

PCG AI中台和推荐系统介绍

潘欣

腾讯平台与内容事业群(PCG) AI中台技术负责人

大纲

- Venus机器学习平台
- 无量推荐系统

开源同类产品简介

- TFX

- 谷歌内部广泛使用的工业级机器学习套件 (ExampleGen, Transform, Trainer, Evaluator, Serving, ...)
- 比较耦合TF计算框架 (数据格式, 计算框架, API设计等方面)
- 平台无关, 可以在谷歌内部, 公有云, 私有集群上使用

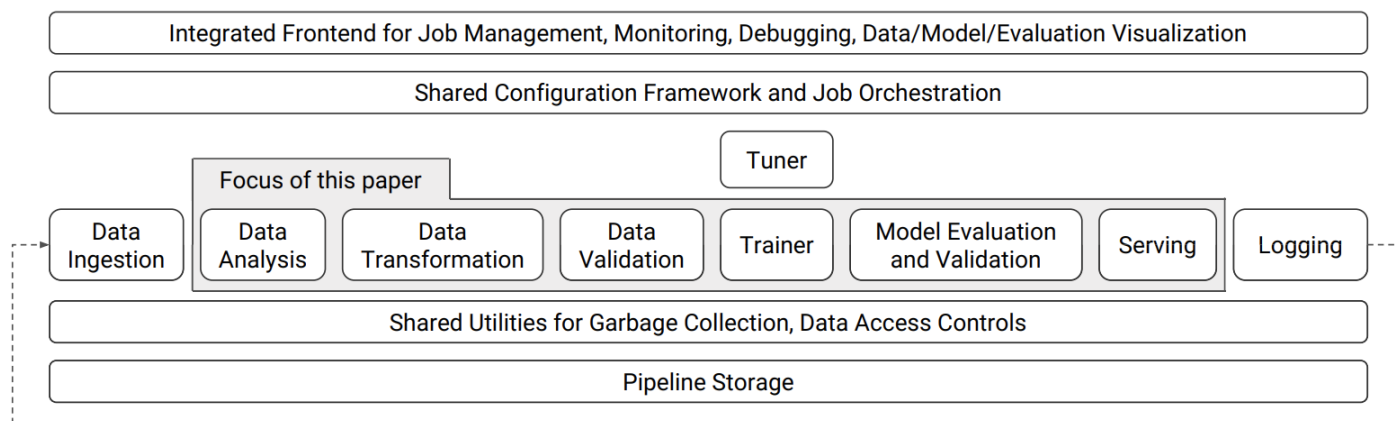


Figure 1: High-level component overview of a machine learning platform.

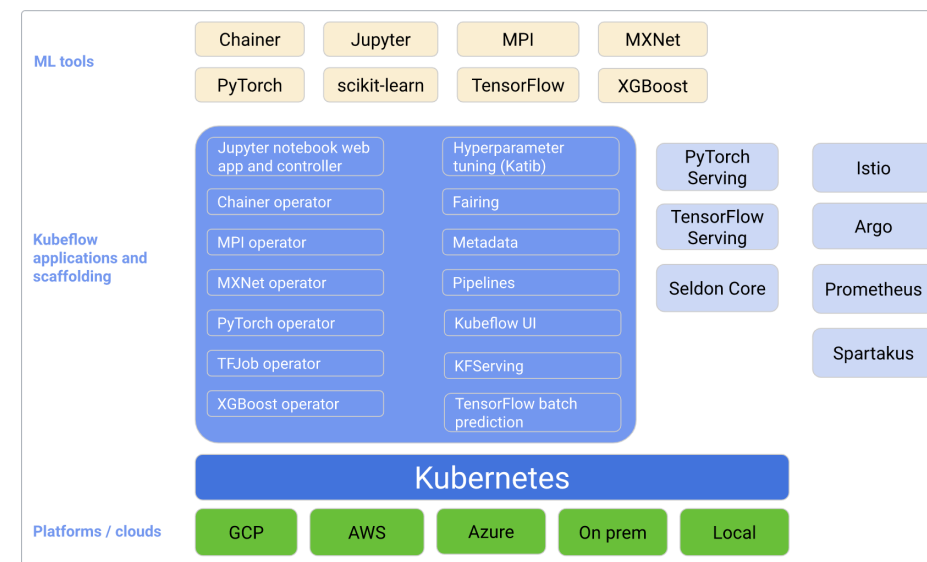
TFX: A TensorFlow-Based Production-Scale Machine Learning Platform, KDD 2017

开源同类产品简介

• KubeFlow

- 深度绑定K8S的机器学习平台。
- 常见的云平台，如GKE, 腾讯tke, PCG 123平台底层都用了K8S。未来大部分计算，存储，服务上K8S恐怕是一个必然趋势。
- 通过op的形式，容易自定义支持不同的机器学习组件。
 - op通常就是镜像。op执行时就是容器的运行。op config就是YAML。
 - Python SDK编写的kubeflow pipeline转YAML

```
@dsl.pipeline(  
    name='foo',  
    description='hello world')  
def foo_pipeline(tag: str, pull_image_policy: str):  
  
    # any attributes can be parameterized (both serialized string or actual PipelineParam)  
    op = dsl.ContainerOp(name='foo',  
        image='busybox:%s' % tag,  
        # pass in init_container list  
        init_containers=[dsl.InitContainer('print', 'busybox:latest', comma  
        # pass in sidecars list  
        sidecars=[dsl.Sidecar('print', 'busybox:latest', command='echo "hel  
        # pass in k8s container kwargs  
        container_kwargs={'env': [V1EnvVar('foo', 'bar')]}},  
    )  
  
    # set 'imagePullPolicy' property for 'container' with 'PipelineParam'  
    op.container.set_pull_image_policy(pull_image_policy)
```



Venus

- Venus简介
 - PCG的大数据平台。主要负责数据的管理，任务Workflow的配置，计算任务的调度，快速在线编码调试。
 - 提供可扩展的自定义插件和市场。
 - **UI画布拖拽**和**可编程WorkFlow**（内测中）两种入口。
- Venus机器学习平台
 - 构建在大数据平台上的机器学习平台。
 - 特征数据计算平台，模型训练和管理，自动化上线和部署等模块。

Venus

- 基于画布

VENUS

配置管理

实例运维

项目

数据

资源

训练管理

在线编码

系统管理

请输入搜索内容

10+

项目概览

无量2.0全流程Demo

在当 workflows 中搜索作业

工具箱

画布刷新

调度重载

保存

运行

撤销

重做

全屏

打印

删除

30 10 ***

AT_n_md_te...ous_demo_dnn

AT_n_md_te..._dnn_monitor

无量2.0 (作业ID: 814988366)

作业名称

AT_n_md_test_numerous_demo_dnn

作业名称可以使用

程序参数

版本

dailybuild(不可用于生产)

运行文件或包

numerous_demo_dnn.py

上传文件

编辑

真正运行的python文件或tar包(最大200MB)

执行程序名称

numerous_demo_dnn.py

用作启动运行程序的入口文件名

执行程序参数

Venus

- 基于可编程Workflow

```
def _create_pipeline():
    examples = external_input(csv_input_location)
    example_gen = CsvExampleGen(input=examples)
    transform = Transform(...)
    ...
    trainer = Trainer(...)
    model_analyzer = Evaluator(...)
    ...
    return pipeline.Pipeline(
        components=[
            example_gen, statistics_gen, infer_schema, validate_stats, transform,
            trainer, model_analyzer, model_validator, pusher
        ])

if __name__ == '__main__':
    pipeline = _create_pipeline(...)
    config = kubeflow_dag_runner.KubeflowDagRunnerConfig(...)
    kfp_runner = kubeflow_dag_runner.KubeflowDagRunner(config, ...)
    kfp_runner.run(pipeline)
```

```
#!/usr/bin/python
# coding: utf-8
from venus.dag import VenusDag
from venus.dag import VenusDagRuntime
...
from venus.op import TimerOperator

VenusDagRuntime.setAppKey("xyzxyzxyzxyzxyzxyzxyz")
dag = VenusDagRuntime.getOrCreateDag()
#支持set模式
timer = TimerOperator()
timer.setCycleFormat("YYYYMMDD")
dag.addOperator(dataCheckJob)
#以及支持字典模式的构造函数
dataCheckJob = StatDataCheckOperator(name="...")
dag.addOperator(dataCheckJob)
hiveStatJob1 = StatHiveOperator(
    "name", cluster="sz", noticeUser="bob", checkNullResult=False, sql="insert ...")
dag.addOperator(hiveStatJob1)
#配置依赖关系 timer -> dataCheckJob -> hiveStatJob1
timer.addDownStream(dataCheckJob)
dataCheckJob.addUpStream(hiveStatJob1)
#保存配置到venus服务端
dag.save()
#dag.save会解析成如下的json
#当然也可以直接run起来
cycle = 20200310
params = {} #params表示动态参数, 如果有的话, 有些组件支持动态参数
VenusDagRuntime.run(dag, cycle, params)
#VenusDagRuntime.run会先调用dag.save, 然后调用一个http的新runcycle接口驱动作业。
```

我的组件

jinlan

cztestcA

cztestZC

jinlan

cztestcA

cztestZC

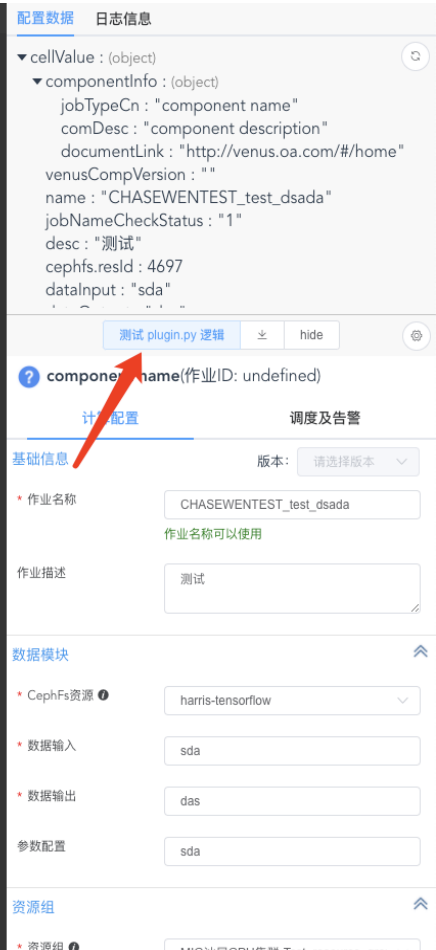
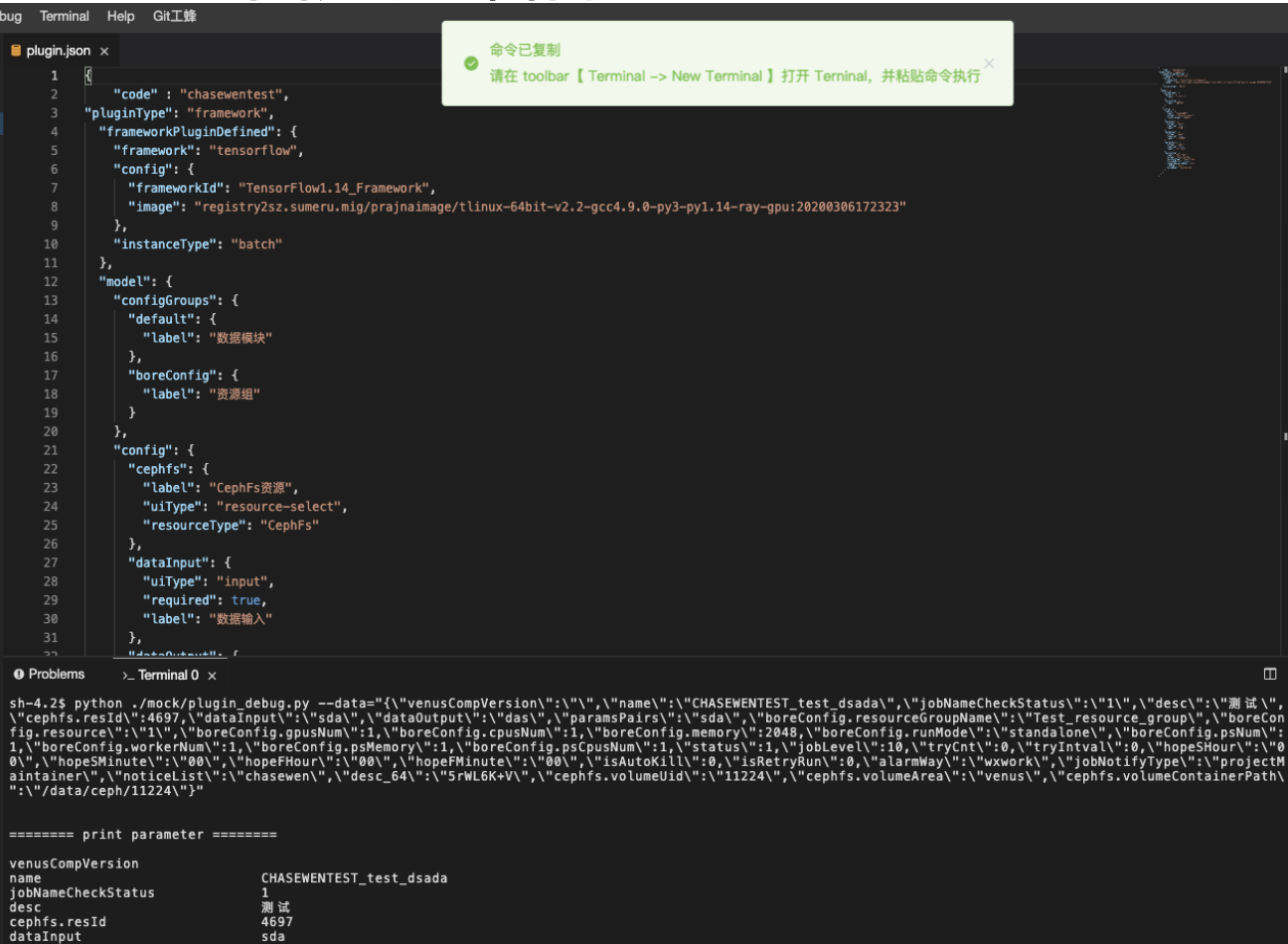
cztestZD

LexLiangDe

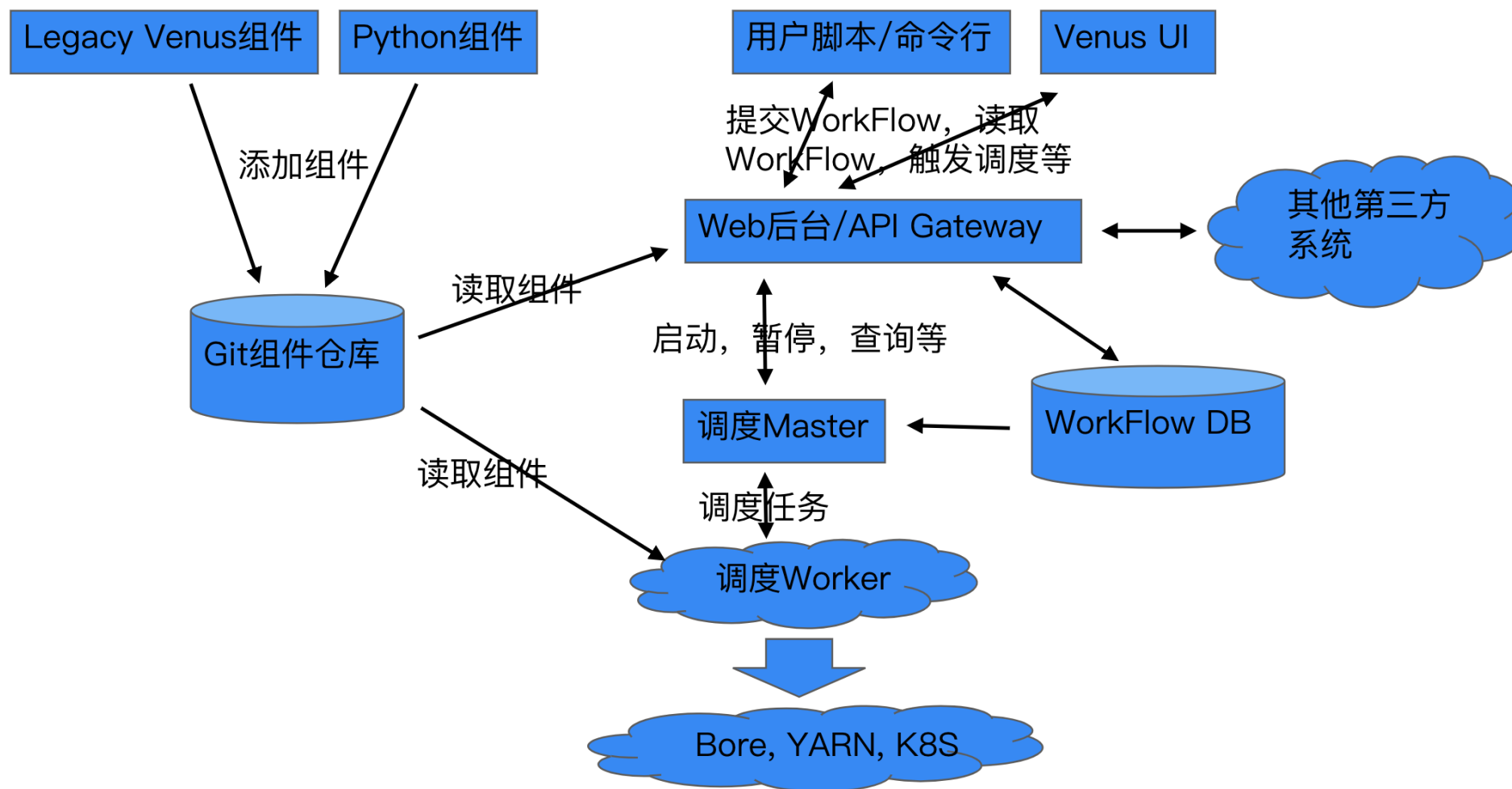
moTo

Venus

• 自定义插件



Venus任务调度架构



无量推荐系统简介

- 承载腾讯平台与内容事业群的推荐场景
 - 腾讯看点（浏览器，QQ，快报，商业化），腾讯新闻，腾讯视频，微视，应用宝，QQ空间，小鹅拼拼等。
 - 服务数亿DAU，每天训练生产数千个模型，千亿级调用。
- 主要包含训练和Serving
 - 基于TensorFlow的定制化深度学习框架。
 - 支持可扩展Servable的Serving框架。

无量推荐系统简介

业务	场景	效果提升数据
短小视频业务	端内精排	单目标（完整度模型）：时长提升5.8%【相对非无量】 多目标（基于单目标融合GRP）：人均时长相对提升10%+，次留相对提升1%，点赞率提升53%，关注率提升35%【相对单目标基线】 多目标（基于MMoE）：时长提升0.66%，次日留存提升0.03%，点赞率提升12.93%，关注率提升5.92%【相对GPR多目标】
	端内粗排	时长提升1.5%(精排条数从3000降到700)
	公众号精排	列表页: 点击渗透率提升13.33, 点击率提升18.47% 推荐页: 人均时长提升40%
图文、视频、信息流	图文	主feeds: 大盘时长提升1.8%, PV提升1.9%
	视频	主feeds: vv提升1.7%,时长提升2.64%
	品类	短内容主推荐: pv提升10%【多目标，相比非无量基线】 小说主推荐: 点击率提升30% 短内容浮层推荐: 时长提升8%, 互动指标提升35%
	增长	用户增长dsp: 点击量提升70%，点击率提升250%，cpc下降25% 某主TL拉新: 下载完成量提升1.54% 某主TL拉新: 下载转化率提升25.67%, 减少35.38%曝光 内渠拉新拉活: 点击率提升10%

业务	进展
长视频	点赞模型提升5%
新闻	底层页: 多目标模型，人均停留时长提升1.7%
社交	热播推荐: 点击率提升5% 浮层推荐: vv提升5%，时长提升8%
APP推荐	发现首页: 分发系数提升6.6%, 广告ARPU提升6.1%

传统推荐架构

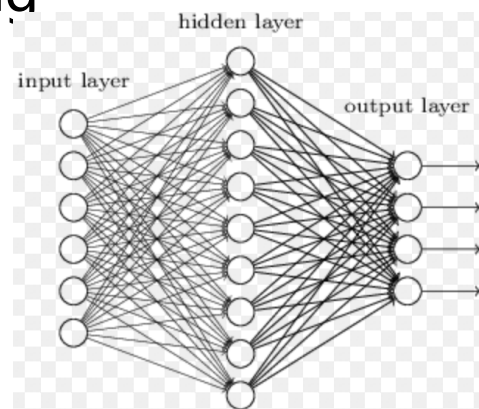
- 稀疏特征与稠密特征

- 稠密特征：图片256x256x3
- 稀疏特征：item id, city, sex
- 交叉稀疏特征：18岁_深圳_男（可达千亿规模）

深圳	北京	广州	上海	东莞	杭州	西安	长沙	...
0	1	0	0	0	0	0	0	0



0.00111	0.0312	-1.0131	1.1223	0.1031	0.12313	-0.1212	0.3132
---------	--------	---------	--------	--------	---------	---------	--------

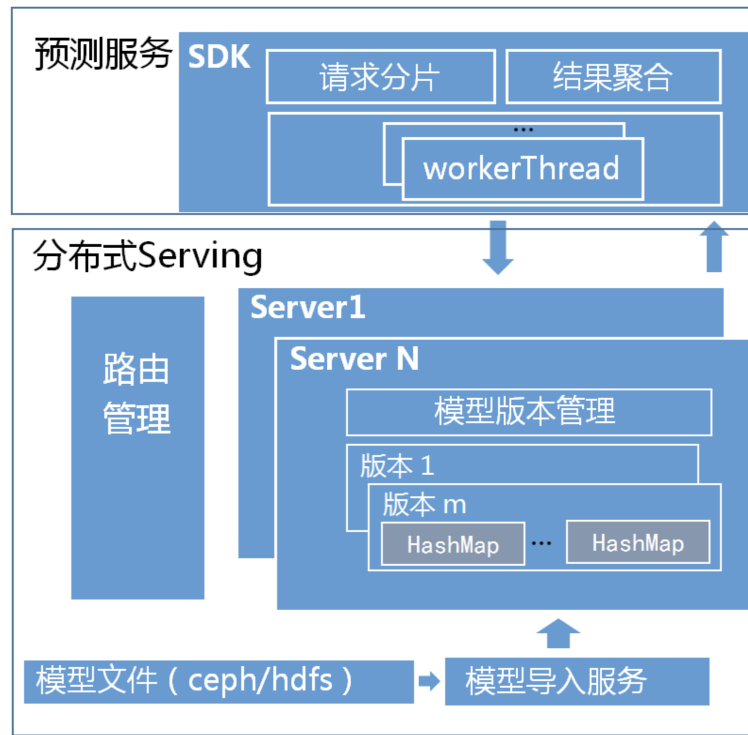
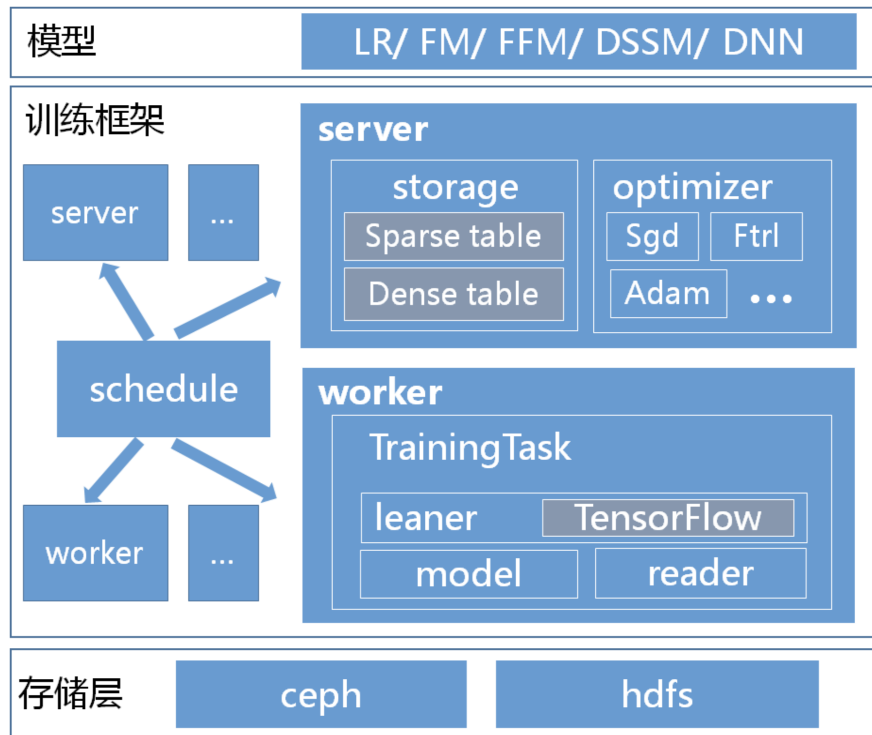


- Embedding

- 用稠密向量来表示稀疏特征。
- 比如“北京”，id: 1111，从hashtable查到对应embedding
- Embedding vector进入DNN训练

传统推荐架构

- Workers+参数服务器 (PS Servers)



LiMu, et.al., Scaling Distributed Machine Learning with the Parameter Server, OSDI 2014

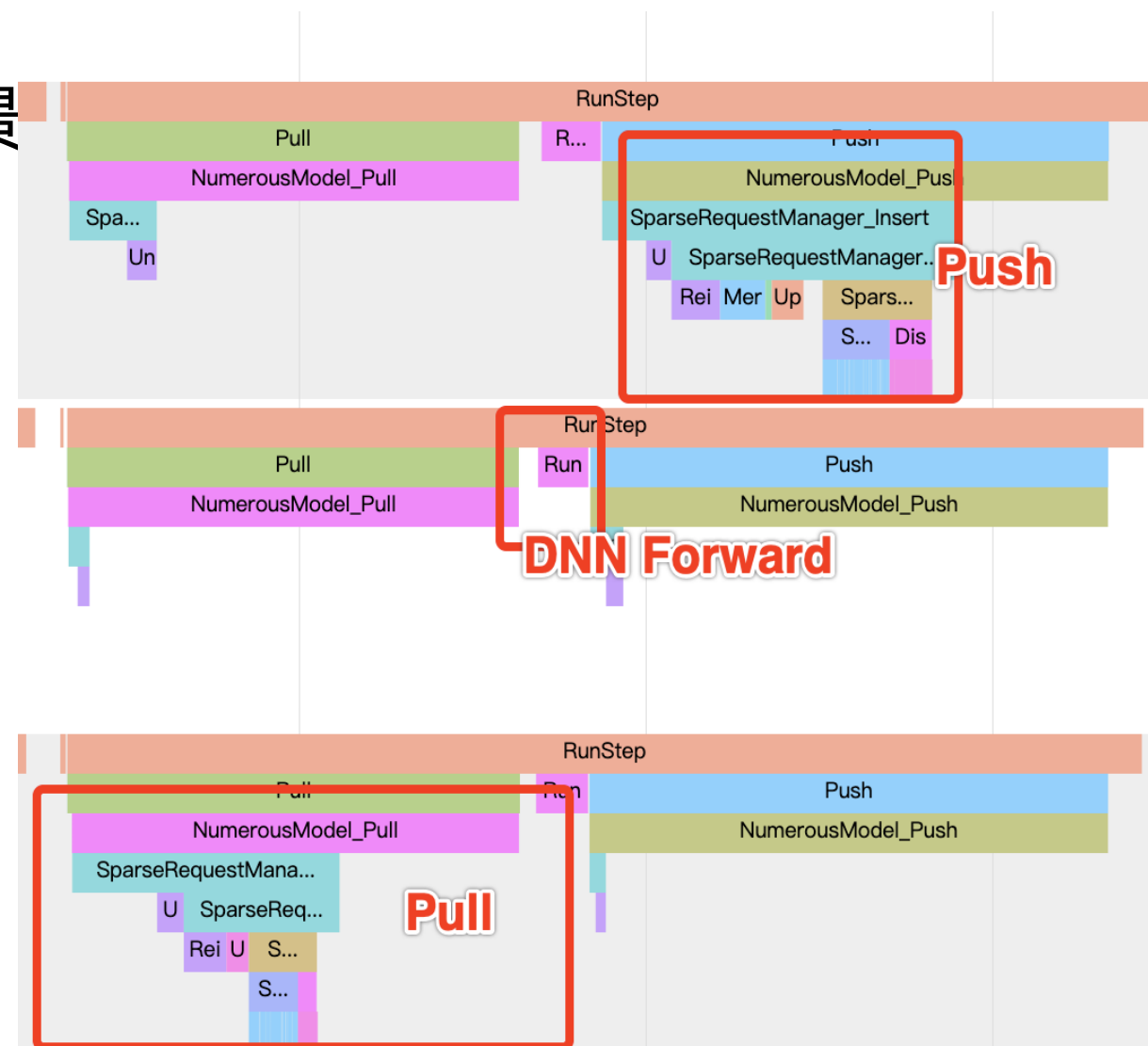
推荐系统后台关键挑战

- 例：某个特征特别多的训练场景

- Pull和Push占用了Worker大多数时间
- DNN Forward其实只占用了一小部分

- 关键挑战

- GB ~ TB级模型，海量Embedding参数
- 持续在线训练，分钟级新模型上线
- 频繁A/B实验，线上服务持续调整
- 模型复杂度，灵活性，计算量逐年提升



无量推荐系统--深度学习框架

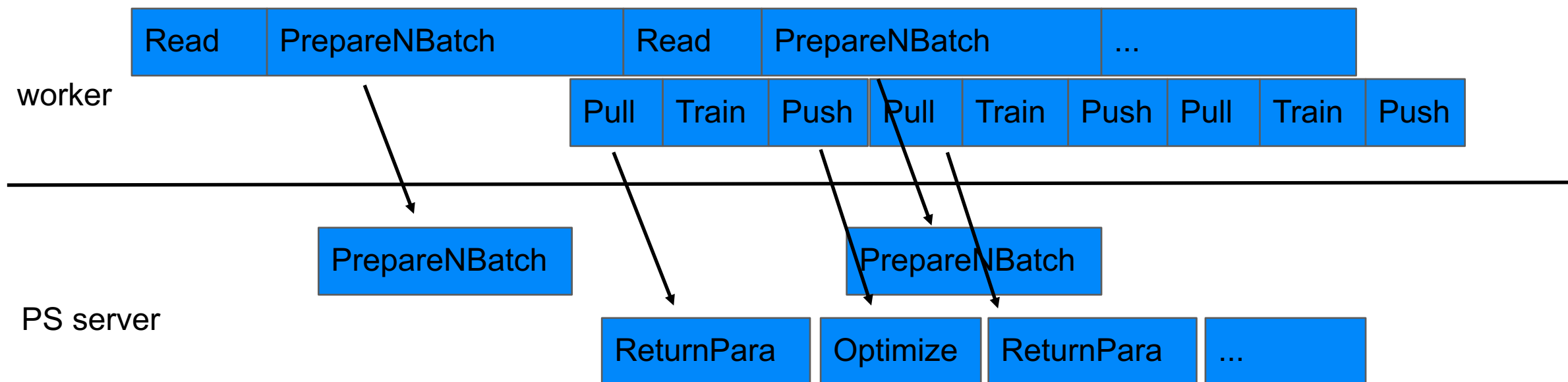
- 基于Pipeline的高性能训练

- 在数据加载时预计算key的PS分配，排序等。
- minibatch pull参数时，在PS缓存minibatch embedding的内存地址。
- minibatch push参数时，只发送gradient，PS基于缓存快速更新，省去key处理和查找。
- ~ 10%性能提升。

- 多minibatch Pipeline训练

- 预计算多个minibatch的key
- out-of-band，通知PS后续N个minibatch的key。预初始化embedding，缓存minibatch
- 基于预取的多级PS存储
- minibatch pull/push时，基于PS缓存地址快速更新

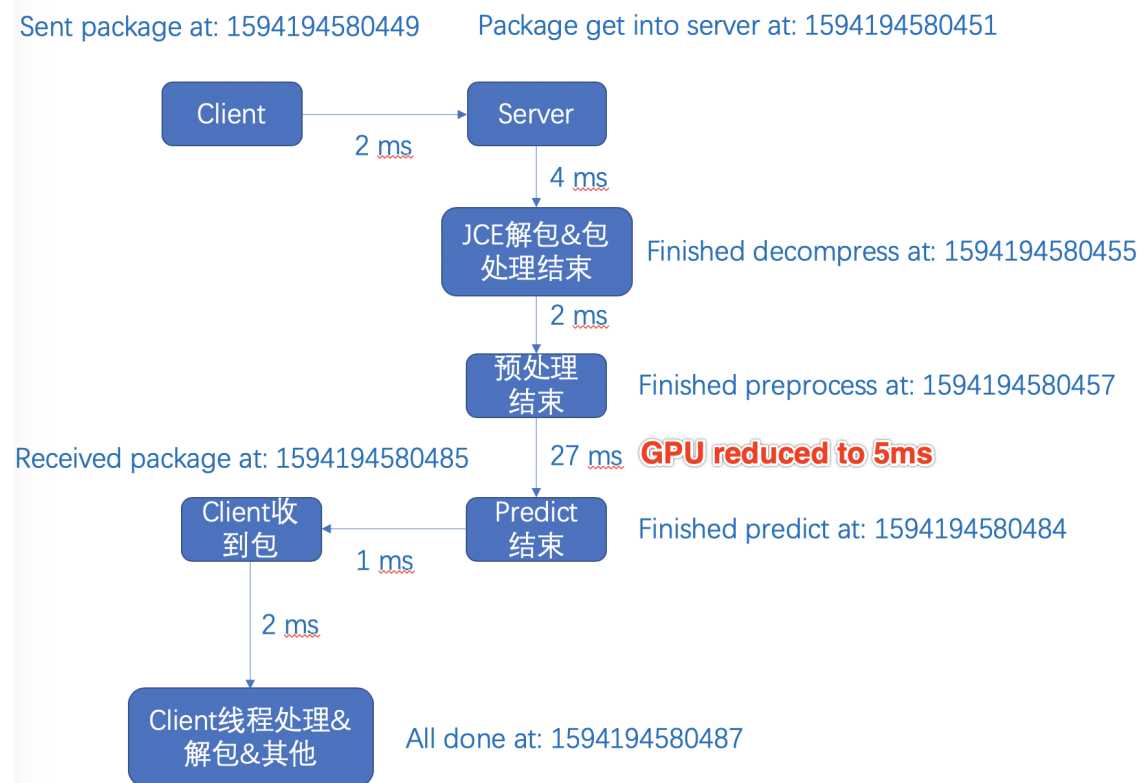
无量推荐系统--深度学习框架



无量推荐系统--深度学习框架

- 基于GPU的Embedding+DNN推理

- 全GPU预测：8~16GB显存容纳4~10GB模型，在某些模型上，Embedding加速约4倍，DNN时间几乎可以忽略。
- 将来DNN变得更复杂，GPU的差异化更明显。



无量推荐系统--深度学习框架

- 基于GPU的Embedding+DNN推理
 - 为了突破显存限制，后续可以实验量化int8，参数稀疏化和淘汰，。
 - 将来32GB显存卡，4倍压缩比达到128GB，淘汰一半参数达到256GB。有一定的探索空间。
 - Intel PMEM单机内存可以达到4TB以上，性能略有下降。
 - 可以消除远端的cache查找，内测中。

无量推荐系统--深度学习框架

- HPC思路，将存储和算力密集的放在高速互联硬件上。

- NVLink (GPUDirect) 替换万兆网卡完成pull/push。
- 最大限度消灭无效计算，最大限度利用GPU并行算力。

- 多级存储

- GPU单机8卡，256GB显存
- Intel PMEM单机可以超过5TB内存（无需SSD）

- Pipeline

- 预计算epoch需要的key。（并行掩盖）
- 加载epoch的所有embedding到显存。（少次批量）
- 在GPU内完成一个epoch的多卡训练。

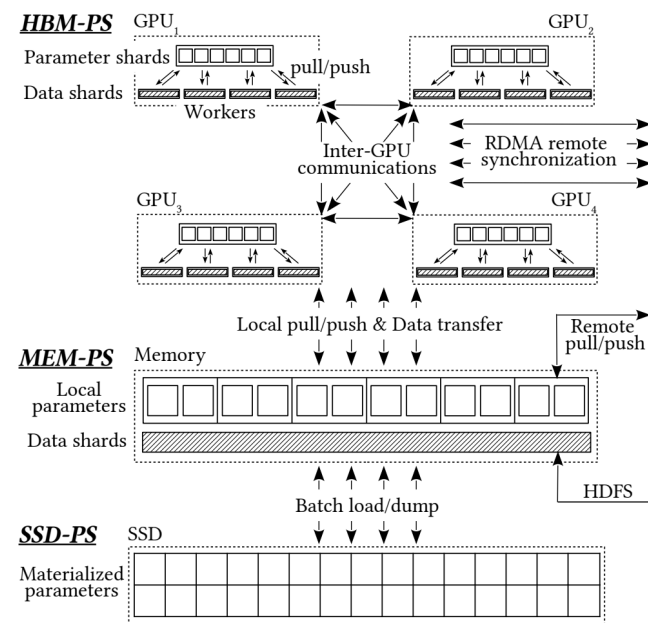
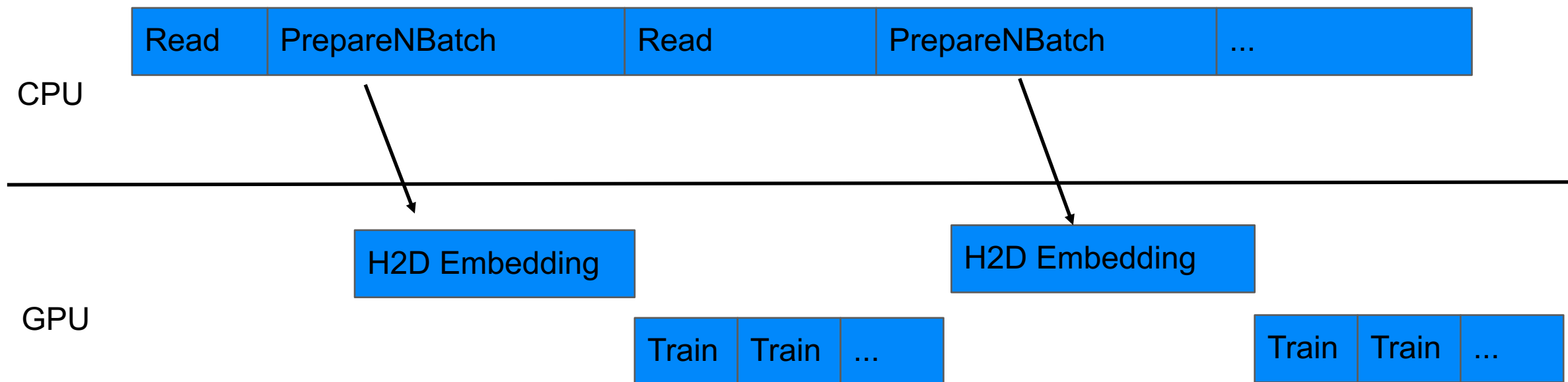


Figure 2. Hierarchical parameter server architecture.

Distributed Hierarchical GPU Parameter Server
for Massive Scale Deep Learning Ads Systems,
MLSys 2020 from Baidu

无量推荐系统--深度学习框架



无量推荐系统--深度学习框架

- 无量Estimator API
 - 保留TensorFlow原生接口的关键能力。
 - 基于原生tf API编写DNN模块。
 - 单步调试, Fetch中间结果, 基于图依赖分析计算部分子图。
 - 可选高层封装, 快速完成常见的train/evaluate能力。
- 自定义Estimator接口, 放弃tf.Estimator和keras.fit的高层封装。
 - tf python实现底层修改过于复杂, 不利于团队快速迭代。
 - tf 内核实现过于复杂, 不利于定制化高性能计算逻辑。

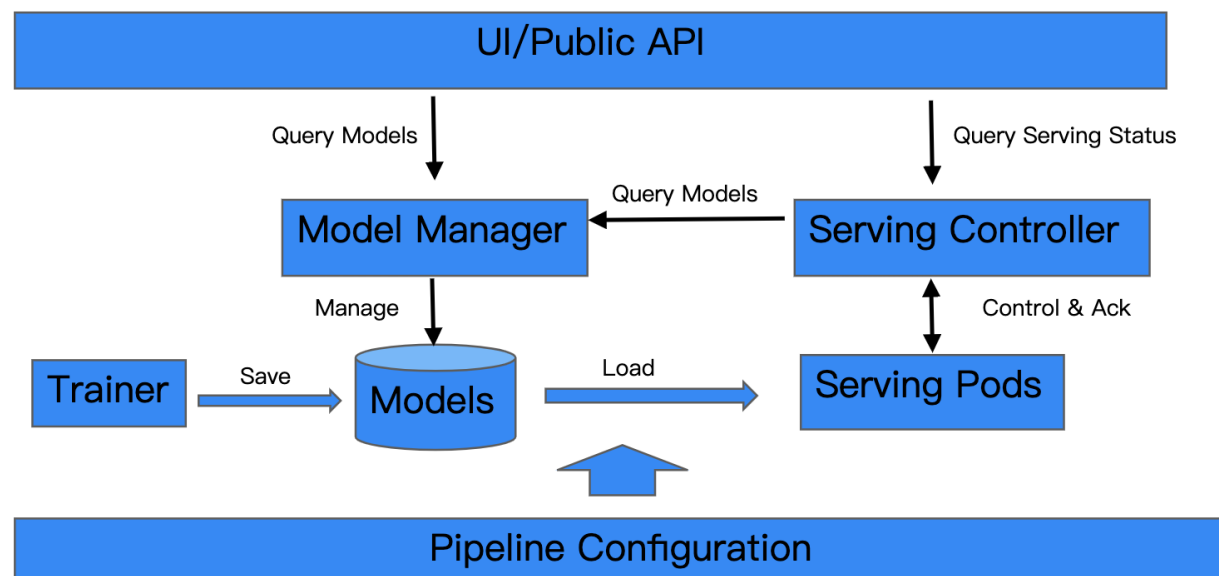
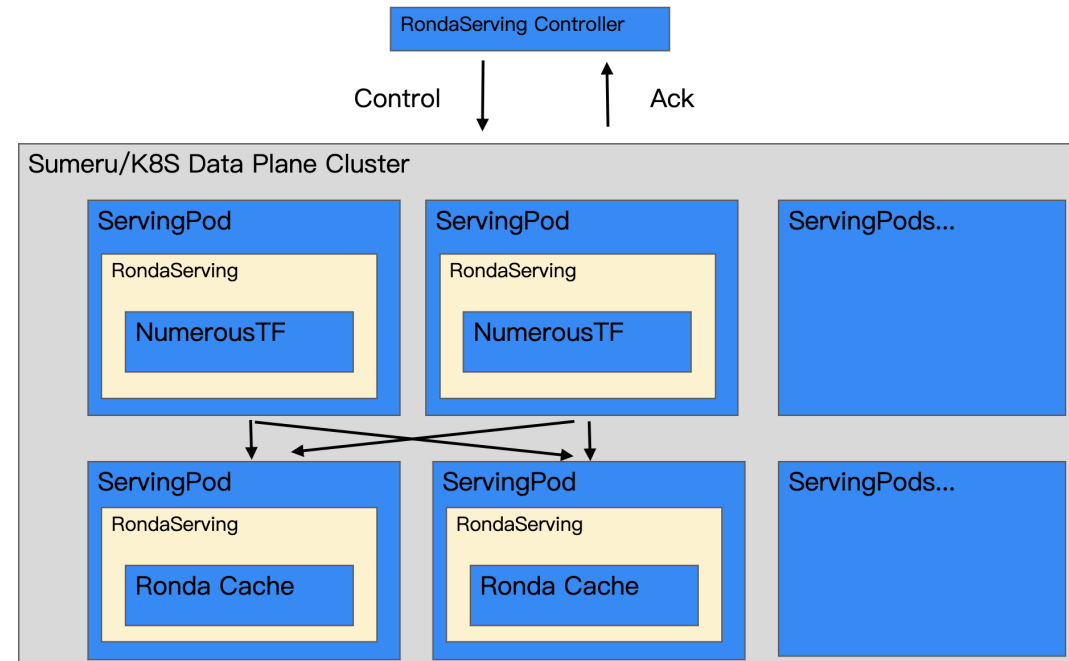
无量推荐系统--深度学习框架

- 无量Estimator API
 - 保留自研PS架构和ring分布式架构，放弃TensorFlow分布式。
 - PS架构：参数变长，淘汰，导出/恢复，统计。计算Pipeline，预存取，预初始化。
 - Ring架构：多级存储，计算Pipeline，批量GPU数据拷贝，HPC化。

```
class Estimator(object):  
    def __init__(self, sess=None):  
        ...  
  
    def train(self, target, steps=None, hooks=[]):  
        """Runs training steps.  
  
        Args:  
            target: numerous.Target instance.  
            steps: Max steps to run. May return before `steps` if data runs out.  
            hooks: A list of SessionRunHook instances used to customize the training logic.  
        """  
  
    def evaluate(self, target=None, steps=None, hooks=[]):  
        """Runs evaluation steps.  
  
        Args:  
            target: numerous.Target instance.  
            steps: max steps to run. May return before `steps` if data runs out.  
            hooks: A list of SessionRunHook instances used to customize the training logic.  
        """
```

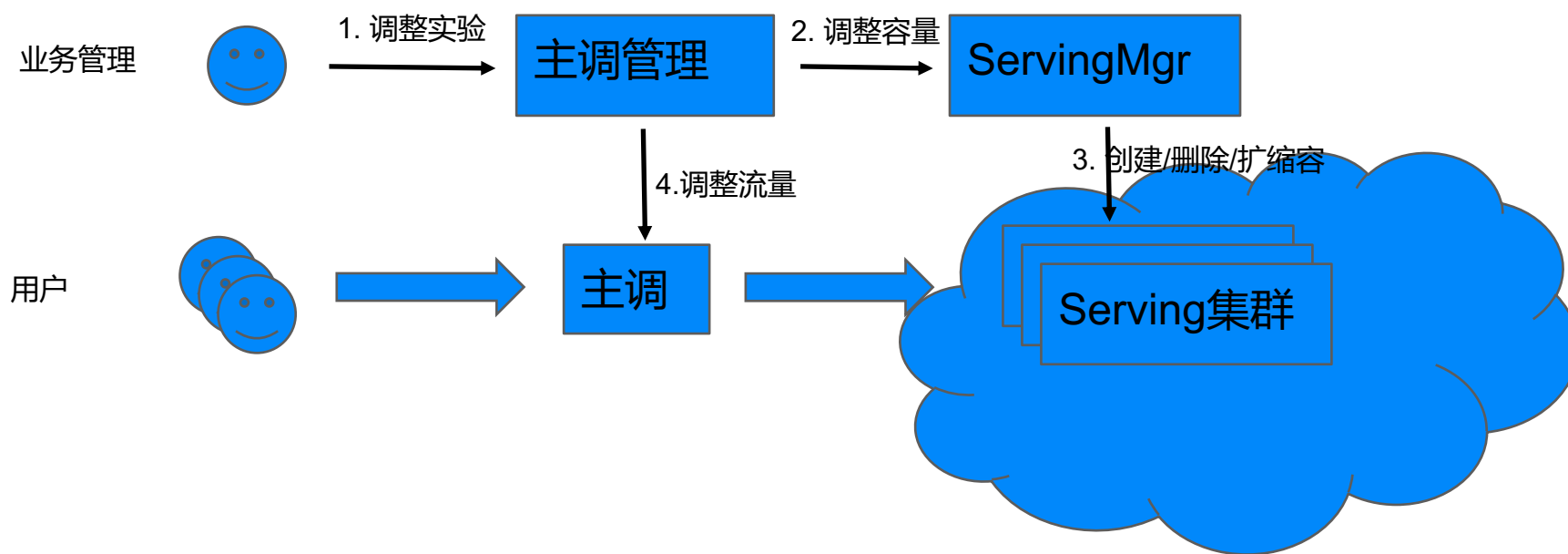
无量推荐系统--Serving

- 模型分片上线案例
 - 训练侧End2End训练，DNN部分通常比较小（MB级），Embedding部分比较大（GB~TB），拆分预测。
 - ServingManager基于上线配置完成服务创建和模型上线。
 1. 调用容器平台完成多个Serving服务创建。e.g. K8S Deployment
 2. 调用Serving服务，完成模型不同模块的加载。
 3. 通知业务完成，开始接受流量。



无量推荐系统--Serving

- A/B实验自动化系统



总结回顾

- Venus

- 插件化，自定义化的组件体系
- 同时支持UI拖拽和可编程Workflow能力

- 无量推荐系统

- 海量embedding，在线训练，AB实验，模型复杂度等带来系统级挑战
- Pipeline优化训练速度
- 基于异构硬件，多级存储新架构
- ServingManager+插件化Servable的Serving系统