SACC

2020 中国系统架构师大会

SYSTEM ARCHITECT CONFERENCE CHINA 2020

架构融合 云化共建

live 2020年10月22日 - 24日网络直播

云上会议

58同城AI算法平台的演进与实践

陈兴振 2020-10-24

目录

- **58同城AI算法平台演进**
- 大规模分布式机器学习
- 深度学习平台架构实践
- GPU/CPU上推理性能优化
- GPU资源调度优化

58AI算法平台

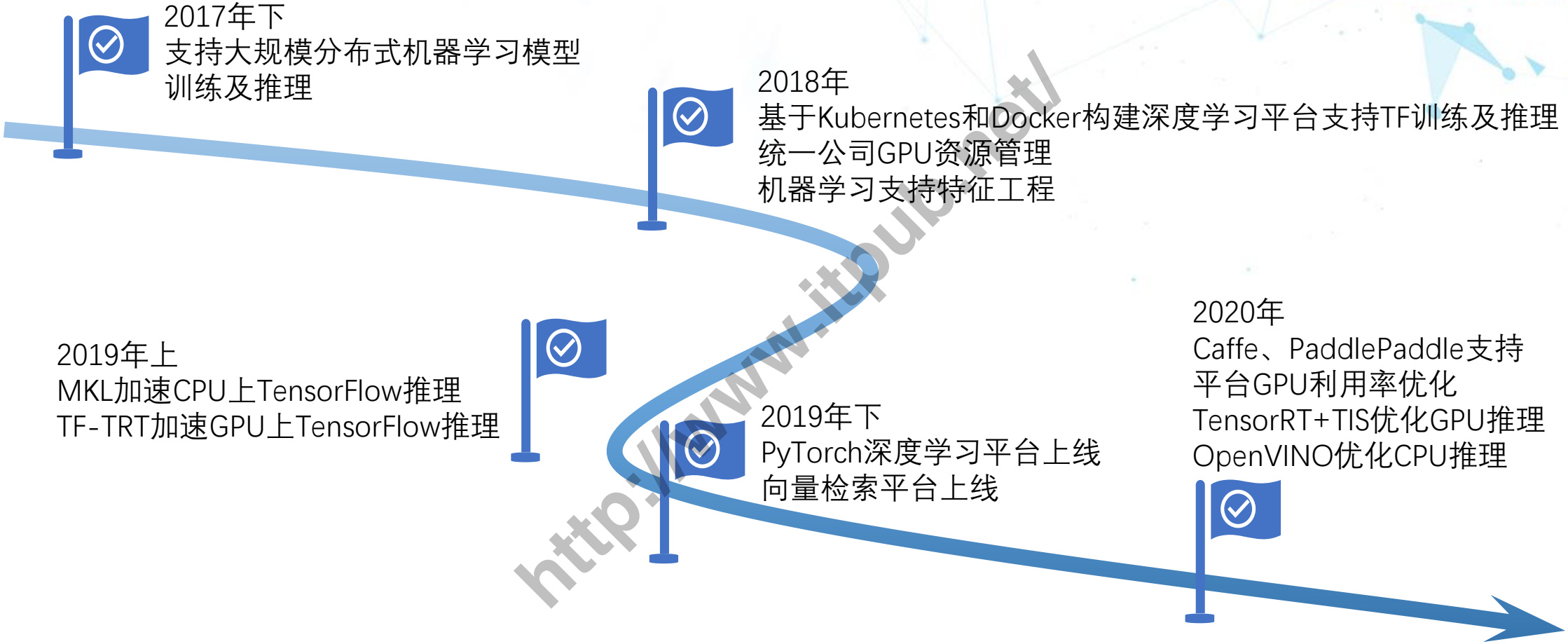
- 58同城AI Lab架构师
- 2016年加入58，目前主要负责AI算法平台及周围子系统的建设工作

搜索　推荐　NLP　语音　图像

58AI算法平台

58 AI Lab公共号

58AI算法平台演进历程



58AI算法平台规模

500+张
集群GPU卡数

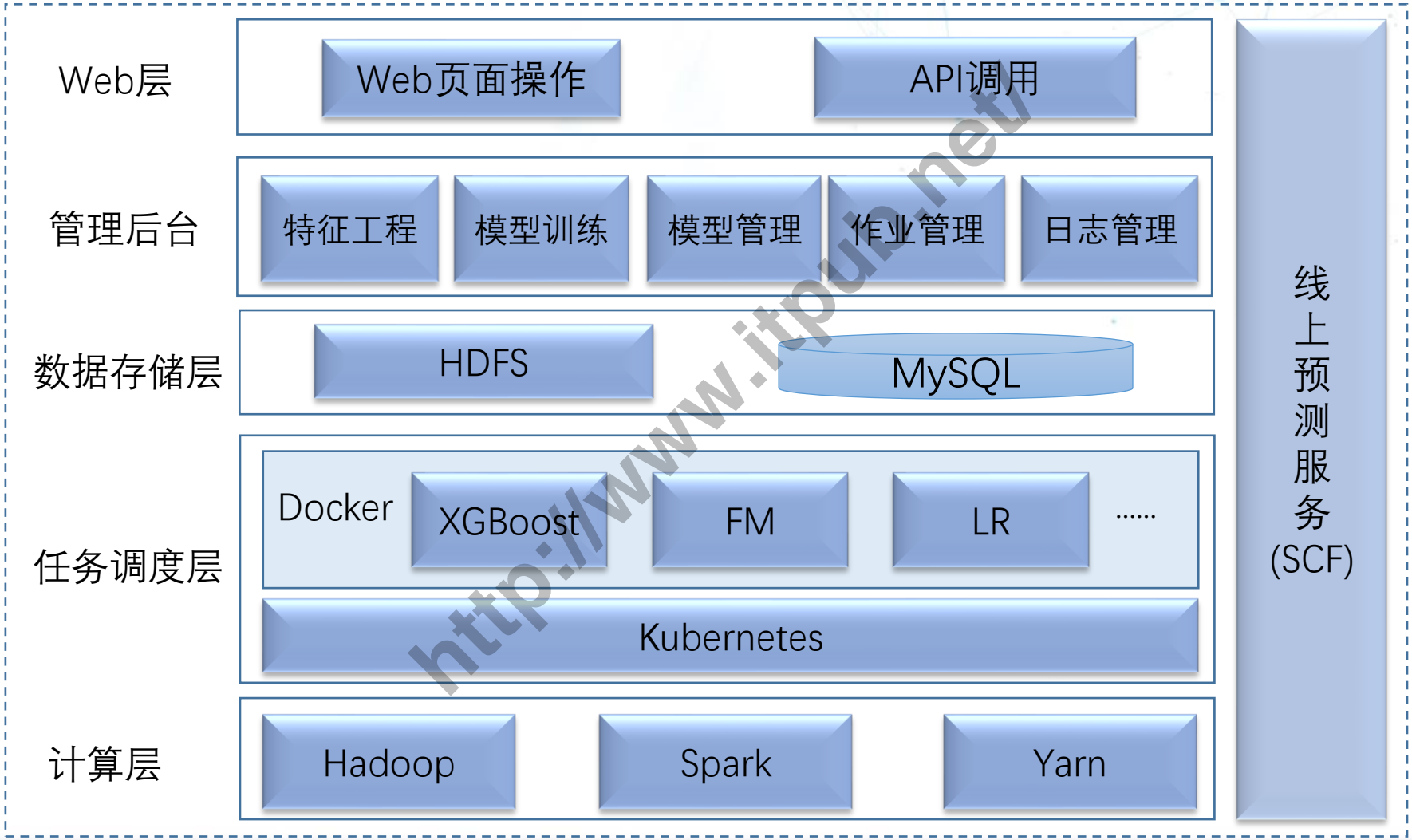
600+
线上模型数

50亿
每天推理流量

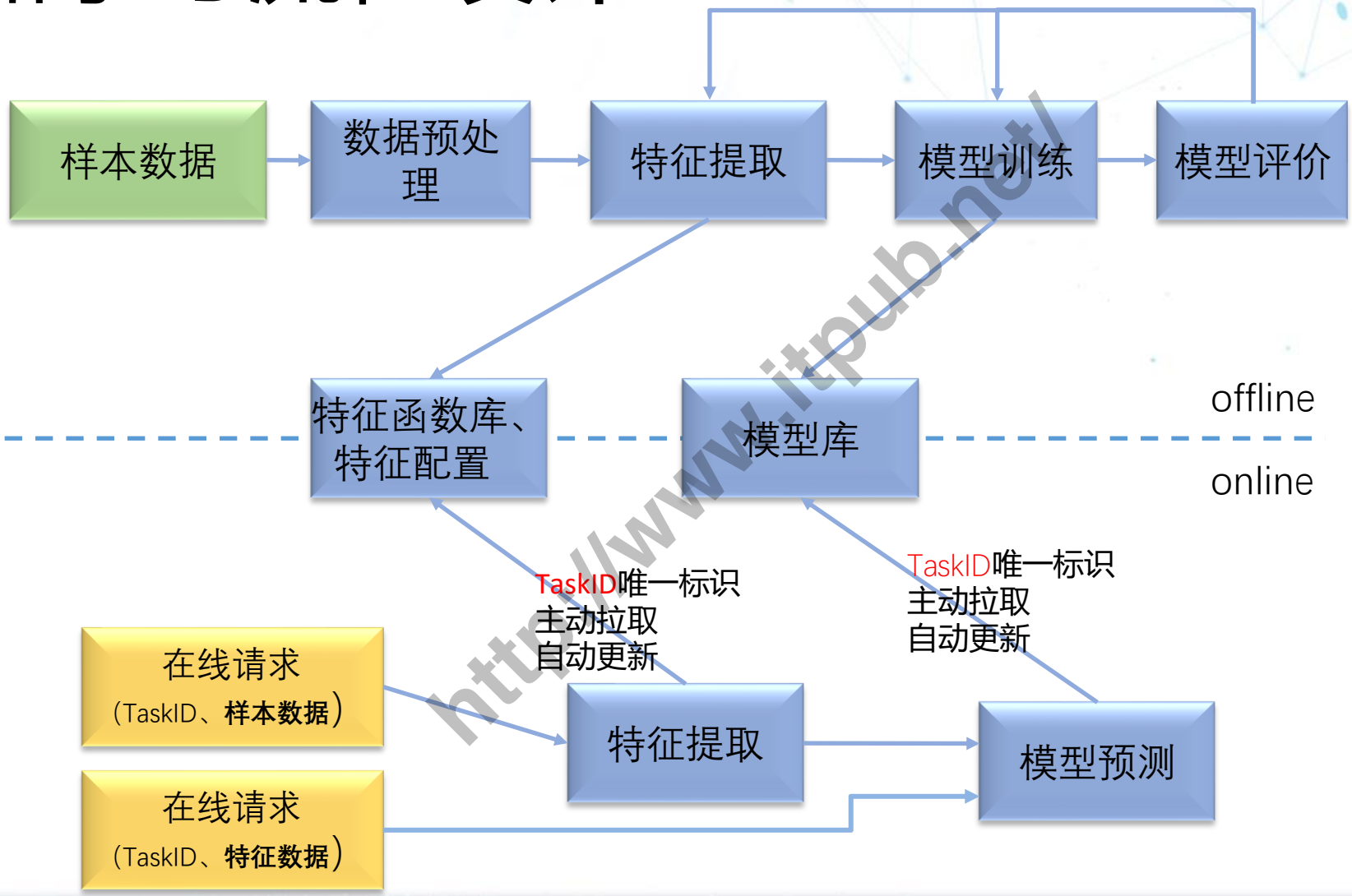
目录

- 58同城AI算法平台演进介绍
- **大规模分布式机器学习**
- 深度学习平台架构实践
- GPU/CPU上推理性能优化
- GPU资源调度优化

机器学习架构设计



机器学习流程设计



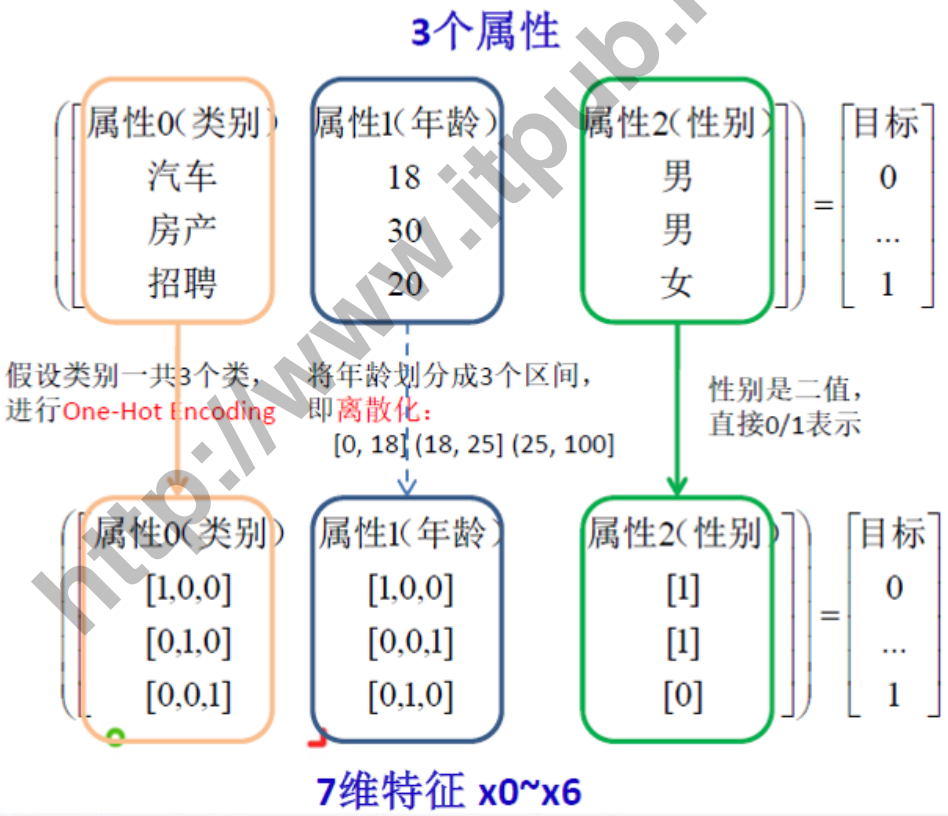
特征工程

- 特征抽取：将样本集合的属性转换为数学表示，常见的特征抽取方法：
  - One-Hot Encoding
  - 离散化
  - 归一化
  - 特征交叉

特征工程-特征抽取举例

- 通过One-Hot Encoding将类别属性映射成 x0~x2这三维特征
- 通过离散化将年龄属性映射成x3~x5这三维特征

3个属性映射成7维特征



特征工程平台实现流程

特征配置

特征提取列表

特征: 0	提取字段:	3	提取方法:	discreteWithZone	等频处理 为 20 份
特征: 1	提取字段:	4	提取方法:	oneHotWithHashBucket	Bucket大小:500
特征: 2	提取字段:	6	提取方法:	oneHotWithInt	
特征: 3	提取字段:	7	提取方法:	sentenceEmbeddingWithBow	平台分词
特征: 4	提取字段:	11,12	提取方法:	generateCrossFeature	
特征: 5	提取字段:	10	提取方法:	oneHotWithEnum	枚举值:male,female

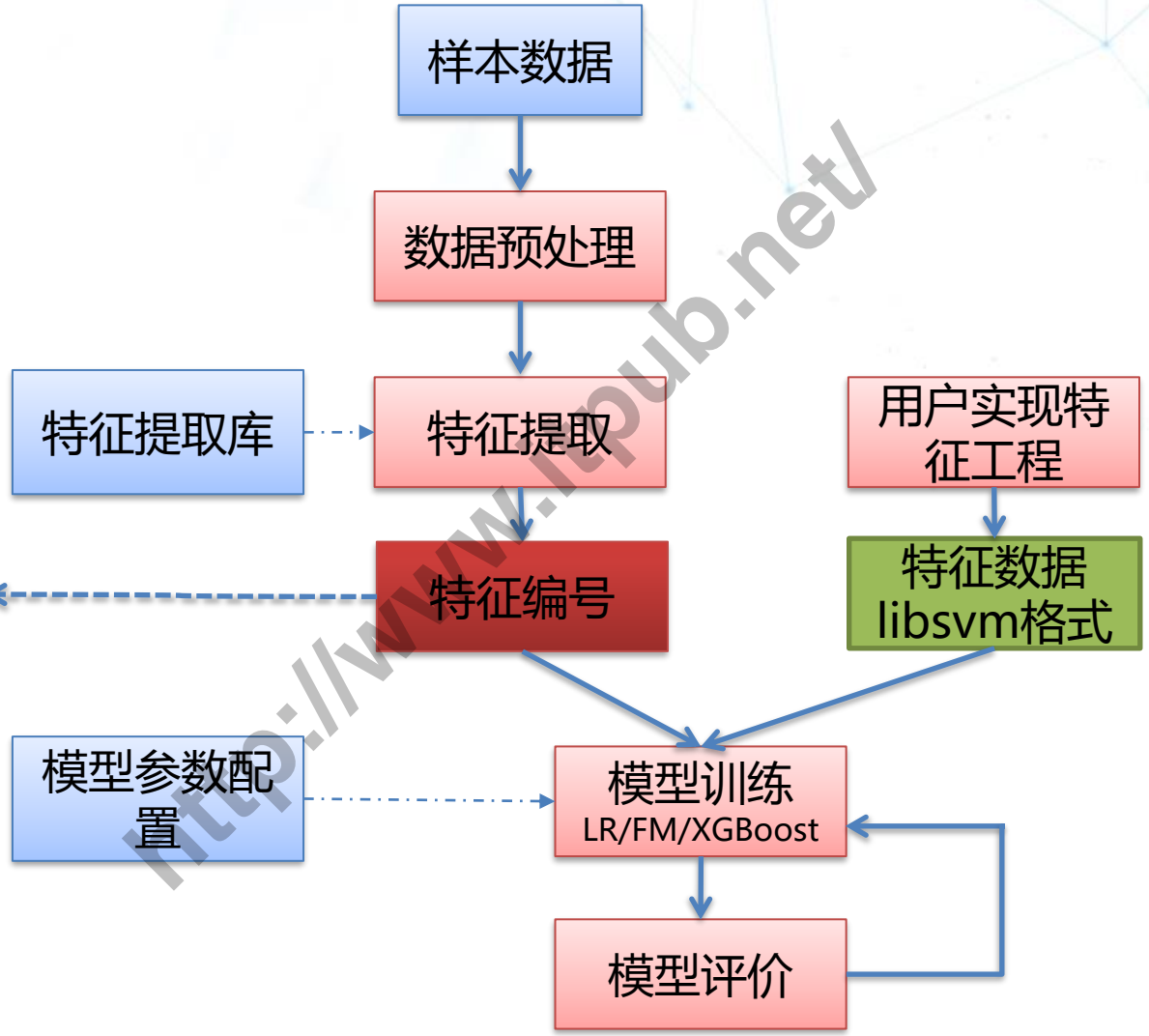
提取函数

```
featureMethodNameMap.put(1, "notDiscrete");
featureMethodNameMap.put(2, "generateBinaryFeature");
featureMethodNameMap.put(3, "oneHotWithEnum");
featureMethodNameMap.put(4, "discreteWithZone");
featureMethodNameMap.put(5, "oneHotWithInt");
featureMethodNameMap.put(6, "oneHotWithHashBucket");
featureMethodNameMap.put(7, "generateCrossFeature");
featureMethodNameMap.put(8, "sentenceEmbeddingWithBow");
```

```
0  0:5:1.0 1:157:1.0 2:101:1.0 3:2866863778:1.0 4:55:1.0 5:71:1.0 6:606:1.0 7:4651:1.0
1  0:3:1.0 1:167:1.0 2:153:1.0 3:2979718301:1.0 4:31:1.0 5:54:1.0 6:0:1.0 7:106663:1.0
0  0:1:1.0 1:37:1.0 2:15:1.0 3:2883437165:1.0 4:106:1.0 5:16:1.0 6:802:1.0 7:4663:1.0
0  0:5:1.0 1:155:1.0 2:153:1.0 3:2721528093:1.0 4:20:1.0 5:65:1.0 6:1516:1.0 7:161300:1.0
```

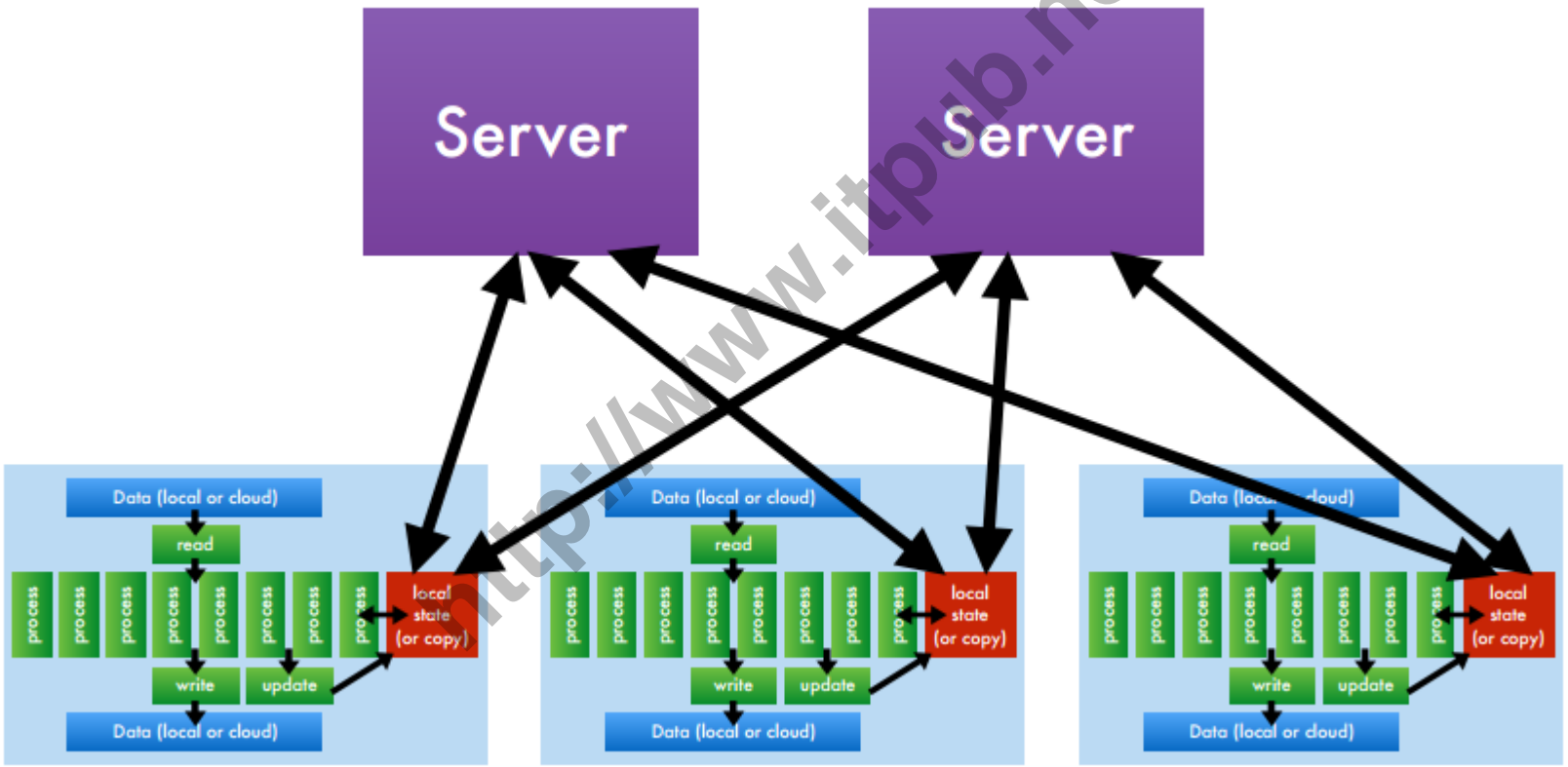
模型训练

特征编号	特征原始表示	出现次数
1	0^C1	1039
2	0^C2	1132
3	0^C3	1356
4	0^C4	1001
5	0^C5	993
6	0^C6	1074
7	0^C7	1061
8	0^C8	918
9	0^C9	1058
10	0^C10	946
11	0^C11	946
12	0^C12	1815
13	0^C13	912
14	0^C14	1194
15	0^C15	1830
16	1^C0	127
17	1^C1	38
18	1^C2	63



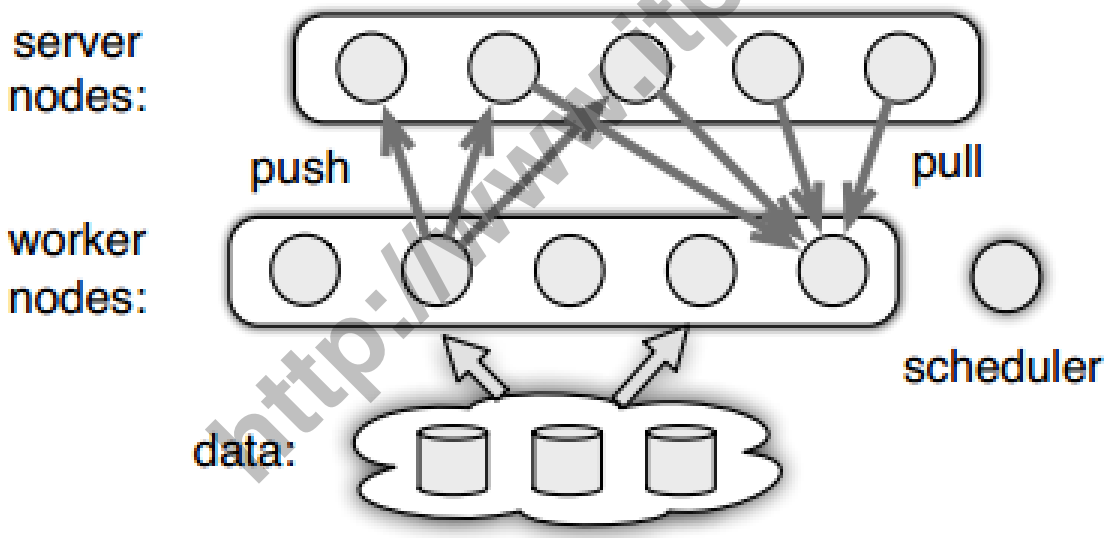
分布式FM实现

- 基于ps参数化服务器



分布式FM实现

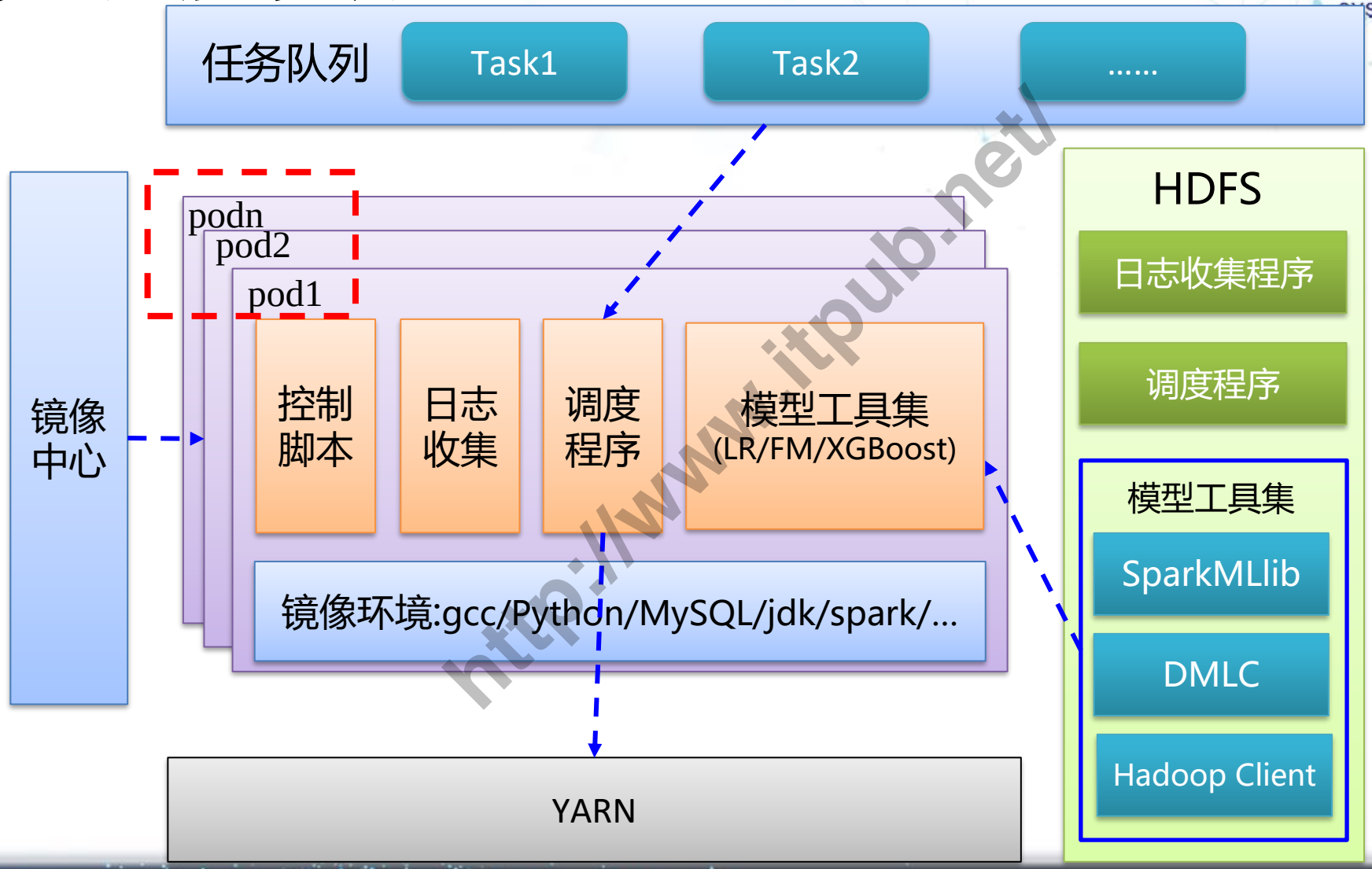
- Push: worker结点将用本地数据计算好的梯度推给指定sever结点
- Pull: worker结点从sever结点请求本地需要的模型参数



XGBoost分布式实现

- 基于RABIT[Reliable Allreduce and Broadcast Interface]
- 按行分割数据
- 并行计算每个特征值的最大最小值，并同步到各结点
- 各个结点进行直方图统计,父结点进行聚合，根结点找到最优分隔特征，并分发到各个结点

任务调度实现

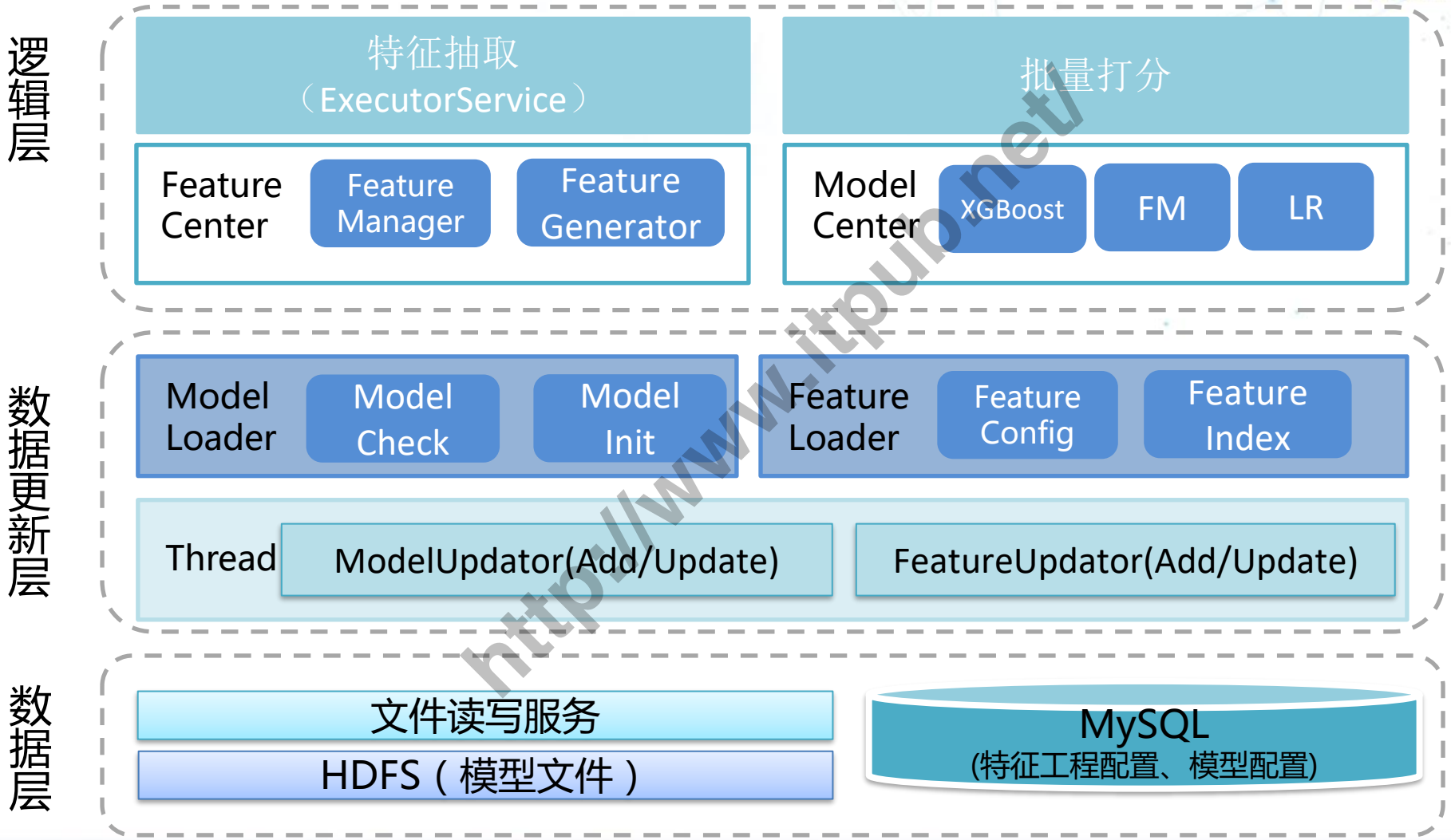


在线预测服务-设计思想

- 阿里预测平台RTP：平台存储样本特征数据，输入样本ID即可预测

- 我们的方案：平台不存储样本特征数据，线上完成转换

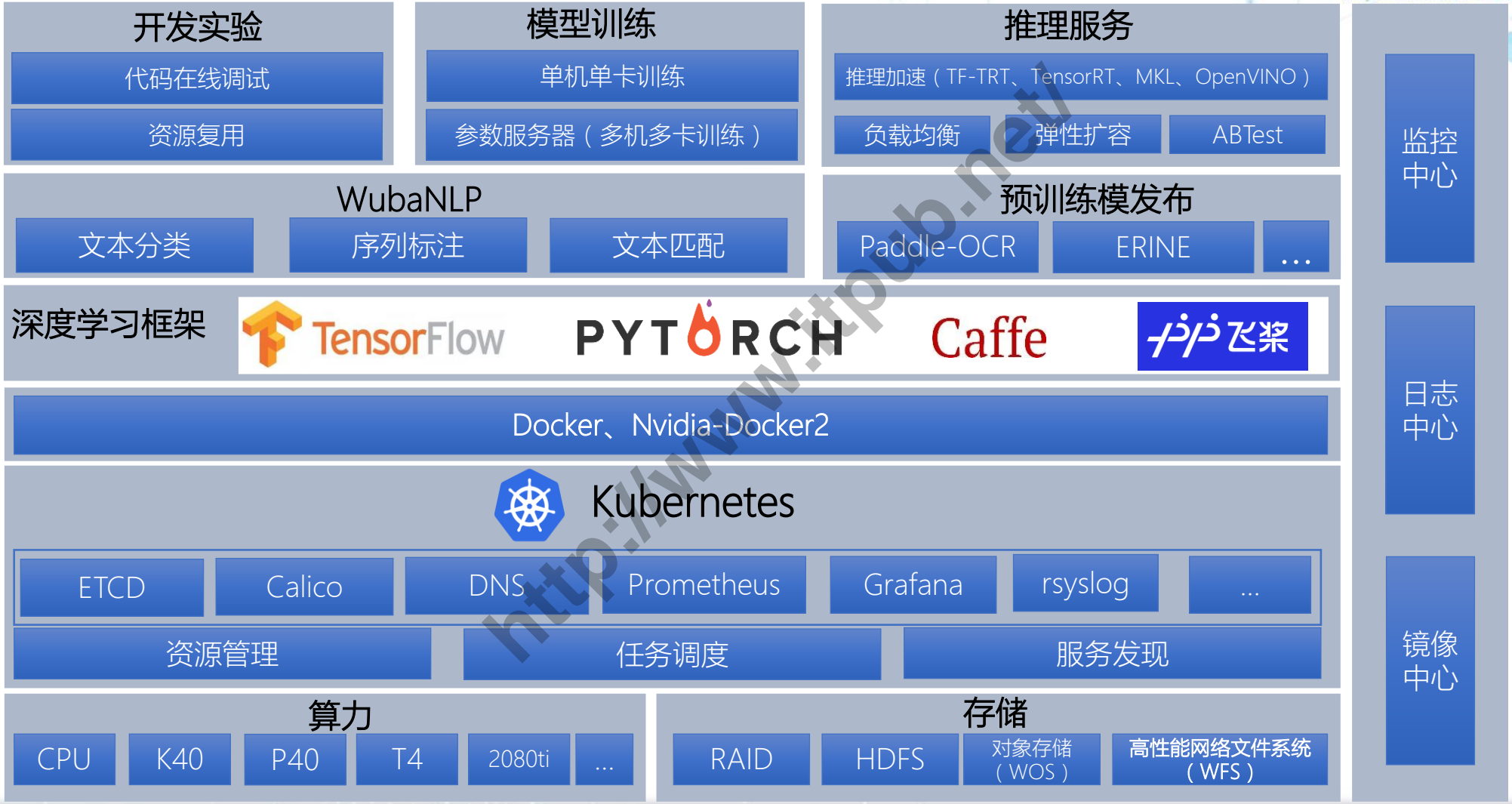
在线预测服务-总体设计



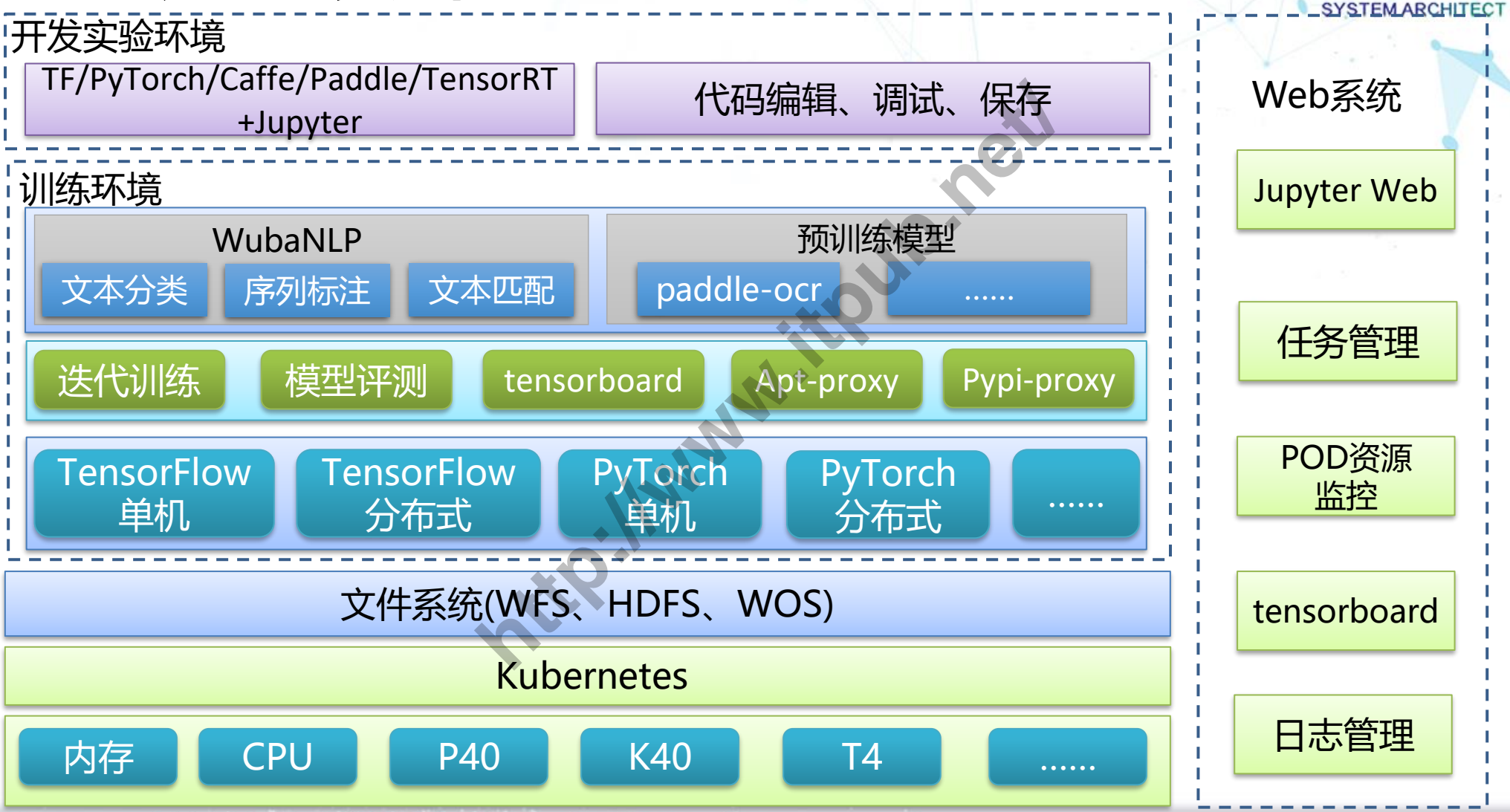
目录

- 58同城AI算法平台演进介绍
- 大规模分布式机器学习
- **深度学习平台架构实践**
- GPU/CPU上推理性能优化
- GPU资源调度优化

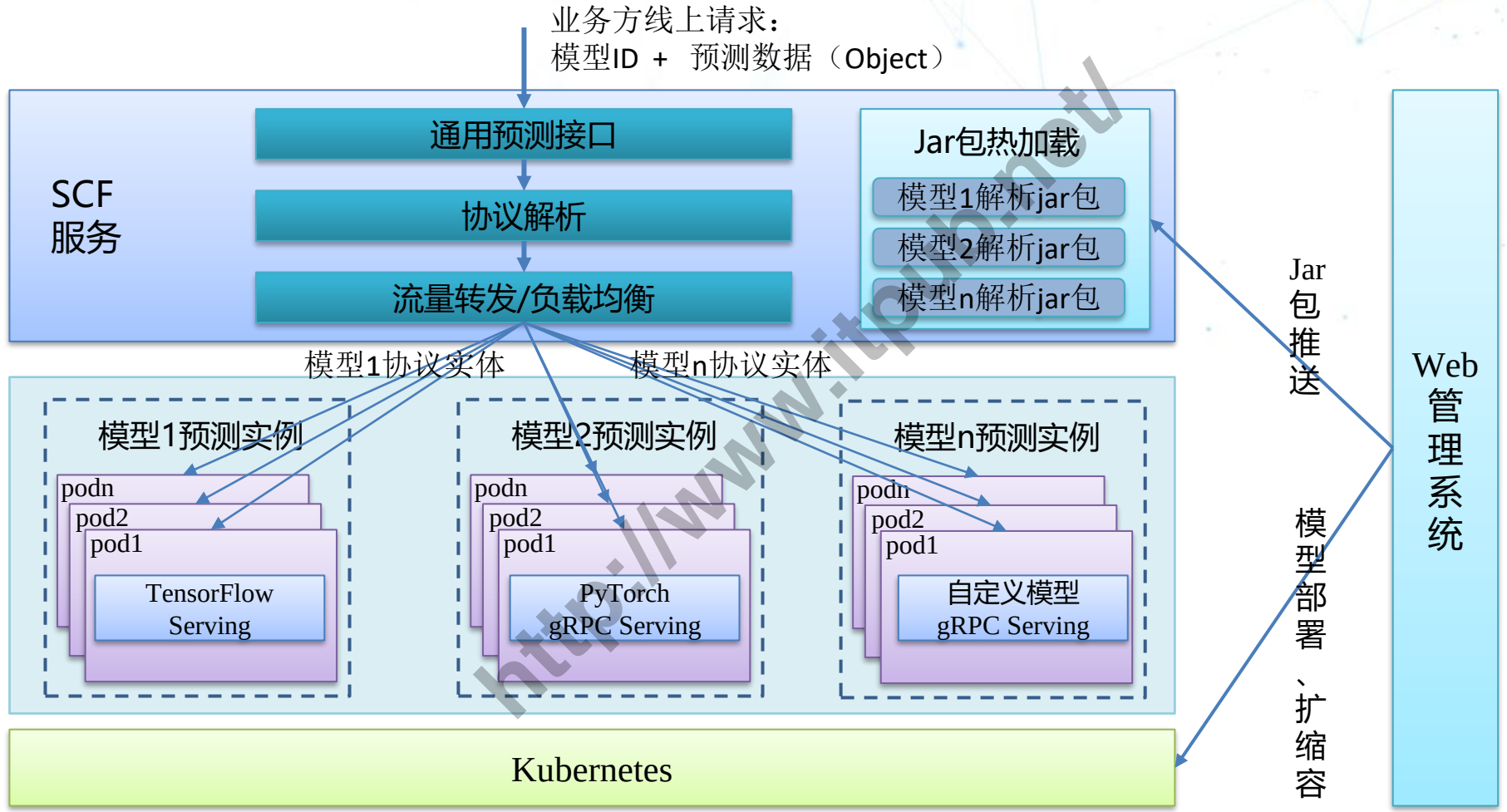
深度学习平台整体架构



离线训练设计

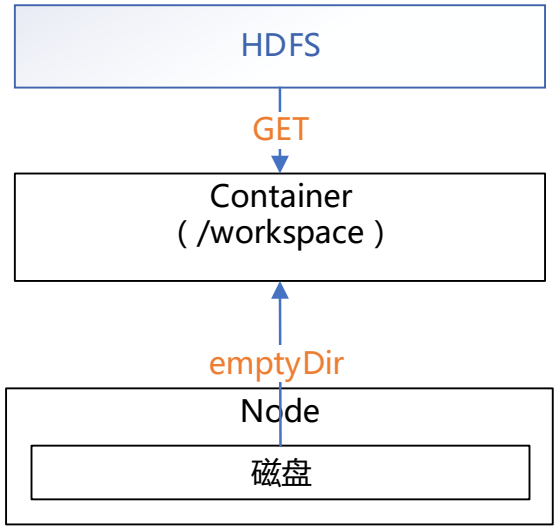


推理服务设计

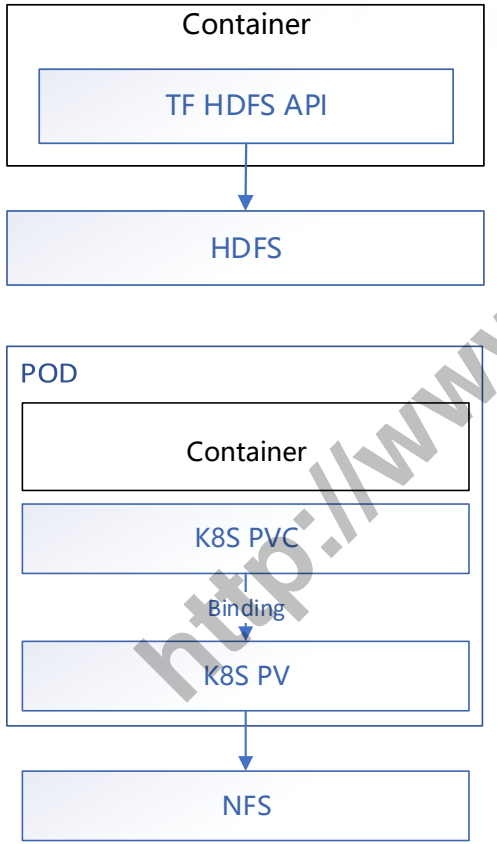


AI文件存储系统

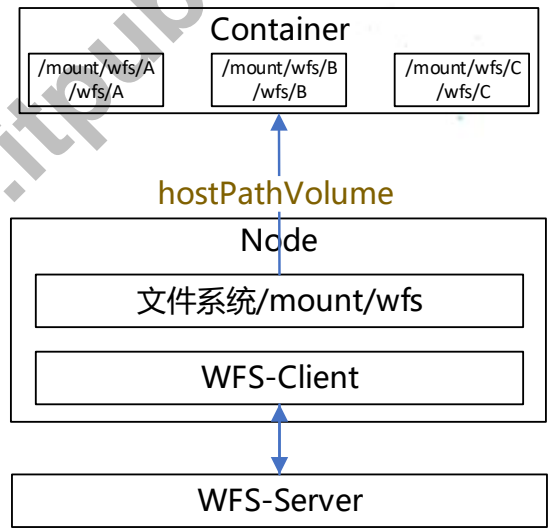
第一阶段：Local



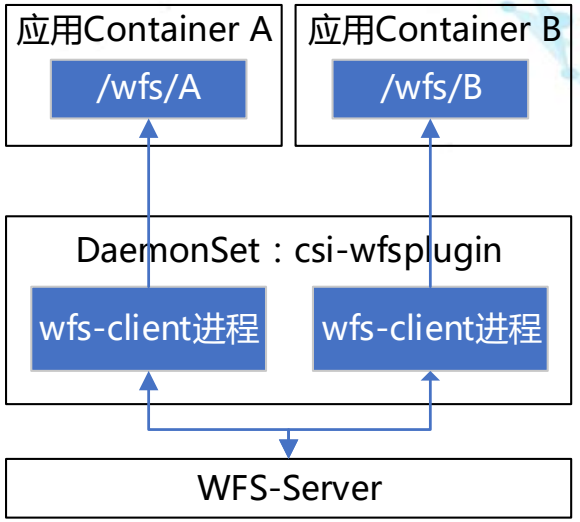
第二阶段：HDFS + NFS



第三阶段：WFS

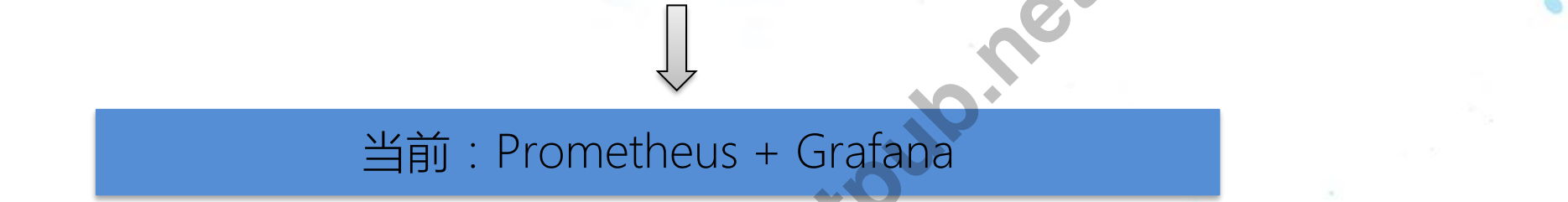


第四阶段：WFS-CSI

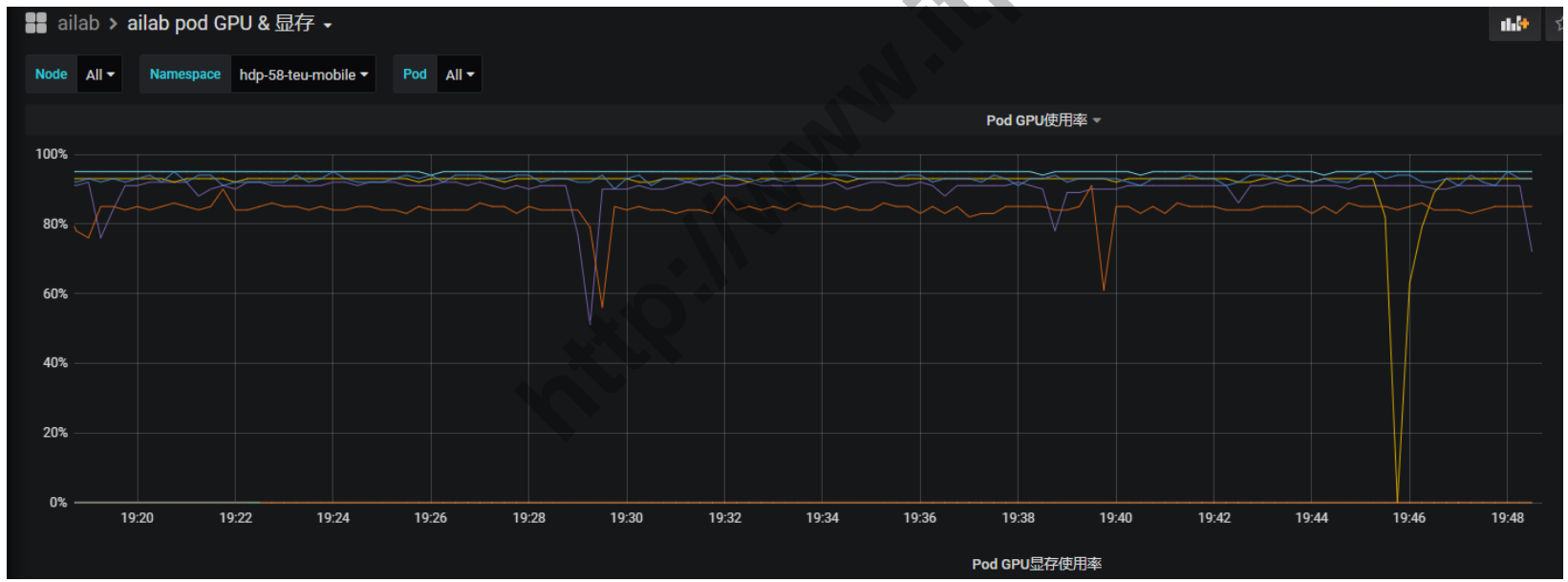


平台监控

前期：Heapster + InfluxDB + Grafana + MySQL



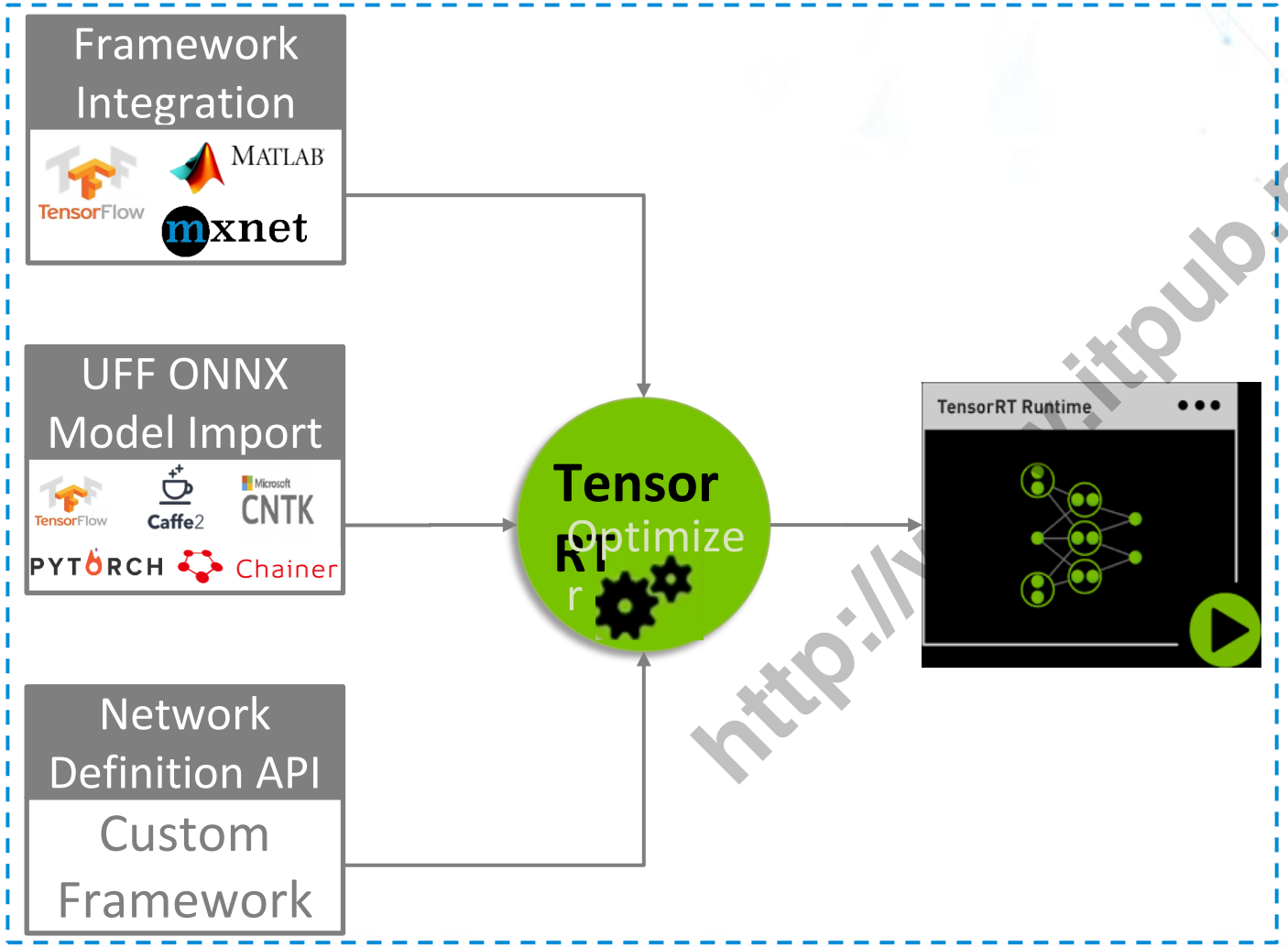
当前：Prometheus + Grafana



目录

- 58同城AI算法平台演进介绍
- 大规模分布式机器学习
- 深度学习平台架构实践
- **GPU/CPU上推理性能优化**
- GPU资源调度优化

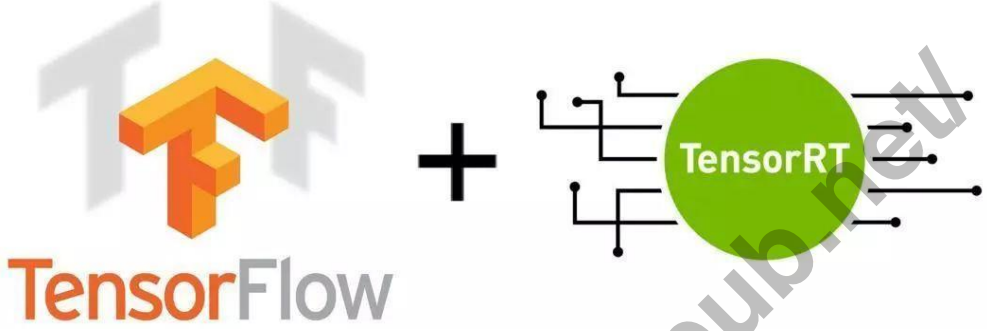
GPU上推理性能优化历程

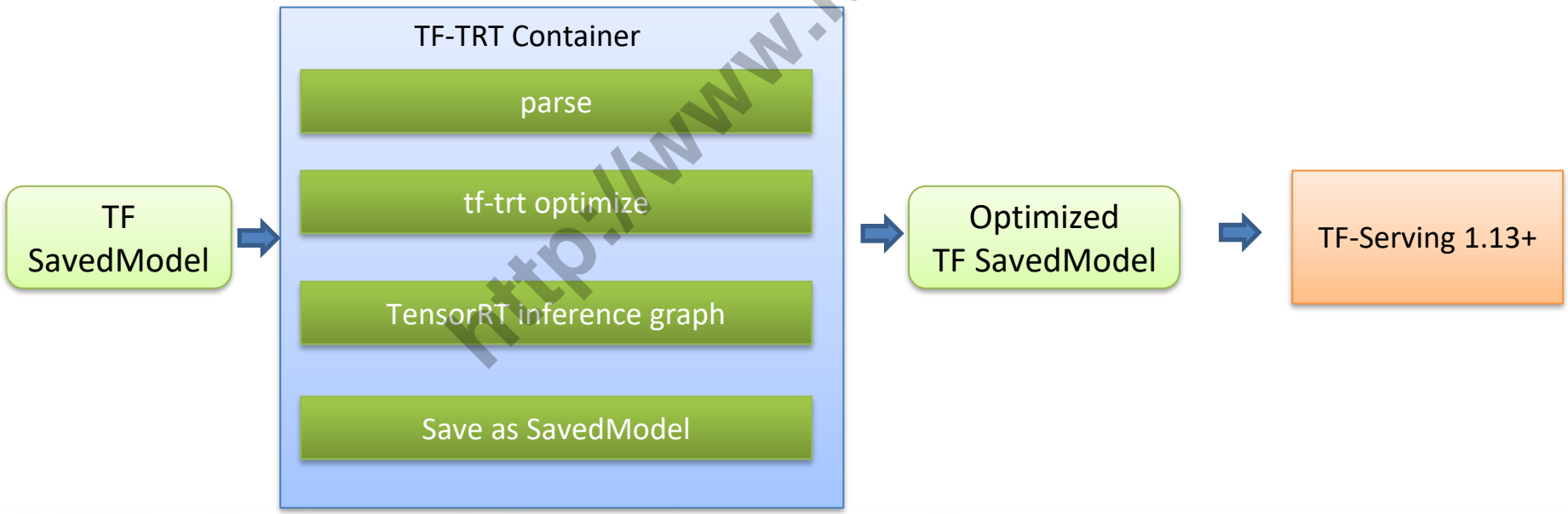


前期：主要支持TensorFlow框架
采用TF-TRT快速上线

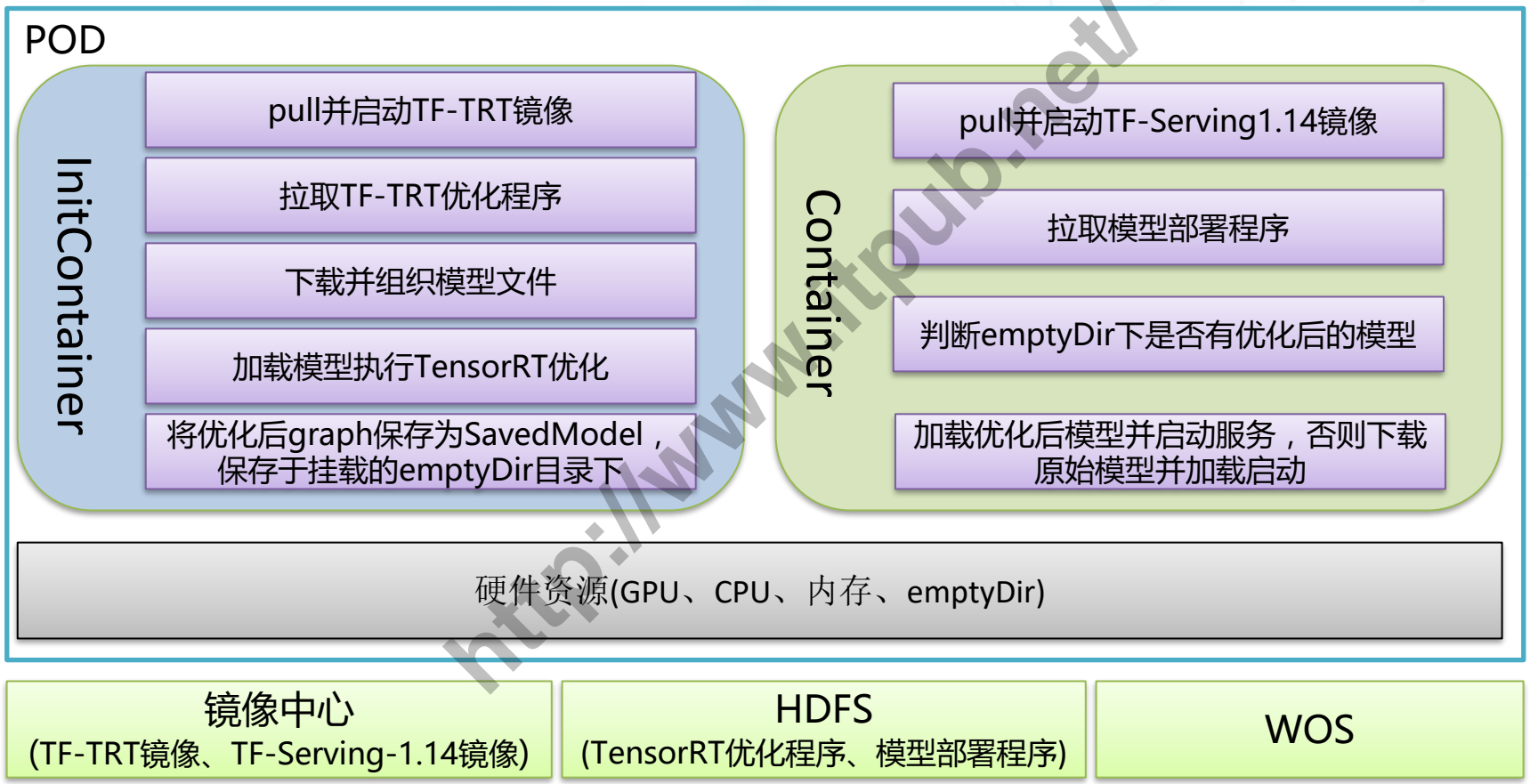
后期：TensorRT+Triton Inference Server支持所有框架

GPU上推理性能优化：TF-TRT应用





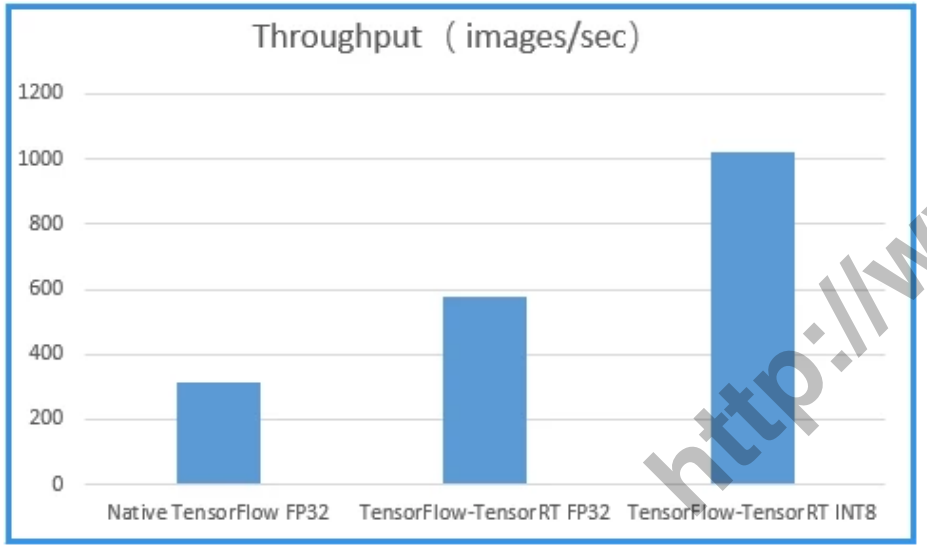
TF-TRT平台应用

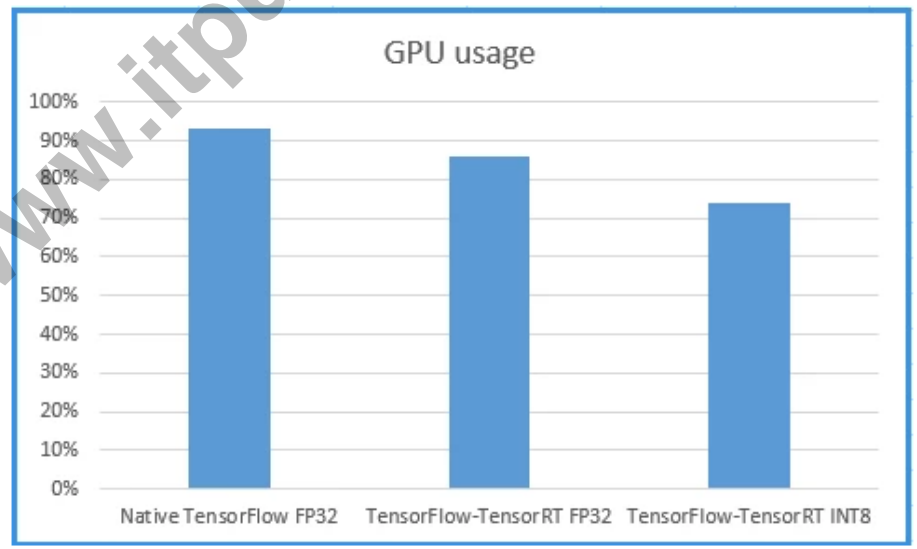


TF-TRT优化效果-Resnet50-v1

- 单张P40卡Resnet50-v1模型对比：数据集ImageNet5万张图片， BatchSize为8

FP32精度下性能提升**1.8**倍，INT8精度下 性能提升**3.2**倍

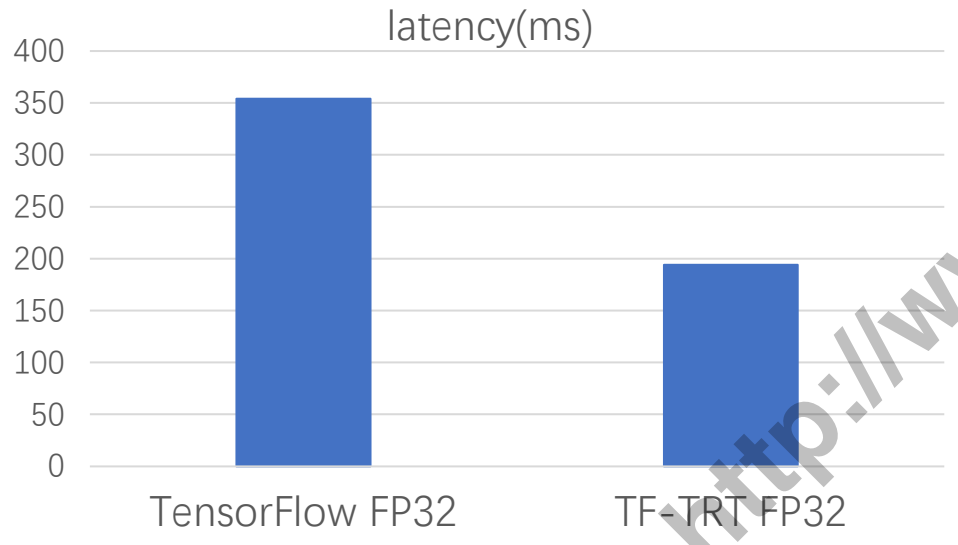


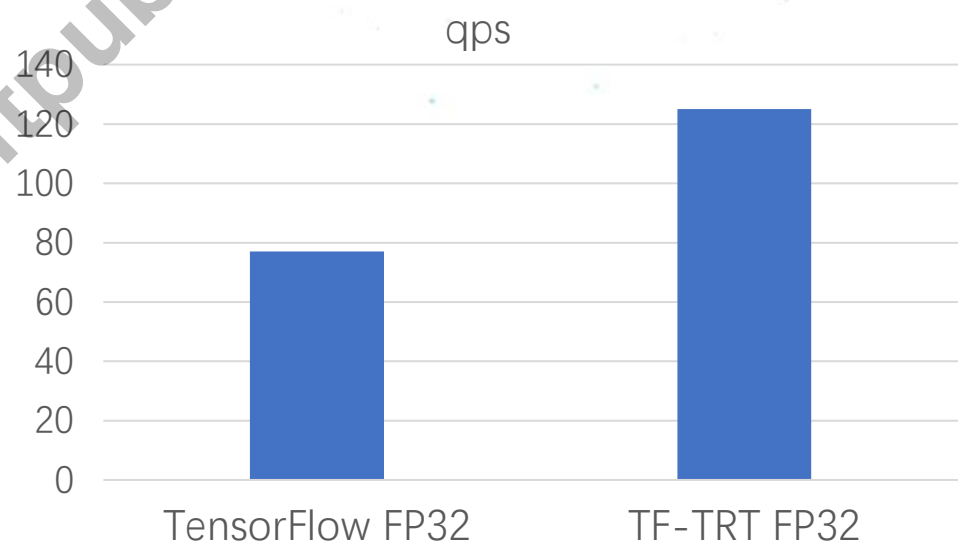


准确率： FP32 76.29% VS INT8 76.09%

TF-TRT优化效果-图像OCR

- NVIDIA Tesla P40 上优化后 耗时降低45%，QPS提升60%





TF-TRT面临的问题

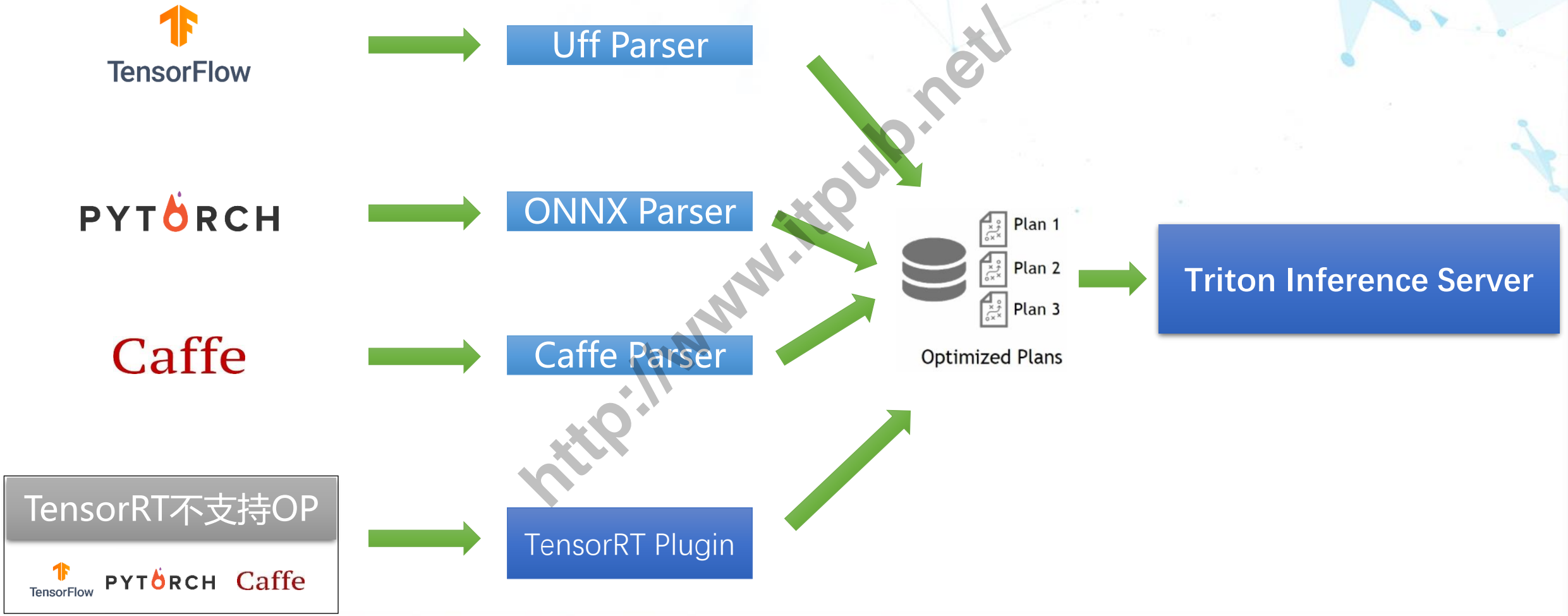
PYTORCH

Caffe

TensorRT不支持OP

TensorFlow

TensorRT+TIS应用



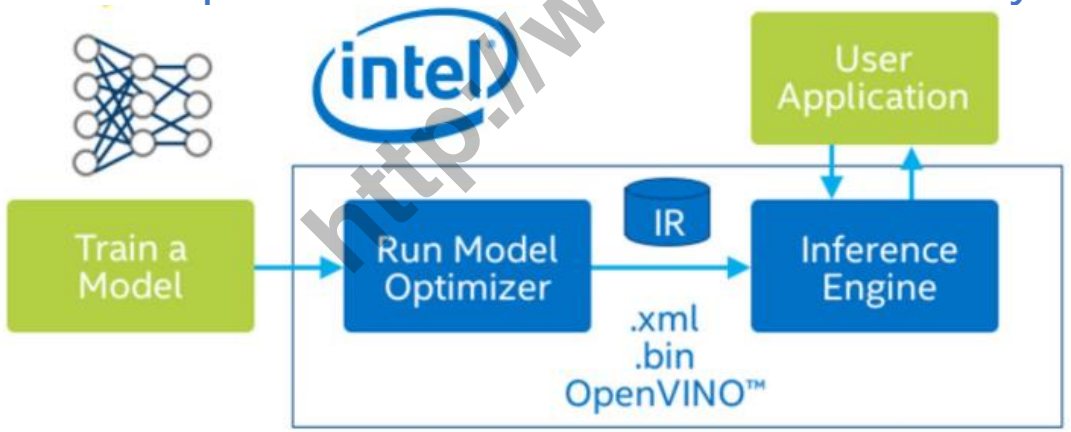
CPU上推理性能优化

前期：编译MKL-DNN优化的TensorFlow-Serving加速TF模型CPU上推理

Intel® MKL

TensorFlow

后期：应用OpenVINO加速TensorFlow、Caffe、PyTorch模型推理

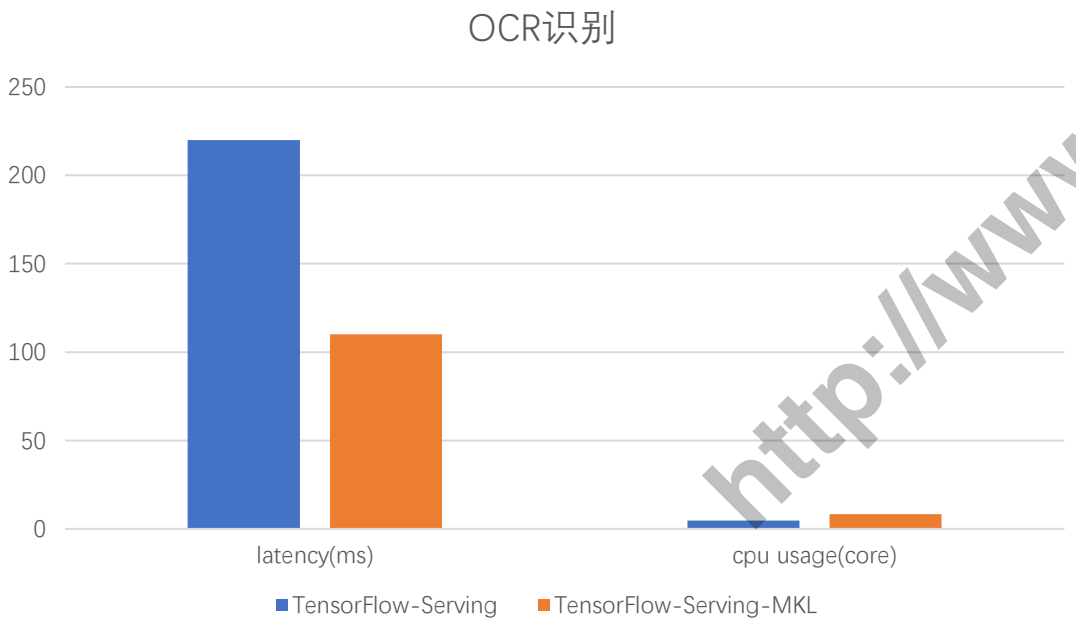


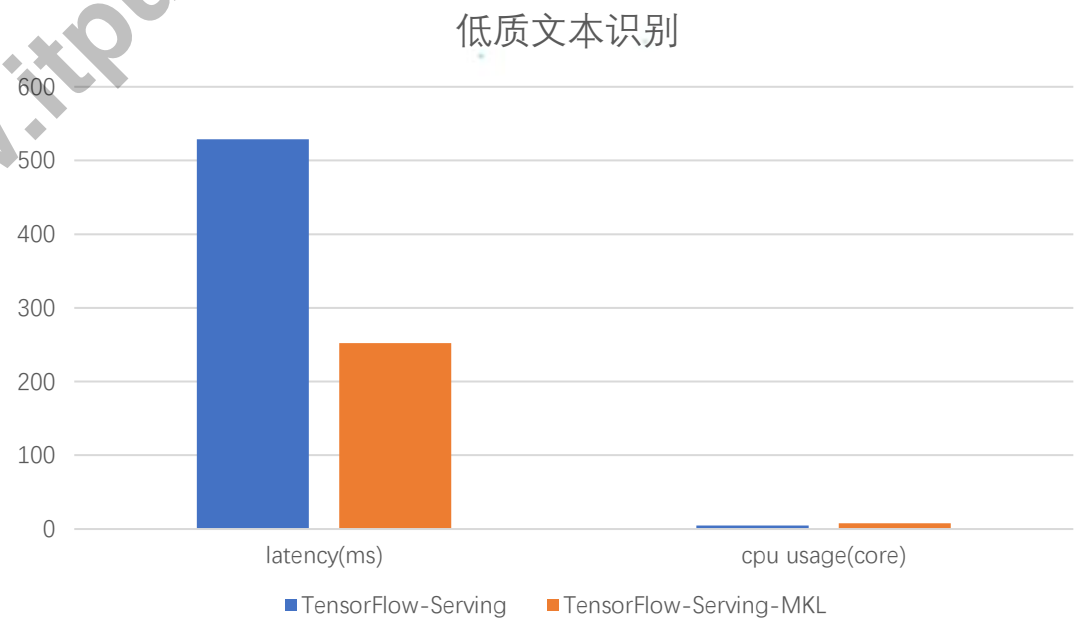
Intel MKL-DNN库应用

- Intel Math Kernel Library for Deep Neural Networks (Intel MKL-DNN).

Intel MKL-DNN库应用

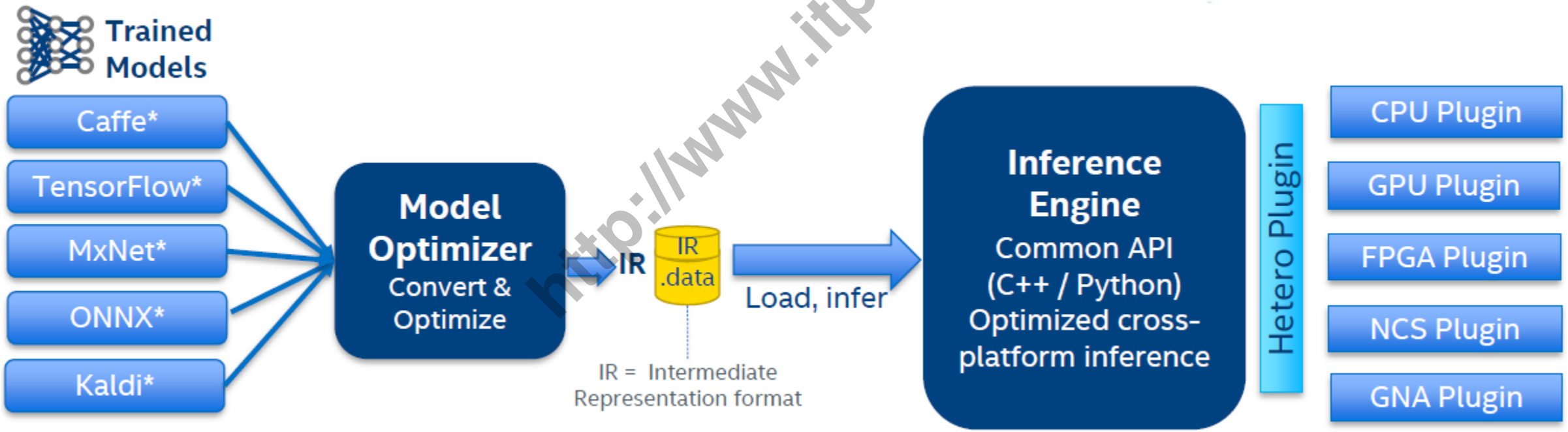
- Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
- 耗时降低一半，CPU资源占用增加70%





OpenVINO介绍

- OpenVINO ToolKit是英特尔发布的一套深度学习推断引擎，支持多种网络框架.
- **Deep Learning Deployment Toolkit** with **CPU, GPU & Heterogeneous** plugins- https://github.com/openvinotoolkit/openvino
- **Open Model Zoo** -Includes pre-trained models, model downloader, demos and samples - https://github.com/openvinotoolkit/open_model_zoo



OpenVINO应用

TensorFlow/PyTorch/Caffe
Trained Models

RPC Server

InitContainer

OpenVINO Model
Optimizer镜像

模型转化

模型优化

Container

OpenVINO Model
Server镜像

模型

IR data（xml +bin）存储于emptyDir

gRPC/TensorFlow-Serving
Client

序列化：
TensorProto.Builder.setTensorContent()

目录

- 58同城AI算法平台演进介绍
- 大规模分布式机器学习
- 深度学习平台整体架构
- GPU/CPU上推理性能优化
- **GPU资源调度优化**

背景

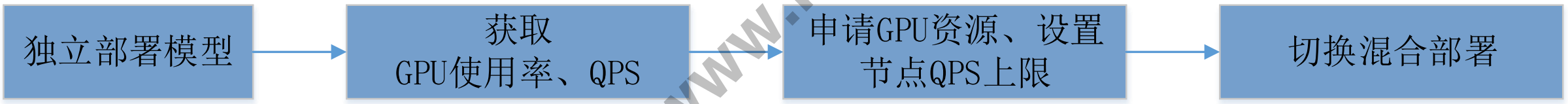
- 开发实验环境分配整张GPU卡比较浪费
- 小流量推理模型GPU利用率低
- 部分模型GPU占用有限，不能打满整张GPU卡
- Kubernetes对GPU卡只能按整数进行调度

TF多模型混合部署

- 场景：TF模型推理需要GPU而流量低，部署一张卡GPU使用率低
- 思路：基于TF-Serving的多模型部署支持，利用k8s实现部署资源调度
- 关键点：模型部署时怎样不影响其他模型？
  模型流量上涨GPU占用增加影响其他模型如何解决？
  独立部署和混合部署怎么快速切换？

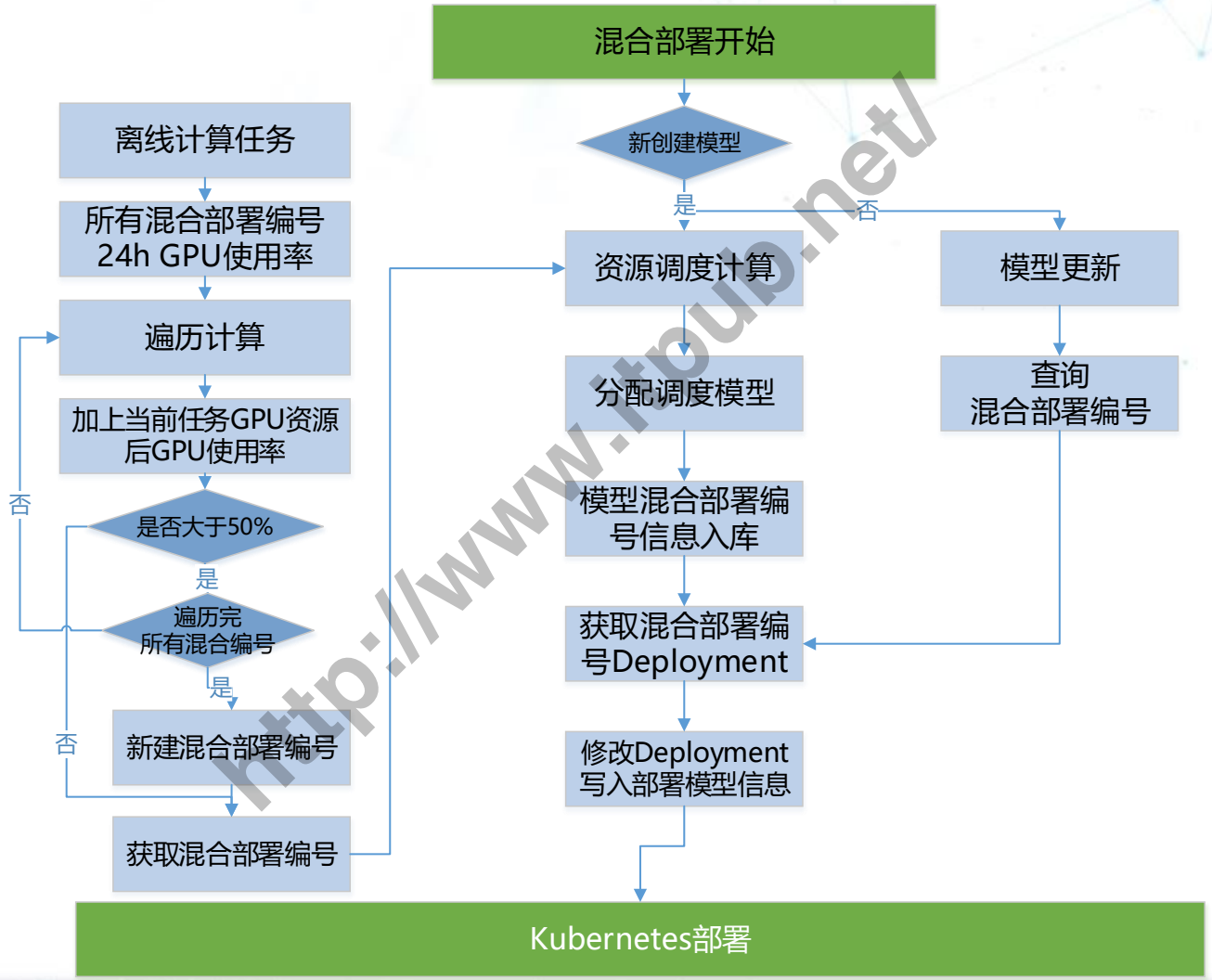
TF多模型混合部署实现

混合部署操作流程



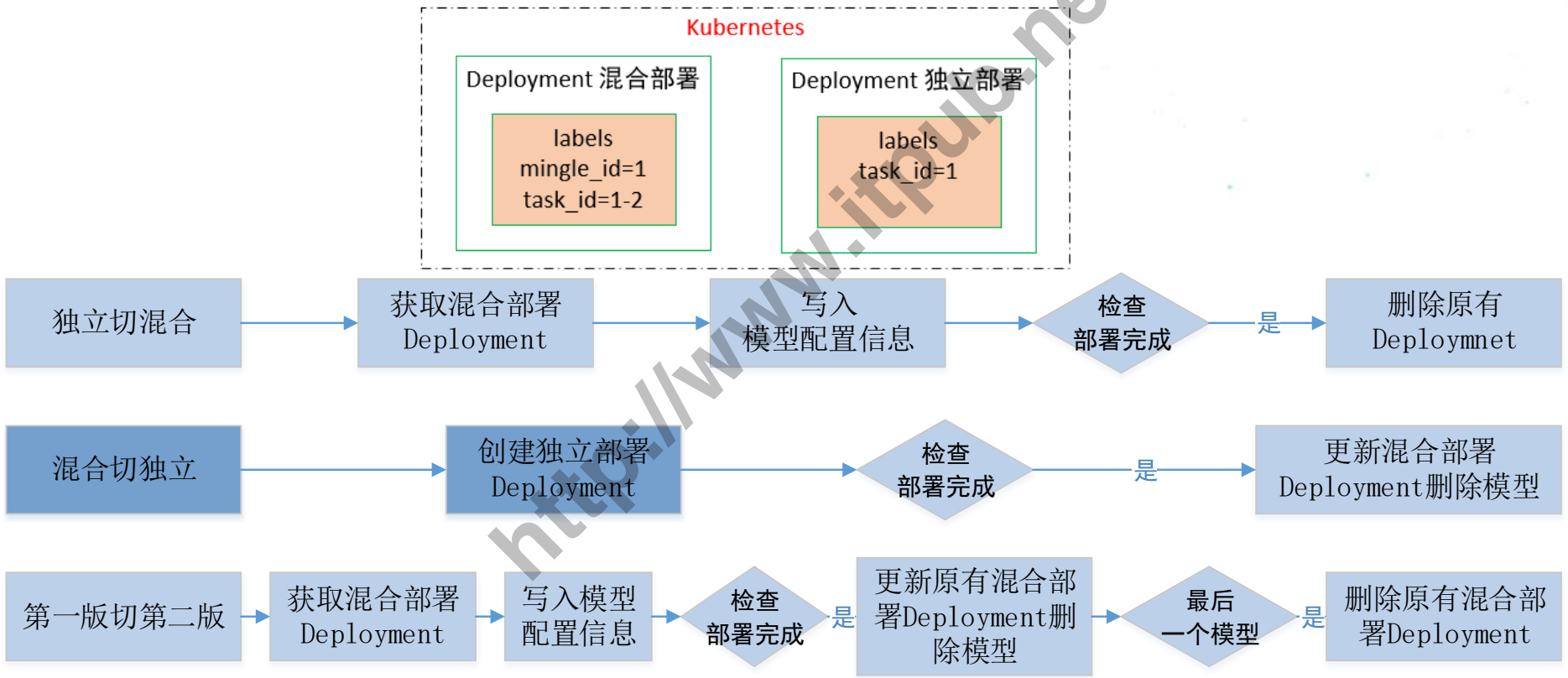
TF多模型混合部署实现

资源统一调度



TF多模型混合部署实现

混合部署 <-> 独立部署 切换



GPU虚拟化应用背景

Kubernetes GPU只能按整数调度

```
resources:
  limits:
    cpu: "6"
    memory: 10Gi
    nvidia.com/gpu: "1"
  requests:
    cpu: "6"
    memory: 10Gi
```

TensorFlow

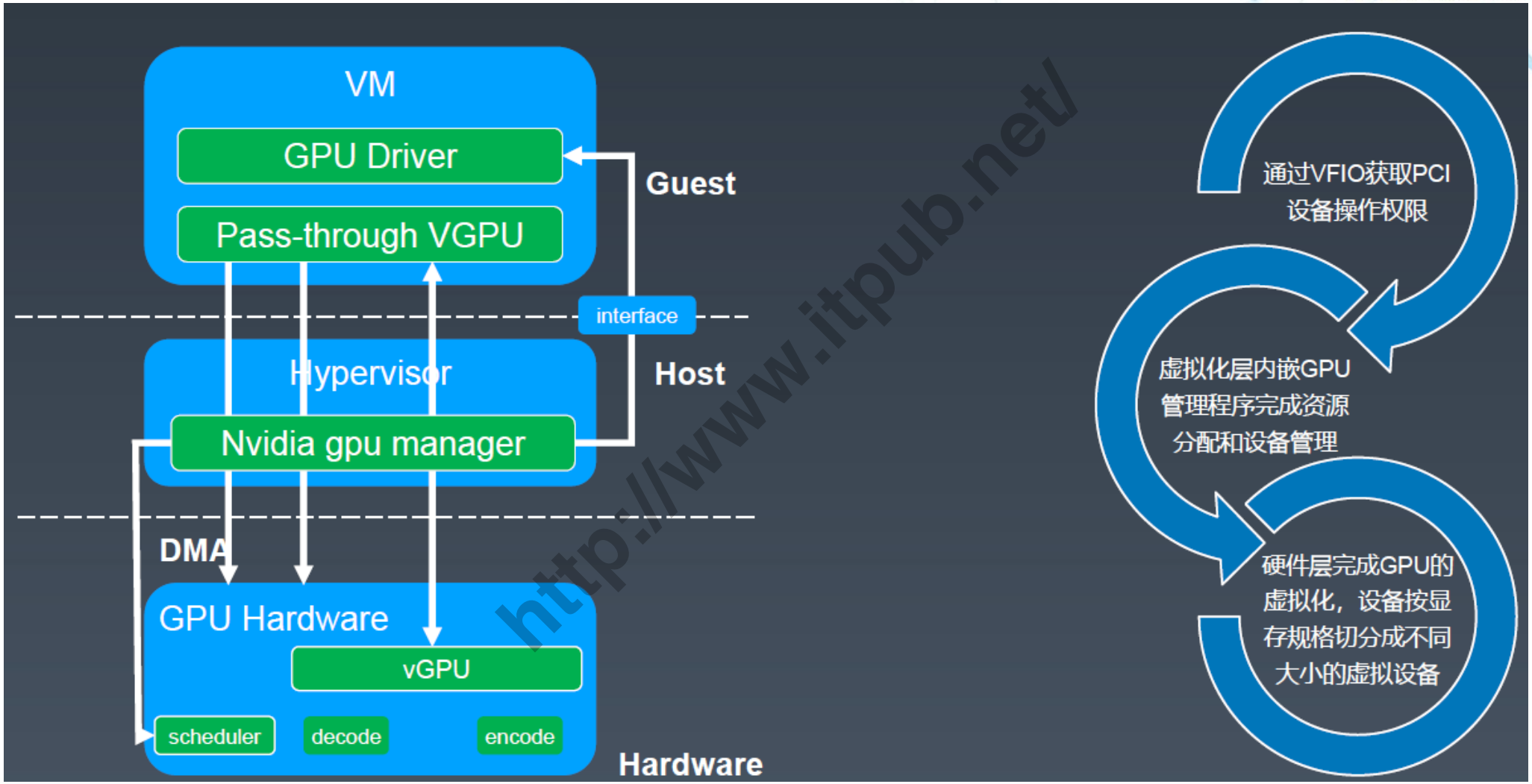
TensorFlow模型混合部署

用户自定义模型

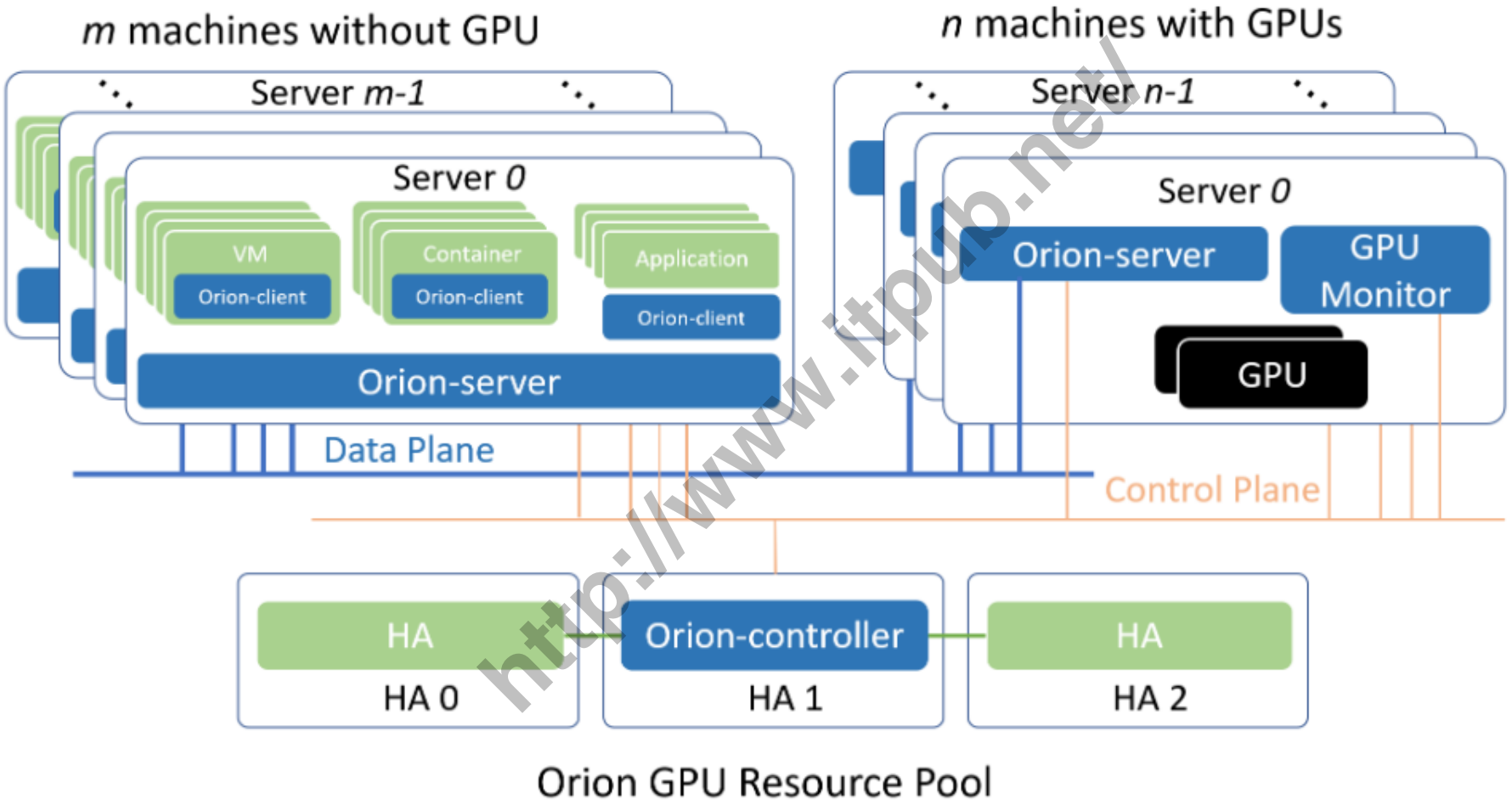
Caffe

PYTORCH

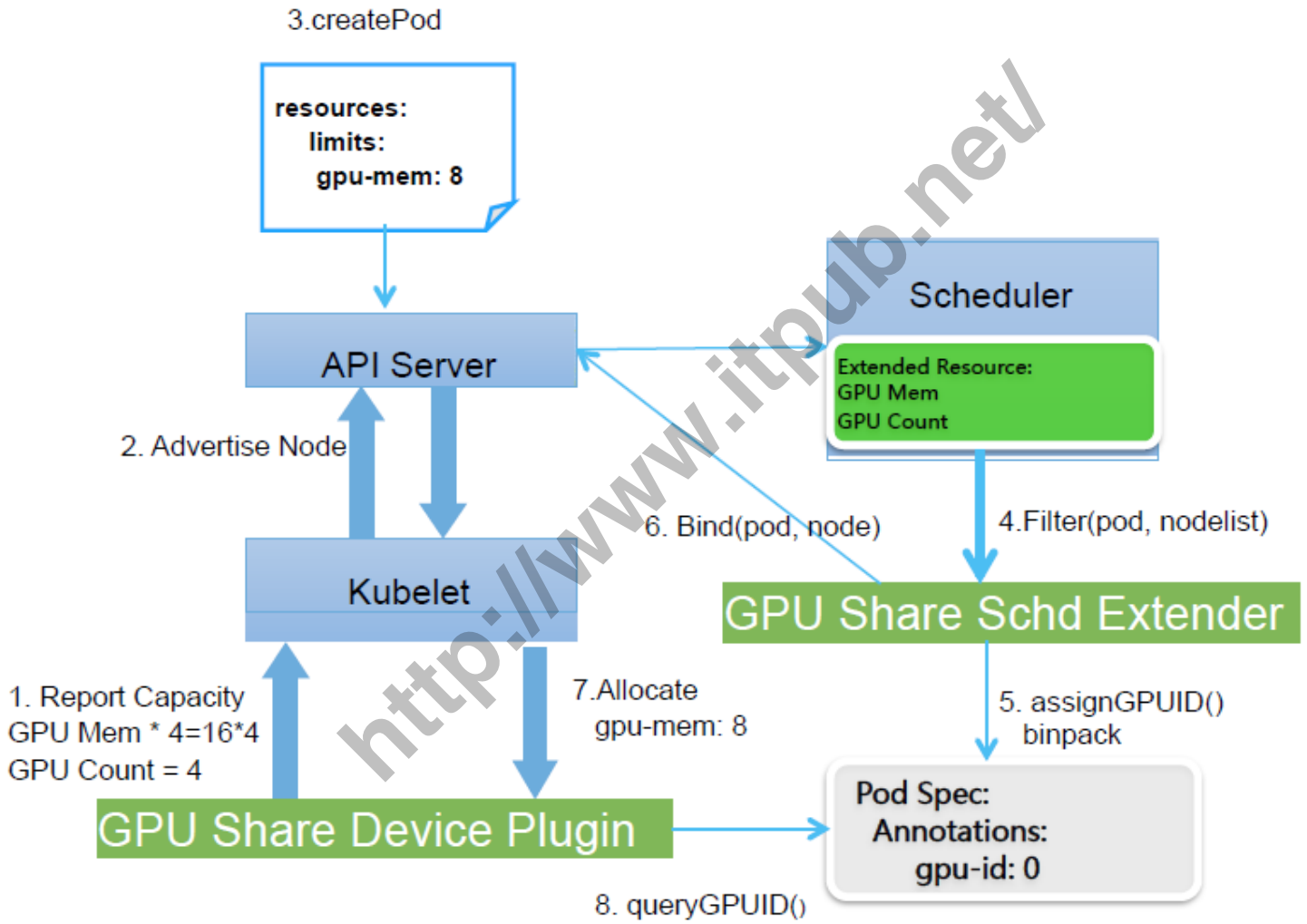
GPU虚拟化技术-Nvidia vGPU



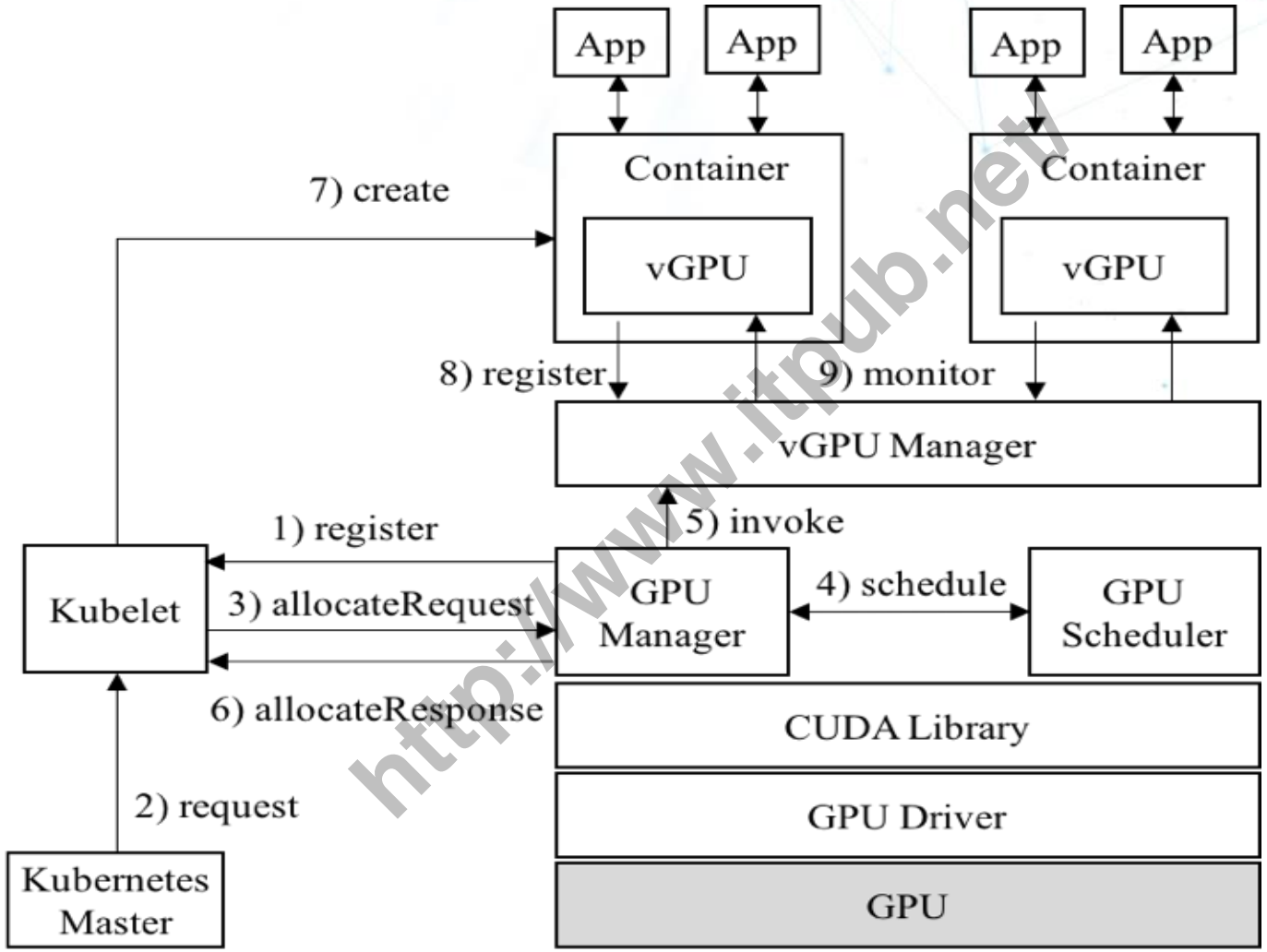
GPU虚拟化技术-OrionX



GPU虚拟化技术-GPU Sharing



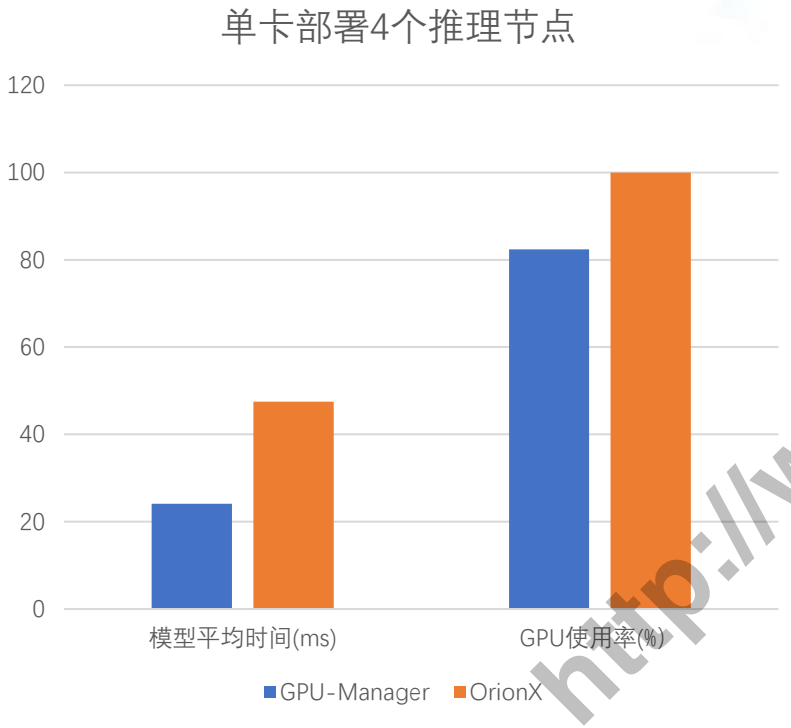
GPU虚拟化技术-GPU Manager

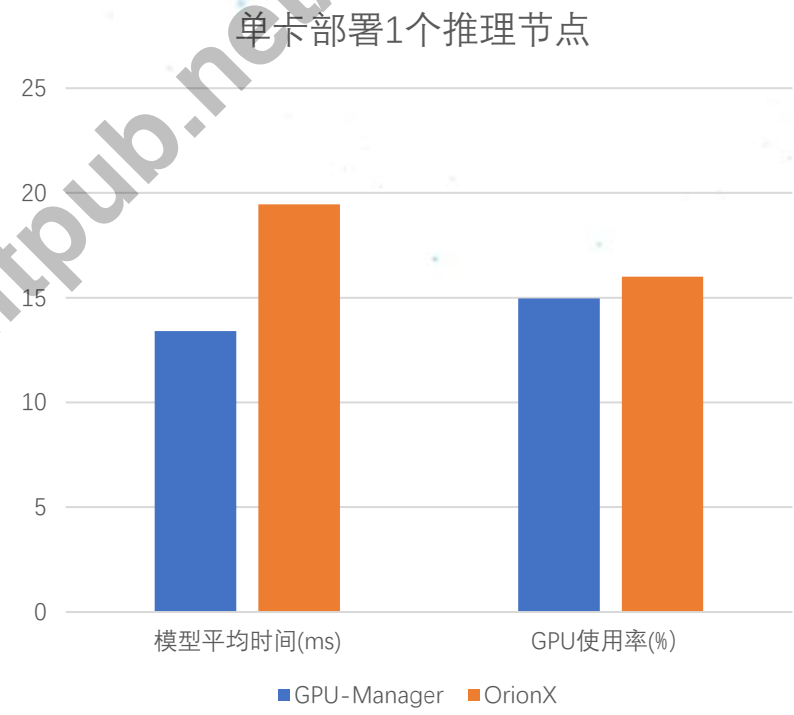


GPU虚拟化技术

虚拟化方案	优点	缺点
Nvidia vComputeServer	GPU硬件兼容更好	收费较高，使用复杂
OrionX	提供资源监控，可以基于算力和显存分配资源	性能损失较大
GPU Sharing	开源，使用简便	仅基于显存分配
GPU Manager	开源，可以基于算力和显存分配资源，使用简便	没有资源监控

GPU虚拟化技术





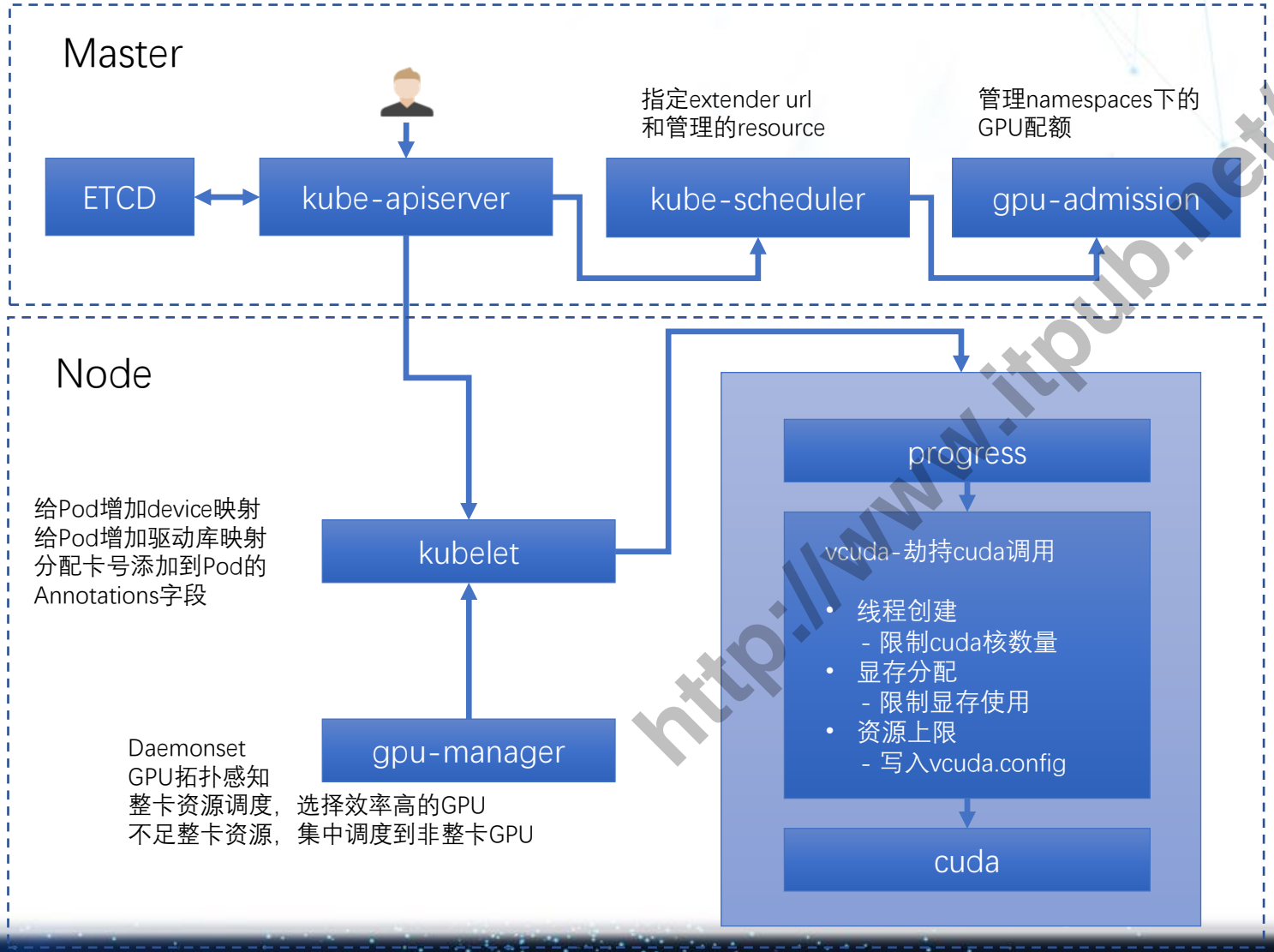
GPU虚拟化应用—方案选择

- 腾讯云开源的容器层GPU虚拟化方案 GPU Manager

- github：
- https://github.com/tkestack/gpu-manager
- https://github.com/tkestack/gpu-admission
- https://github.com/tkestack/vcuda-controller

- paper：
  https://ieeexplore.ieee.org/abstract/document/8672318

GPU虚拟化应用




```
resources:
  limits:
    cpu: "4"
    memory: 20Gi
    tencent.com/vcuda-core: "20"
    tencent.com/vcuda-memory: "11"
  requests:
    cpu: "4"
    memory: 20Gi
    tencent.com/vcuda-core: "20"
    tencent.com/vcuda-memory: "11"
```

```
apiVersion: v1
kind: Pod
metadata:
  annotations:
    tencent.com/gpu-assigned: "true"
    tencent.com/predicate-gpu-idx-0: "1"
```

```
    resources:
      limits:
        cpu: "4"
        memory: 20Gi
        tencent.com/vcuda-core: "20"
        tencent.com/vcuda-memory: "11"
      requests:
        cpu: "4"
        memory: 20Gi
        tencent.com/vcuda-core: "20"
        tencent.com/vcuda-memory: "11"
```

优化效果

小流量模型混合部署

GPU虚拟化

推理GPU卡占用
减少40%

在用卡GPU使用率
提升150%

欢迎推荐人才

- **后端开发工程师**
- **NLP算法工程师**
- **语音识别算法工程师**

58AILab小秘书

北京 朝阳

扫一扫上面的二维码图案，加我微信

邮箱：chenxingzhen@58.com

岗位介绍：https://mp.weixin.qq.com/s/idqOKY0uPs0pxcn0S-YIdg

欢迎关注

欢迎关注58AI Lab公众号

开源项目：通用深度学习推理服务

https://github.com/wuba/dl_inference

开源项目：基于深度学习的问答匹配工具

https://github.com/wuba/qa_match

相关文章：

直播回放 | 通用深度学习推理服务dl_inference开源项目解析

开源丨dl_inference：通用深度学习推理服务

开源丨qa_match：一款基于深度学习的层级问答匹配工具

如何提高AI算法研发效率？58是这样解决的

58深度学习在线预测服务的设计与实现

58人工智能平台WPAI设计与实现

如何利用TensorRT加速GPU上深度学习模型推理

欢迎关注

欢迎关注58技术公众号

Thanks