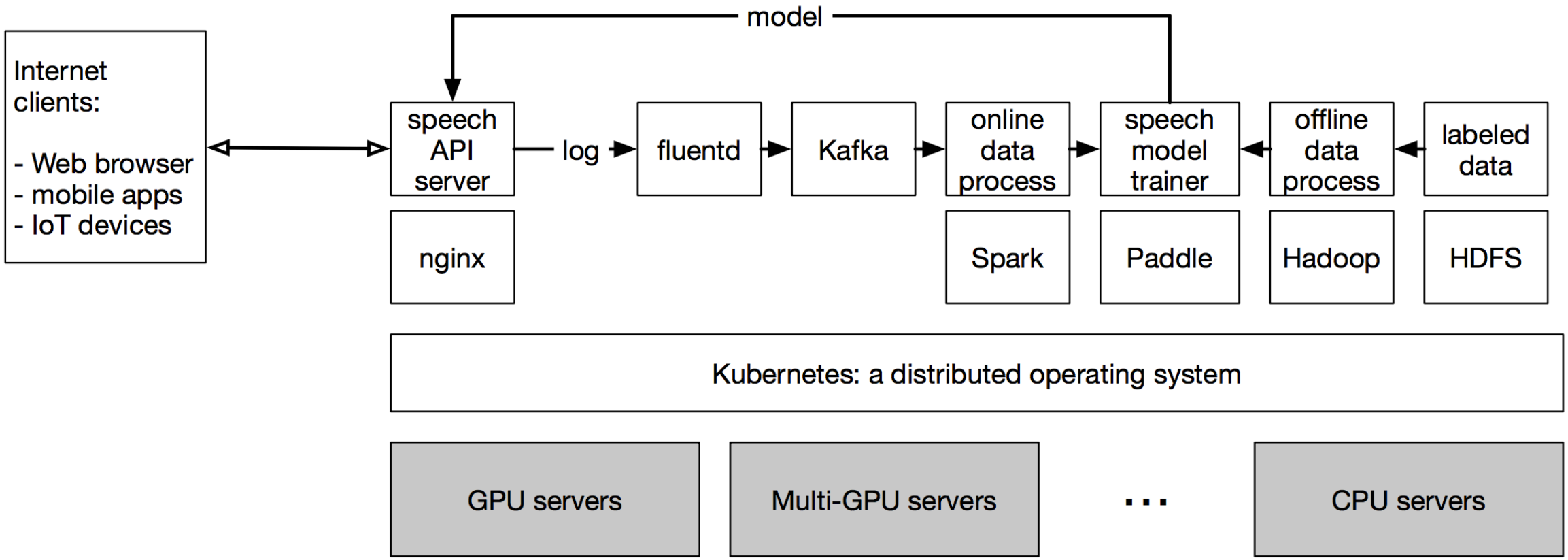
- 摒弃专用集群的概念，使集群资源无指定用途，达到资源共享、分时复用的目的；
- 不需要开发新的framework来适配Kubernetes，只需支持Containerization即可；

Why Kubernetes – 支持异构



- 使用Kubernetes Label来添加标签
- 使用Node Selector功能来调度服务

Why Kubernetes – AI解决方案



百度云

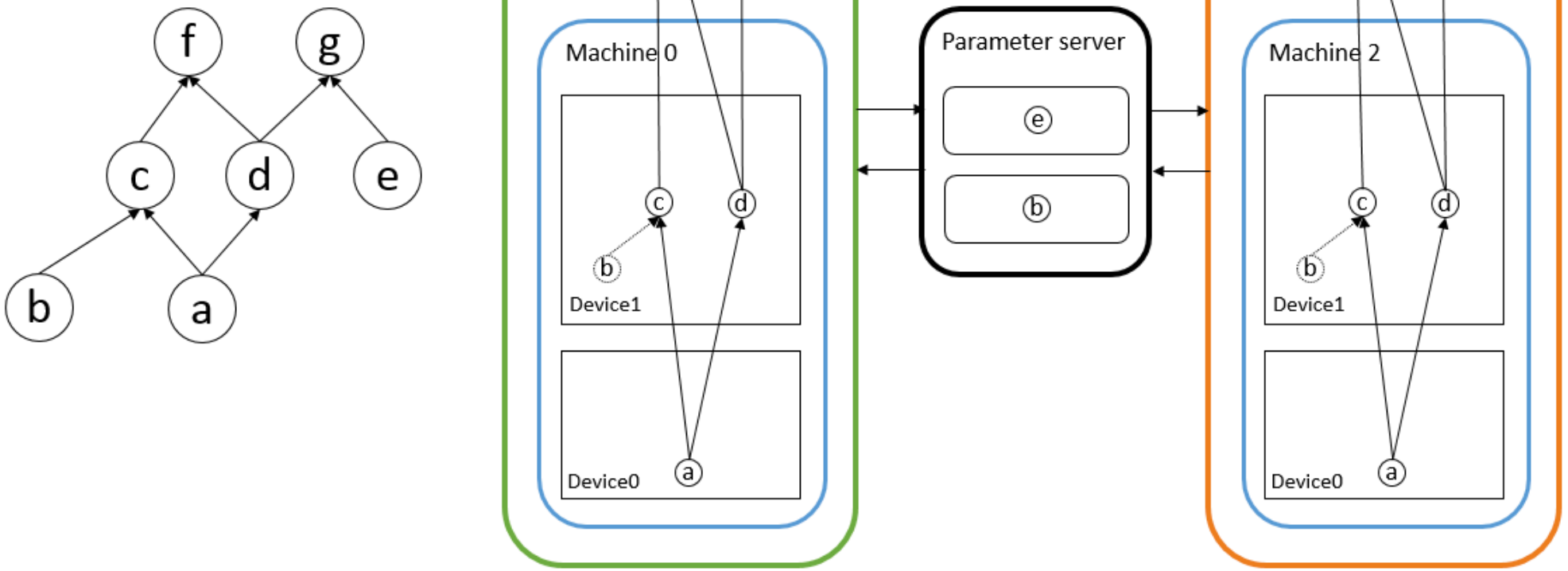
PaddlePaddle介绍

PaddlePaddle背景

- PaddlePaddle（PArallel Distributed Deep LEarning）是百度于2016年9月开源的一款深度学习平台，具有易用，高效，灵活和可伸缩等特点，为百度内部多项产品提供深度学习算法支持 - http://www.paddlepaddle.org/

- 已经应用于包括搜索、翻译、电商和计算基础架构等方面共五十余个应用，该项目开源地址 - https://github.com/PaddlePaddle/

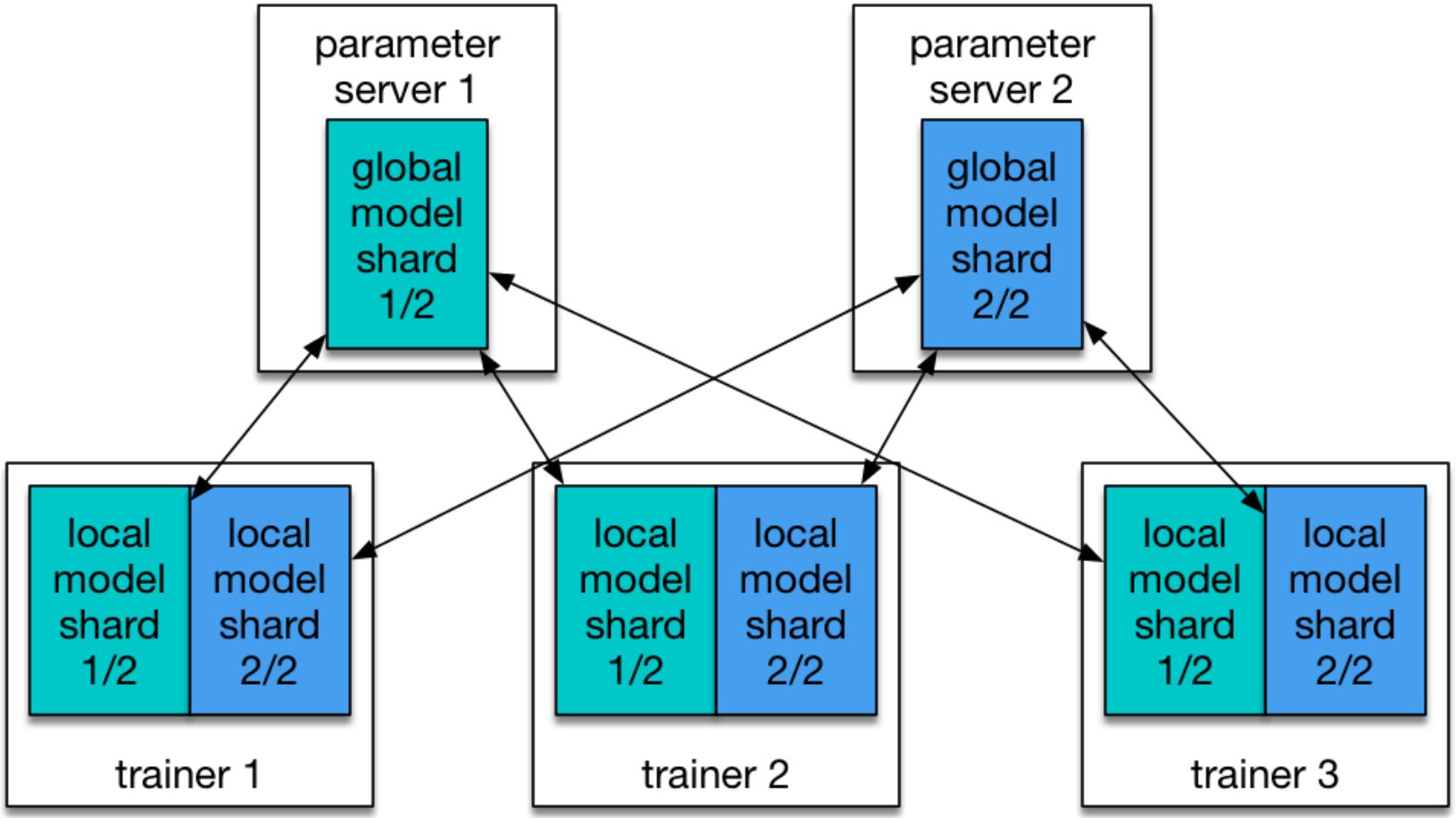
PaddlePaddle PServer架构



PaddlePaddle分布式结构

特点：

- 更为灵活的数据一致，pull可以并非等到push完全完成后再进行，可以允许一定的未完成度来确保并行。
- 相对MPI框架而言，有fault tolerant，节点fail不影响训练。



百度云

PaddlePaddle on Kubernetes技术细节

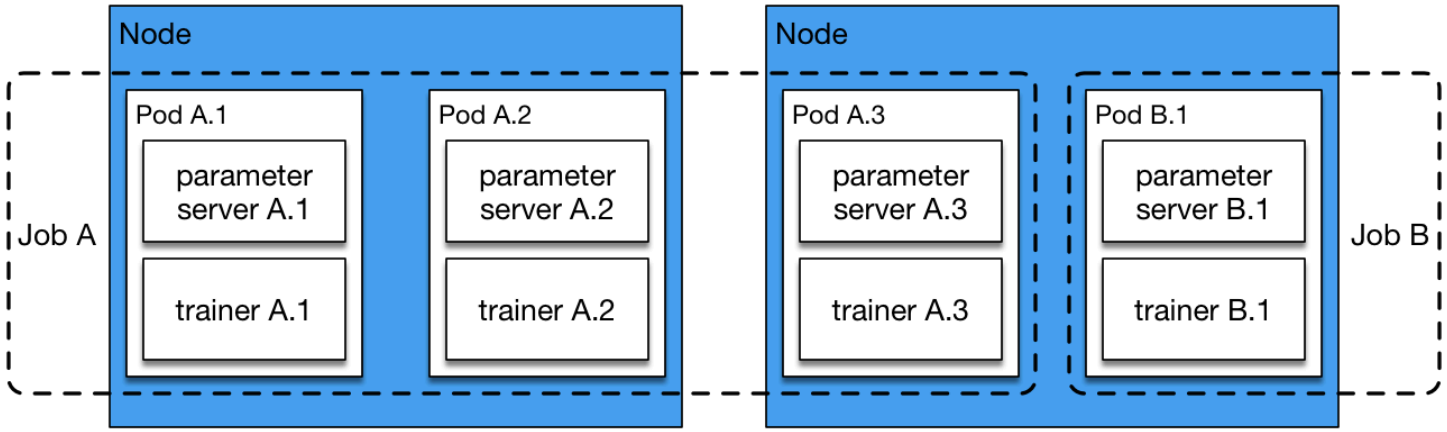
PaddlePaddle部署的挑战

- 各种类型的应用
  - PaddlePaddle Serving服务， PaddleBook服务
  - PaddlePaddle训练任务，Cronjob
- 服务之间的隔离
  - 线上线下PaddlePaddle共享集群
- 在离线混布 – 优先级抢占
  - Serving服务优于Training任务
- 集群资源利用率提升
  - CPU，内存等资源通过软硬限的方式进行超发
  - GPU作为一种资源进行统一的调度
- 自动伸缩
  - 白天Serving服务为主，夜间Training任务为主

分布式PaddlePaddle

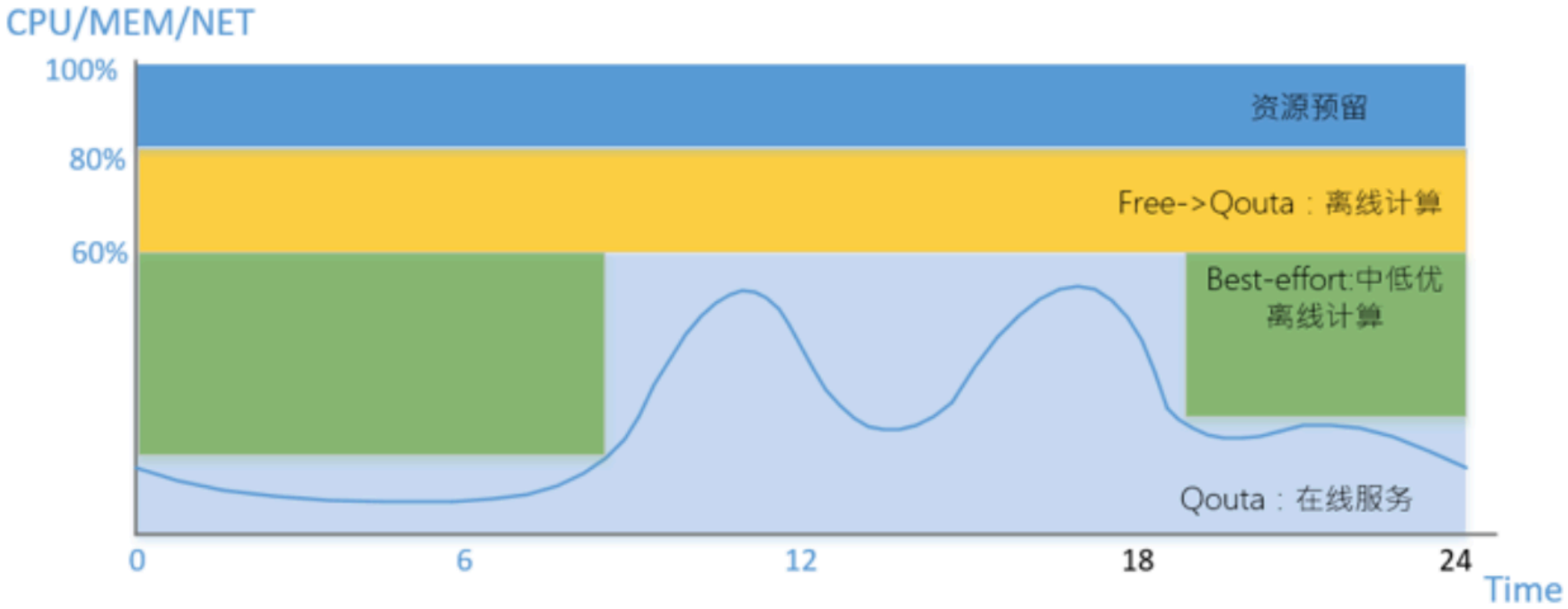
功能点：

- 内嵌式服务发现
- 可定义Job的并发数和完成度
- 可定义训练数据共享目录挂载点
- 可定义训练模型日志输出目录挂载点




```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: paddle-cluster-job-1
spec:
  parallelism: 3
  completions: 3
  template:
    metadata:
      name: paddle-cluster-job-1
    spec:
      volumes:
      - name: nfs
        persistentVolumeClaim:
          claimName: nfs
      containers:
      - name: trainer
        image: paddledev/paddle-tutorial:k8s_train
        command: ["/bin/bash", "/root/start.sh"]
        env:
        - name: SPLIT_COUNT
          value: "3"
        - name: JOB_NAME
          value: paddle-cluster-job
        - name: JOB_PATH
          value: /home/jobpath
        # using downward API to reference pod namespace
        - name: JOB_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
        - name: TRAIN_CONFIG_DIR
          value: quick_start
        - name: CONF_PADDLE_NIC
          value: eth0
        - name: CONF_PADDLE_PORT
          value: "7164"
        - name: CONF_PADDLE_PORTS_NUM
          value: "2"
        - name: CONF_PADDLE_PORTS_NUM_SPARSE
          value: "2"
        - name: CONF_PADDLE_GRADIENT_NUM
          value: "3"
        - name: TRAINER_COUNT
          value: "3"
        volumeMounts:
        - mountPath: "/home/jobpath"
          name: nfs
        ports:
        - name: jobport0
          hostPort: 7164
          containerPort: 7164
        - name: jobport1
          hostPort: 7165
          containerPort: 7165
        - name: jobport2
          hostPort: 7166
          containerPort: 7166
        - name: jobport3
          hostPort: 7167
          containerPort: 7167
      restartPolicy: Never
```

在离线混布 – 资源利用率



利用率提升方式：

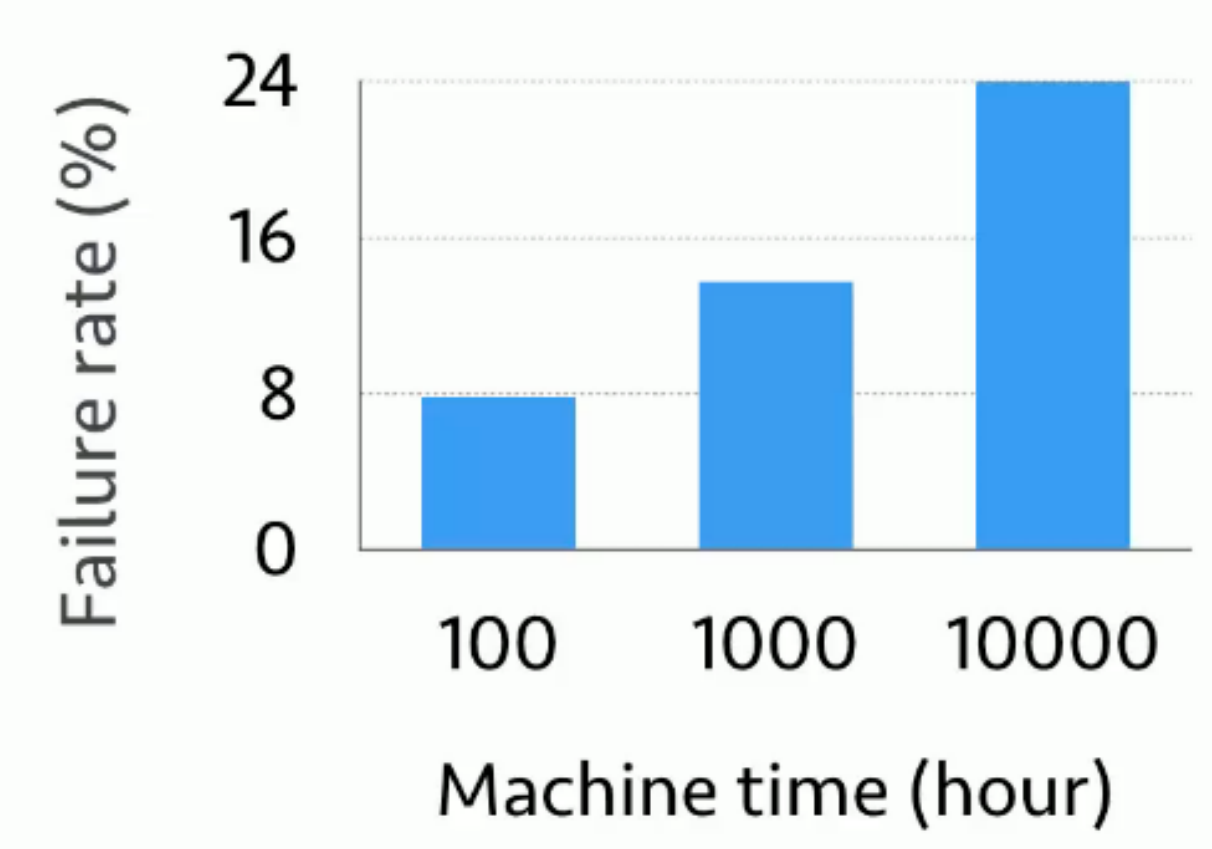
- Limit Quota & Request Quota
- 三层QoS
- 允许超发

PaddlePaddle在离线混布 – 问题

收益：
- 更高的资源利用率

遇到的问题：
- 高优先级在线服务会kill掉离线trainer
- 如果要等待所有trainer的gradients，训练作业就会halt

解决方案：
- 单个trainer fail不影响整体训练



PaddlePaddle在离线服务混布 – 解决方案

解决方案：

- 引入一个master进程；
- 负责把逻辑数据分片分发给“活着的”trainers进程；
- 即便一些trainers或者parameter servers被杀了（抢占了），训练作业也能继续进行；
- 当机群上其他高优先级作业结束之后，Kubernetes会增加作业里的trainers数量;

PaddlePaddle GPU支持

自研：

- 修改Kubelet以支持单机GPU调度能力
- 使用nodeSelector来调度到GPU机器

Kubernetes 1.6:

- Alpha支持GPU资源调度
- 仍然使用nodeSelector来匹配GPU机器

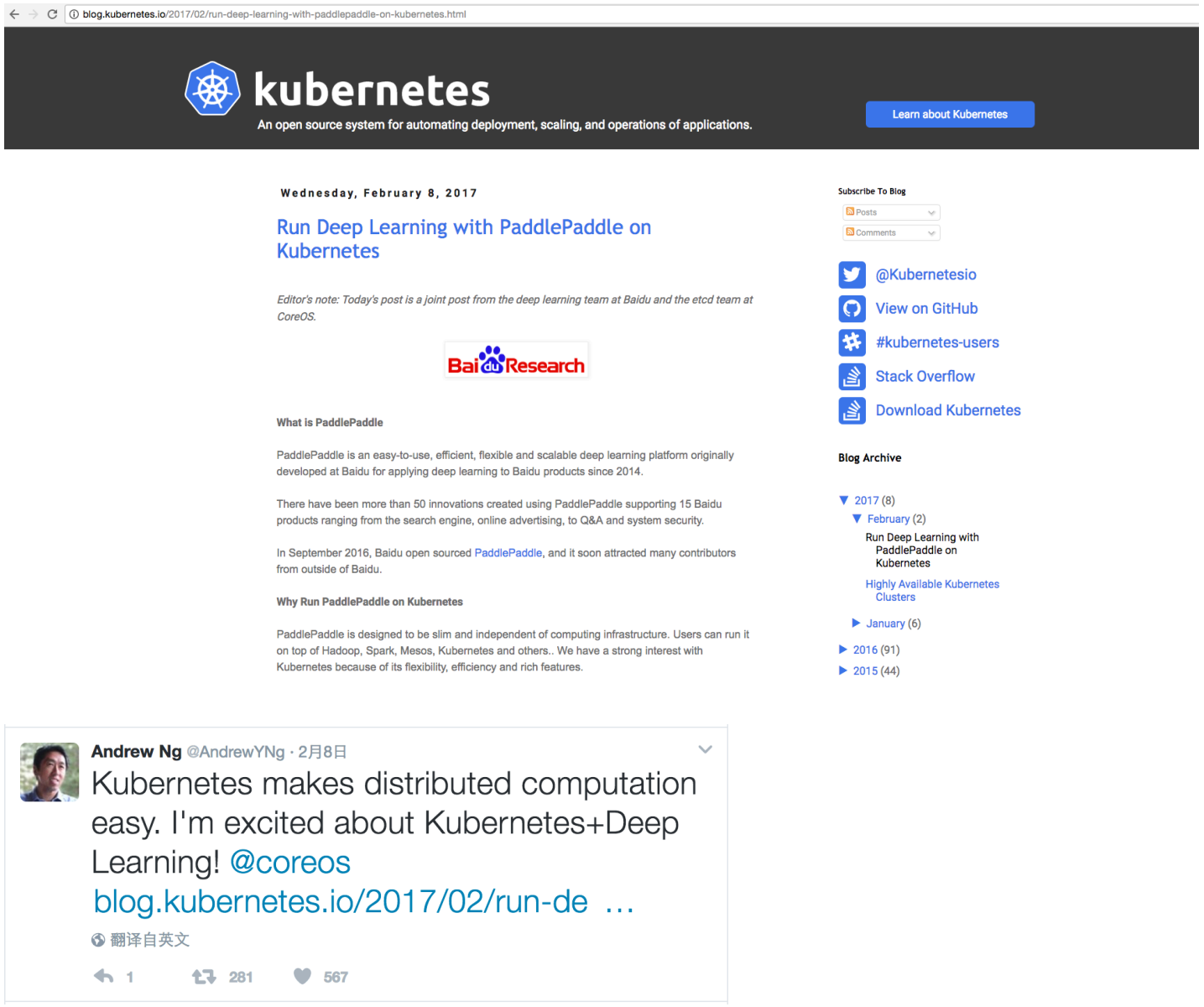
```
kind: pod
apiVersion: v1
spec:
 containers:
 - name: gpu-container-1
   resources:
      limits:
        alpha.kubernetes.io/nvidia-gpu: 2
 - name: gpu-container-2
   resources:
    limits:
      alpha.kubernetes.io/nvidia-gpu: 3
```

```
kind: pod
apiVersion: v1
metadata:
  annotations:
    scheduler.alpha.kubernetes.io/affinity: >
      {
        "nodeAffinity": {
          "requiredDuringSchedulingIgnoredDuringExecution": {
            "nodeSelectorTerms": [
              {
                "matchExpressions": [
                  {
                    "key": "alpha.kubernetes.io/nvidia-gpu-name",
                    "operator": "In",
                    "values": ["Tesla K80", "Tesla P100"]
                  }
                ]
              }
            ]
          }
        }
      }
spec:
 containers:
 - name: gpu-container-1
   resources:
    limits:
      alpha.kubernetes.io/nvidia-gpu: 2
```

PaddlePaddle 发表在 Kubernetes Blog

成果：

- 与Kubernetes, CoreOS等国际团队合作；
- 实现了分布式PaddlePaddle on 百度云/AWS解决方案；
- Kubernetes Blog：http://blog.kubernetes.io/2017/02/run-deep-learning-with-paddlepaddle-on-kubernetes.html



百度云

深度学习未来展望

深度学习未来展望

PaddlePaddle：

- 参考MxNet, Keras, Tensorflow
- 优化api（已完成）
- 优化分布式计算方法

Kubernetes:

- 更高级的资源管理
- 更好地支持任务的管理和调度

百度云：

- CDS, VPC等 Kubernetes plugin回馈社区

百度云

Q & A

THANK YOU

cloud.baidu.com