



陌陌推荐场景的Rank架构最佳实践

郭沫祎

<http://www.it-ebooks.info>



个人介绍



郭沫祎_gamuguo

中国



扫一扫上面的二维码图案，加我微信

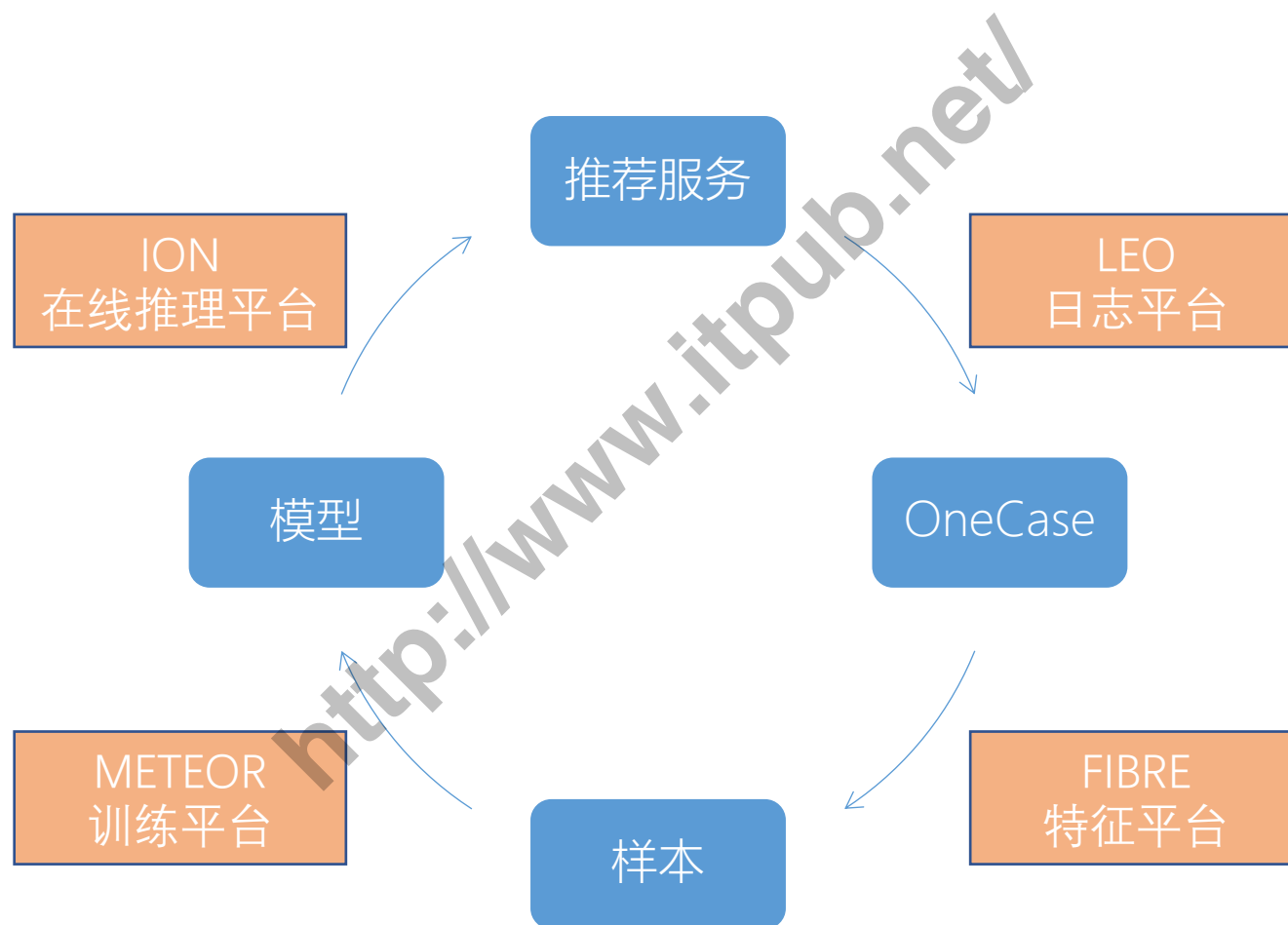


陌陌推荐场景



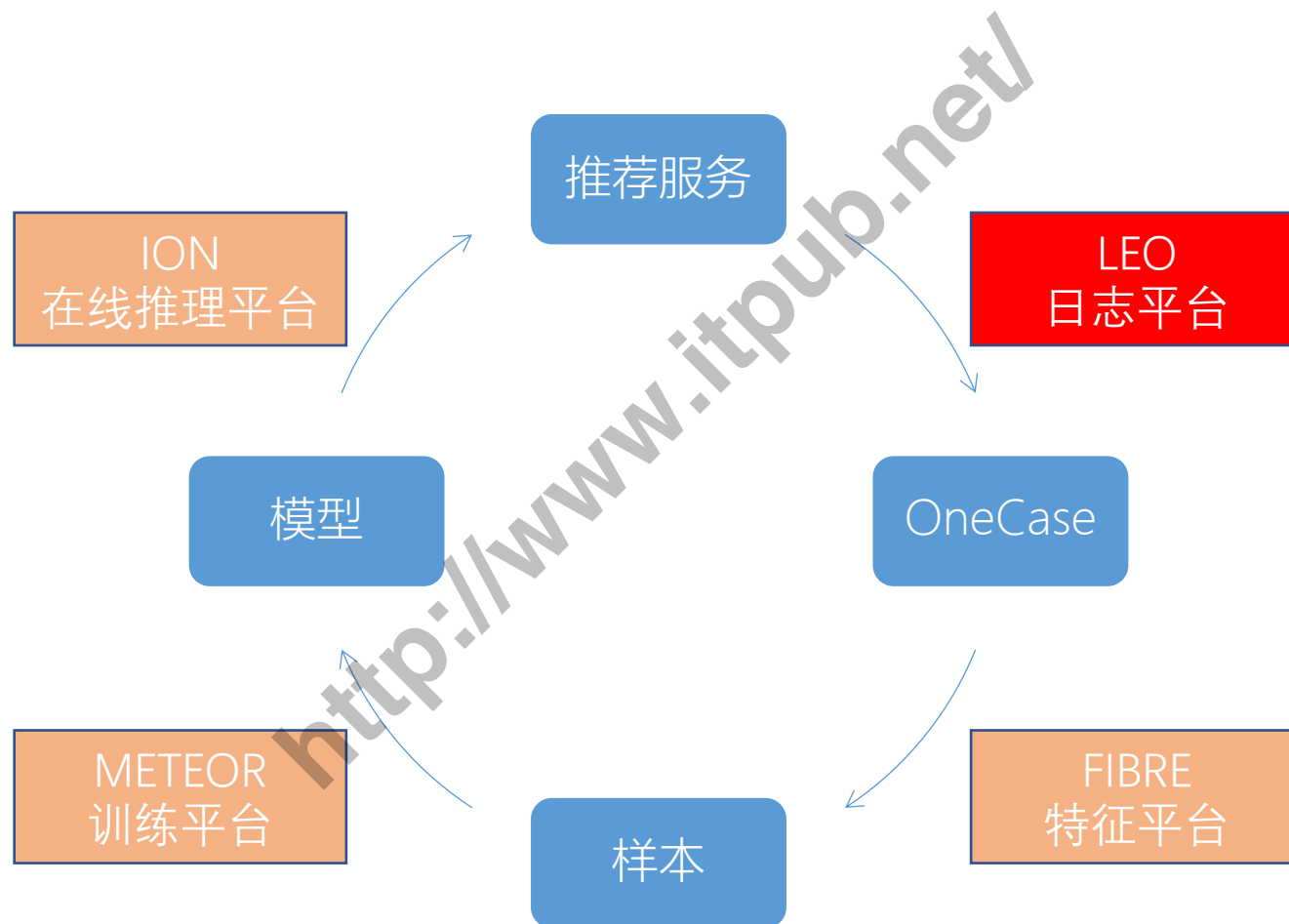


Rank架构（LFMI立方米系统）



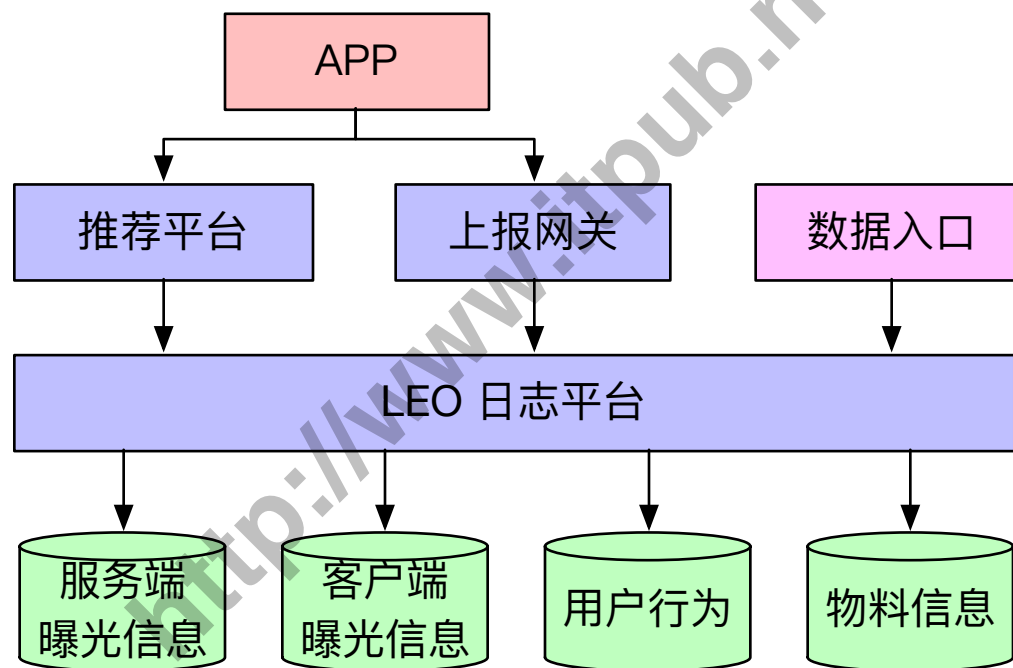


LEO (日志平台)



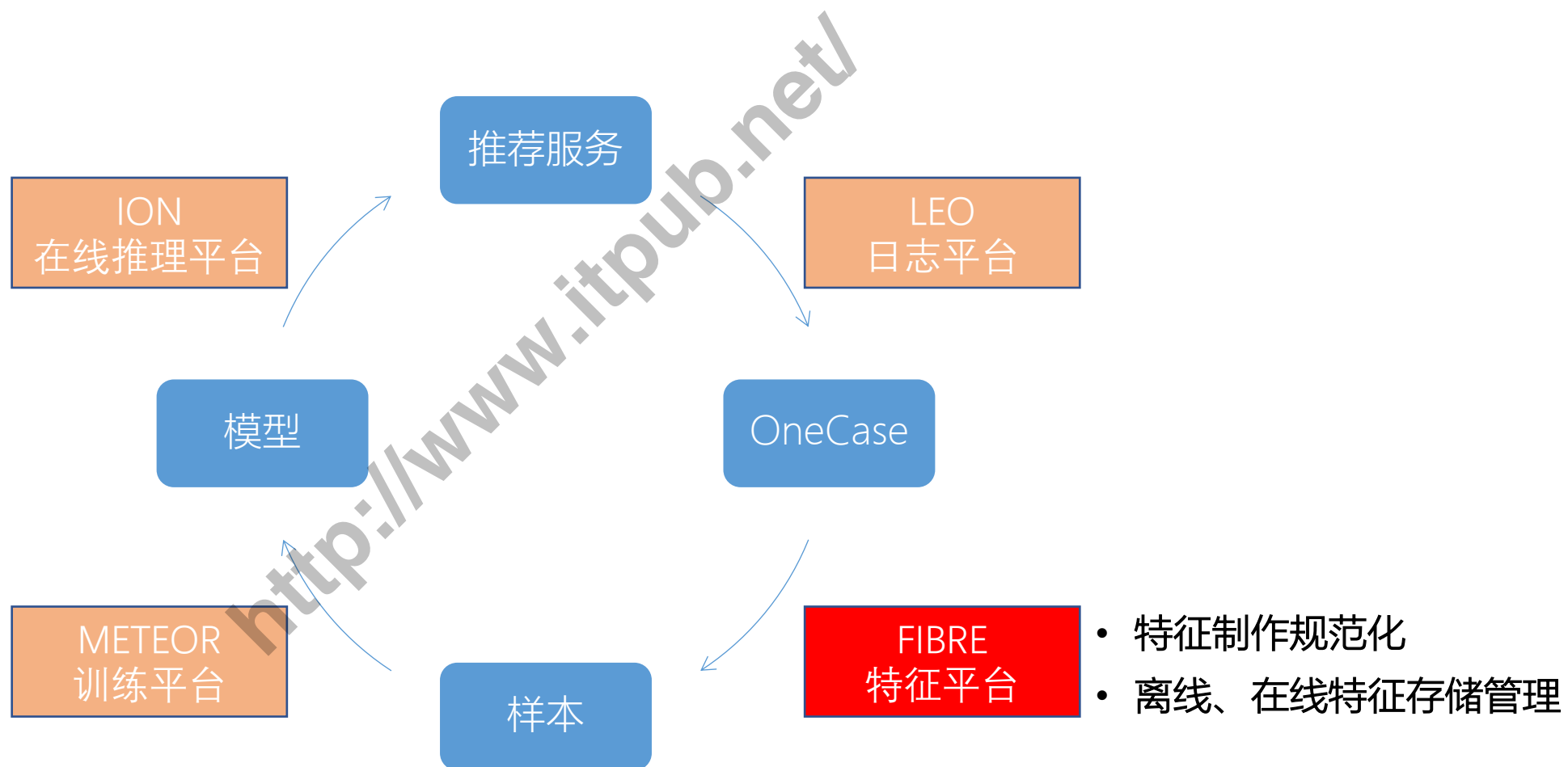


LEO (日志平台)



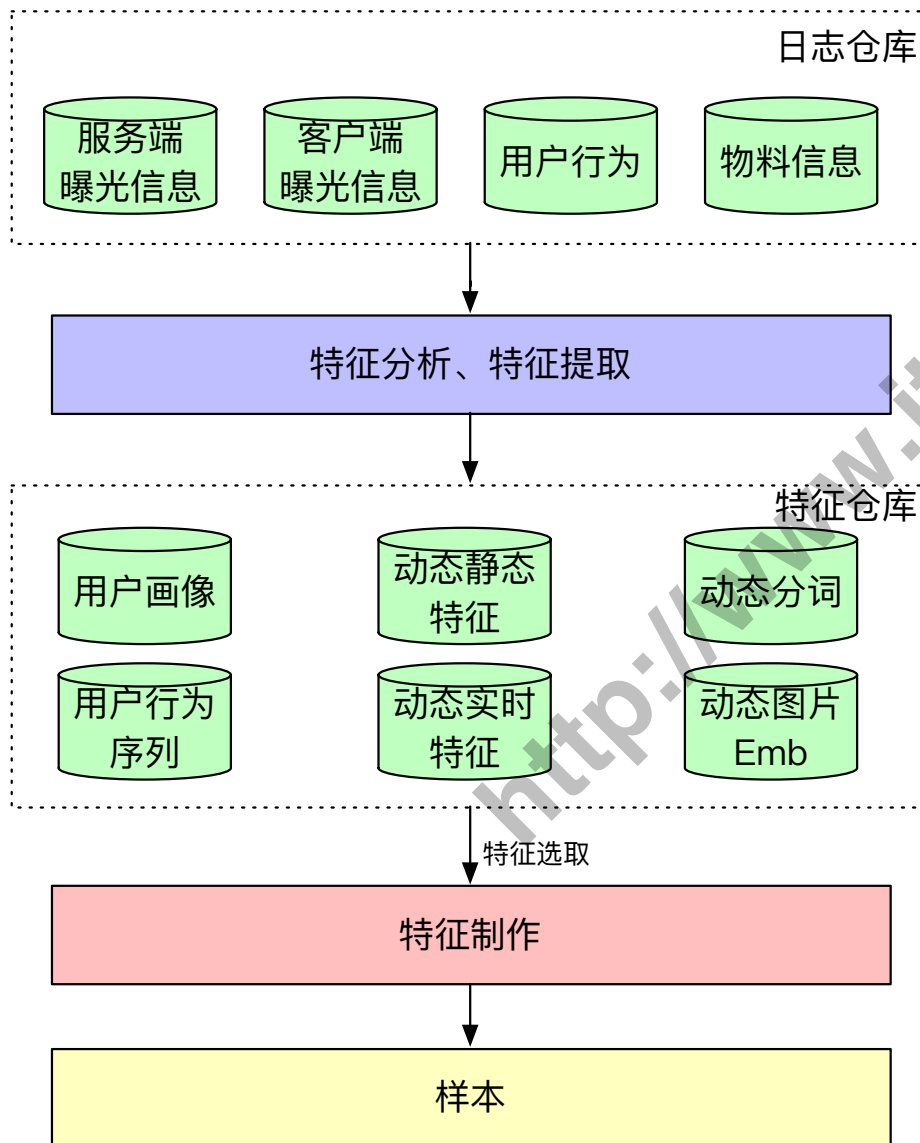


FIBRE (特征平台)





FIBRE (特征平台)



- 特征分析、特征提取

LEO系统的日志仓库中得到特征

- 特征管理

特征聚合形成特征仓库

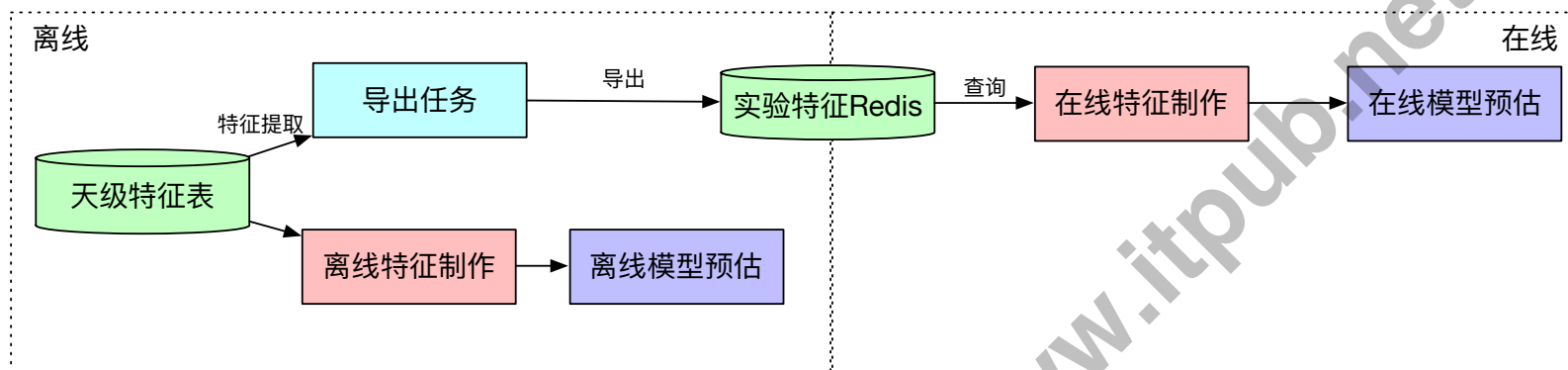
- 特征制作

将选取的特征制作成样本供训练、推理使用



FIBRE（特征平台）——特征制作

1.0 蛮荒时代



• 蛮荒时代

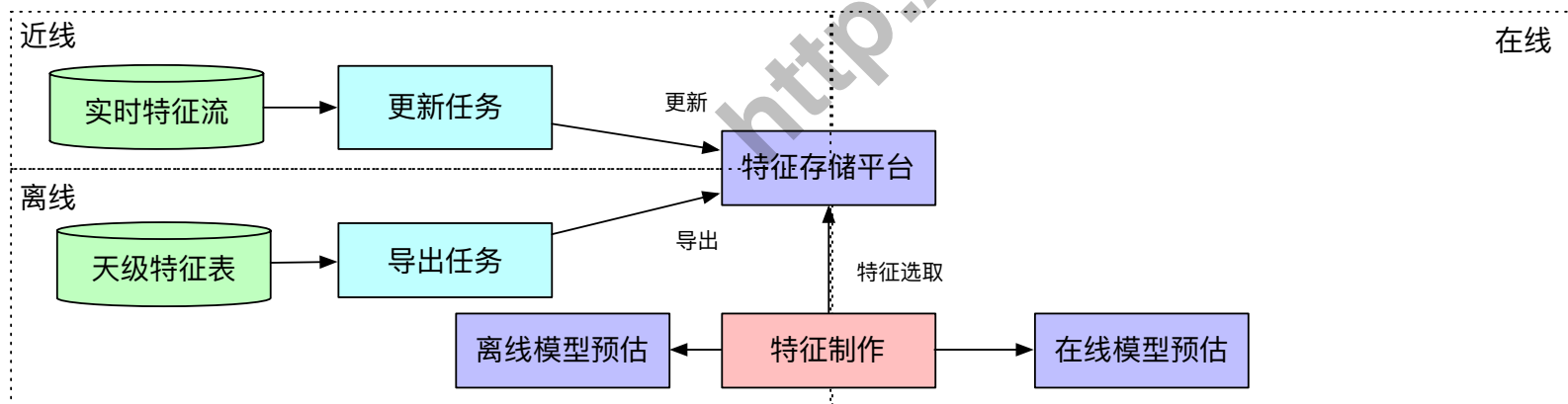
存在问题：特征制作逻辑存在在线、离线不一致情况；特征存储按照实验为维度，占用容量大

优势：策略开发能够掌控全部离线流程，在当时情况可把控

• 文明时代

优势：解决逻辑在线、离线不一致问题；通过特征存储平台，屏蔽策略对存储资源的考量

2.0 文明时代



• 效果

实验上线周期从周级别降低到天级别



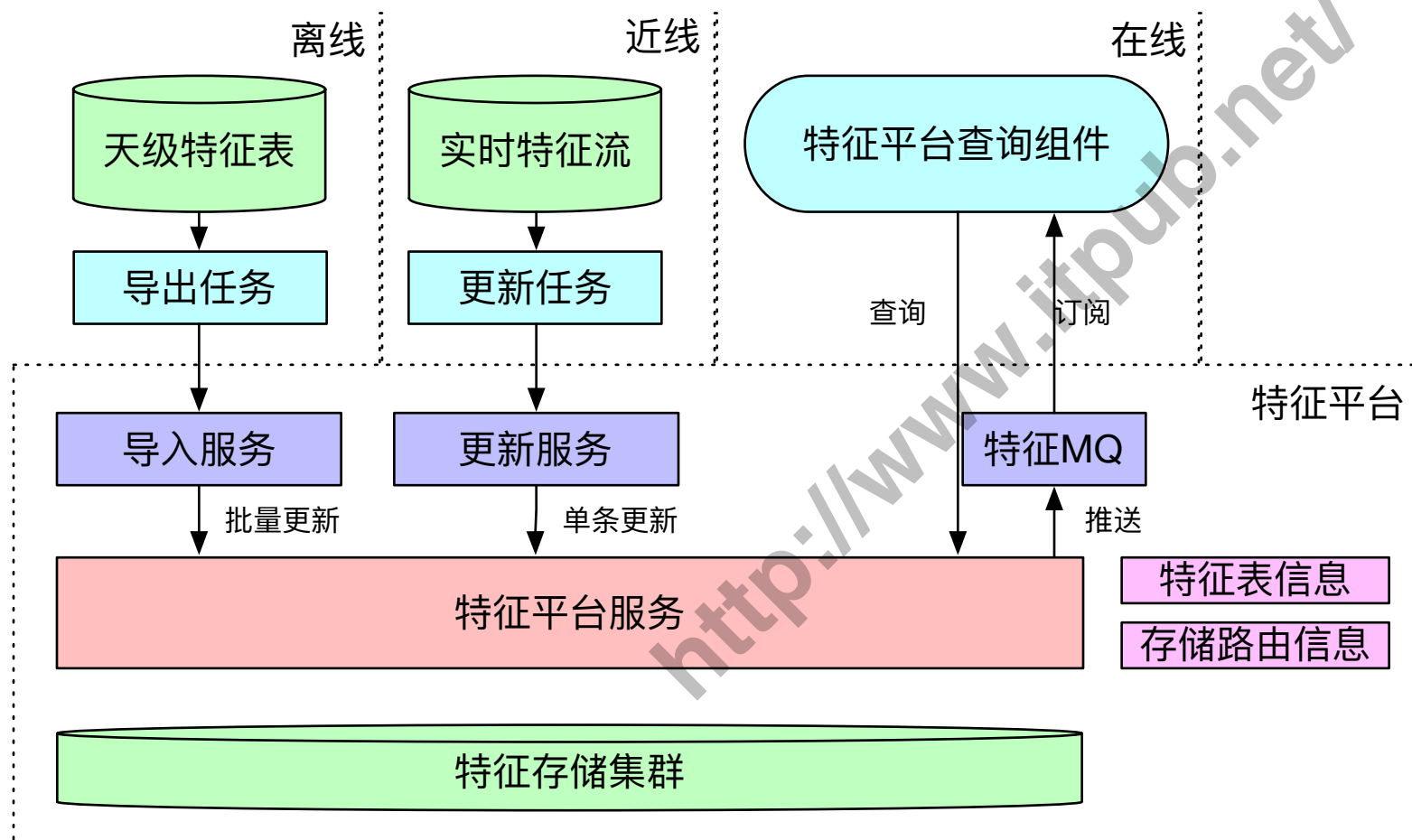
FIBRE (特征平台) —— 特征制作DSL

- 规范化特征选择、制作方法
- 解决离线、在线代码逻辑一致性
- 处理高维特征交叉

```
1 FeatureMaker:
2     {age_u} = hash(bucket(luser.age|, [1,2,3,4]))
3     {level_u} = hash(bucket(luser.level|, [10, 20, 30]))
4     {perfer_s} = hash(match(luser.perfer_gender_map|, lstar.gender|))
5     {click_list_u} = hash(luser.click_list|)
6 Group:
7     DeserializeSparse_31:sparse:{click_list_u}
8     IteratorGetNext15:dense:{age_u},{level_u},{prefer_s}
```



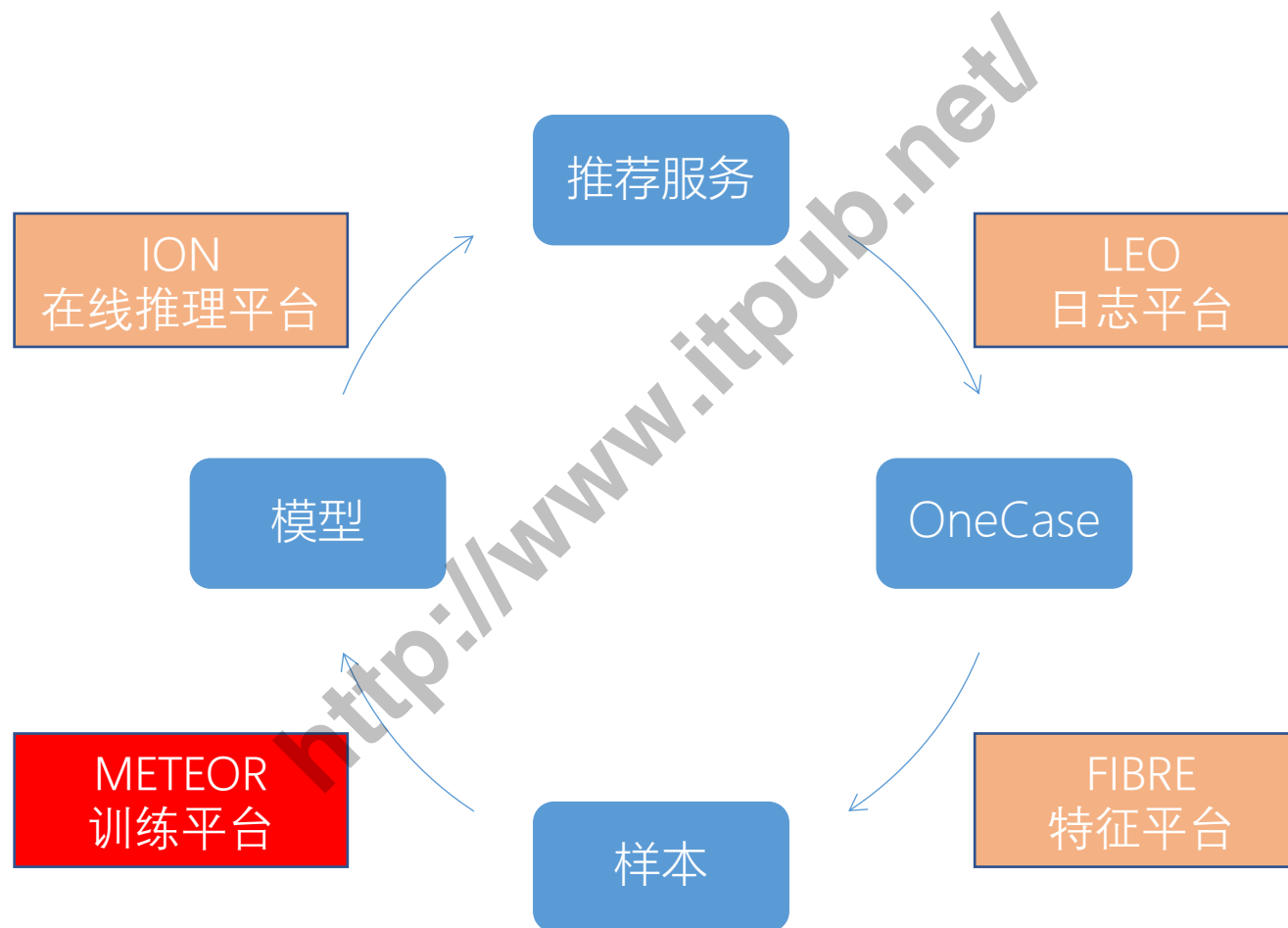
FIBRE (特征平台) —— 特征存储平台



- **统一特征的存储结构**
支持单值、列表、字典等特征结构
- **统一离线、在线的特征概念**
在线完全映射离线特征表结构，便于线上线下特征理解一致
- **提升存储效率**
减少以实验维度存储的重复特征
- **提供统一的查询组件**
支持特征缓存、推送功能，保证数据的最终一致性，提升获取效率



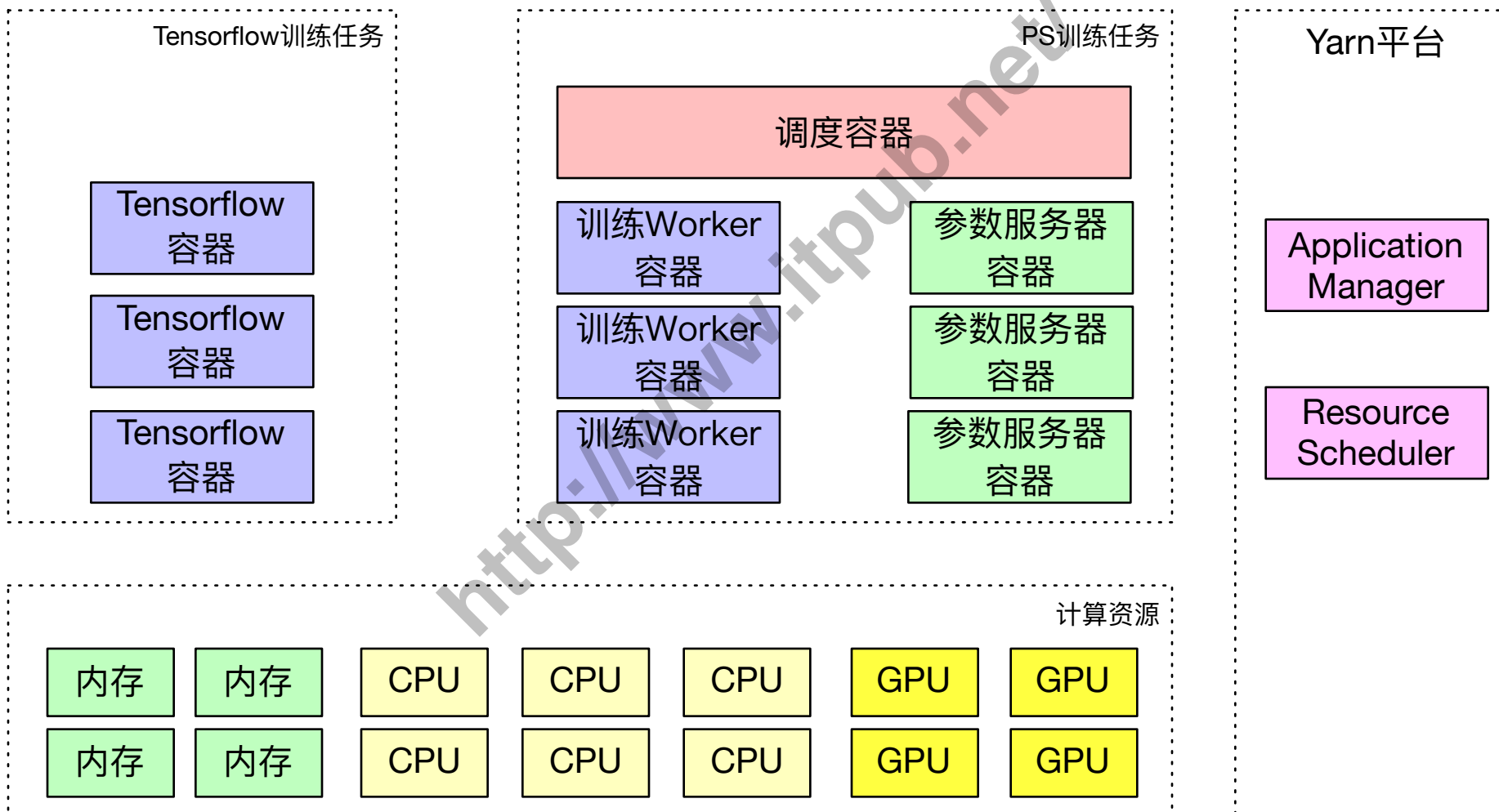
METEOR (训练平台)



- 深度模型训练
- 参数服务器

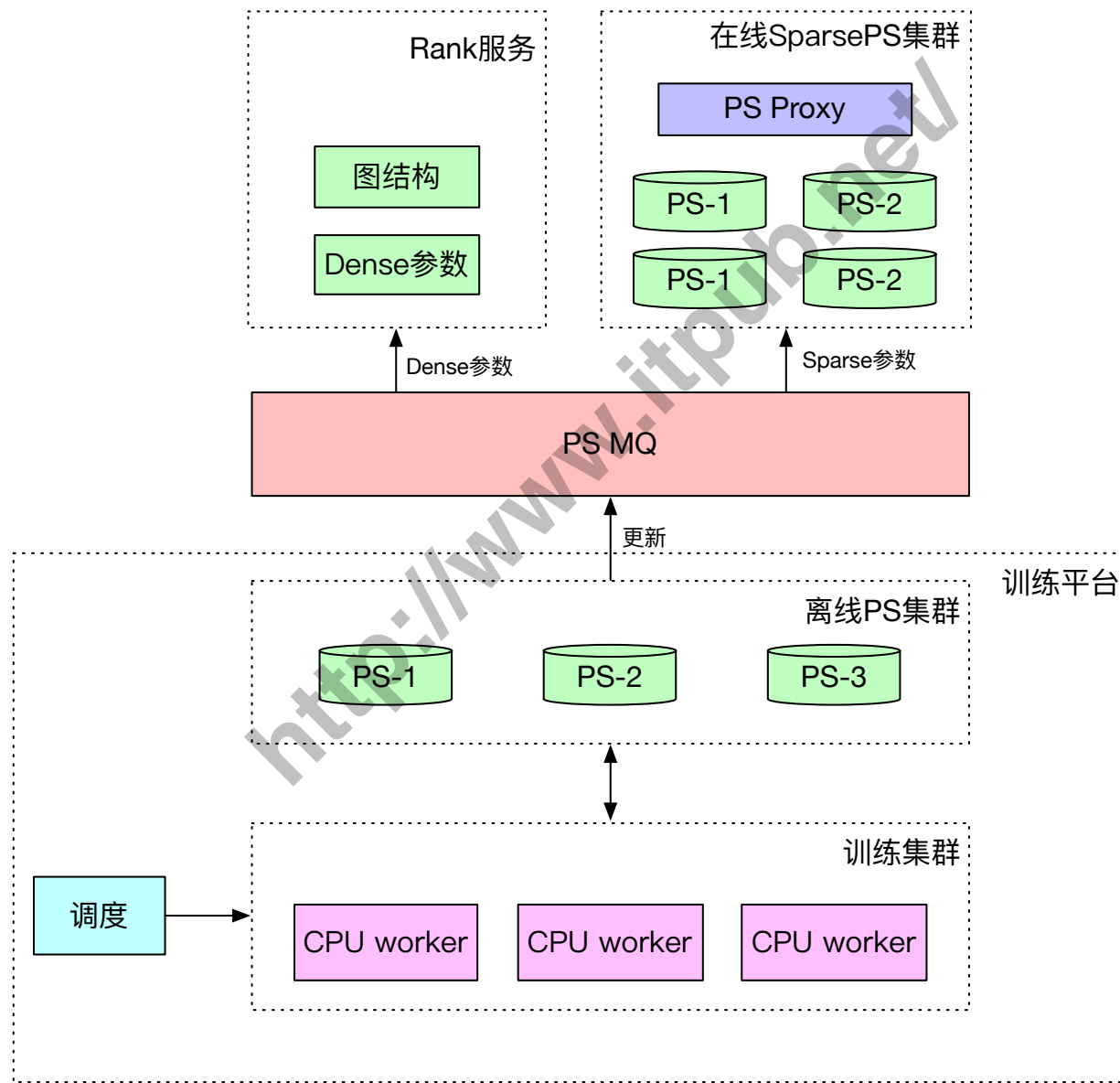


METEOR (训练平台)





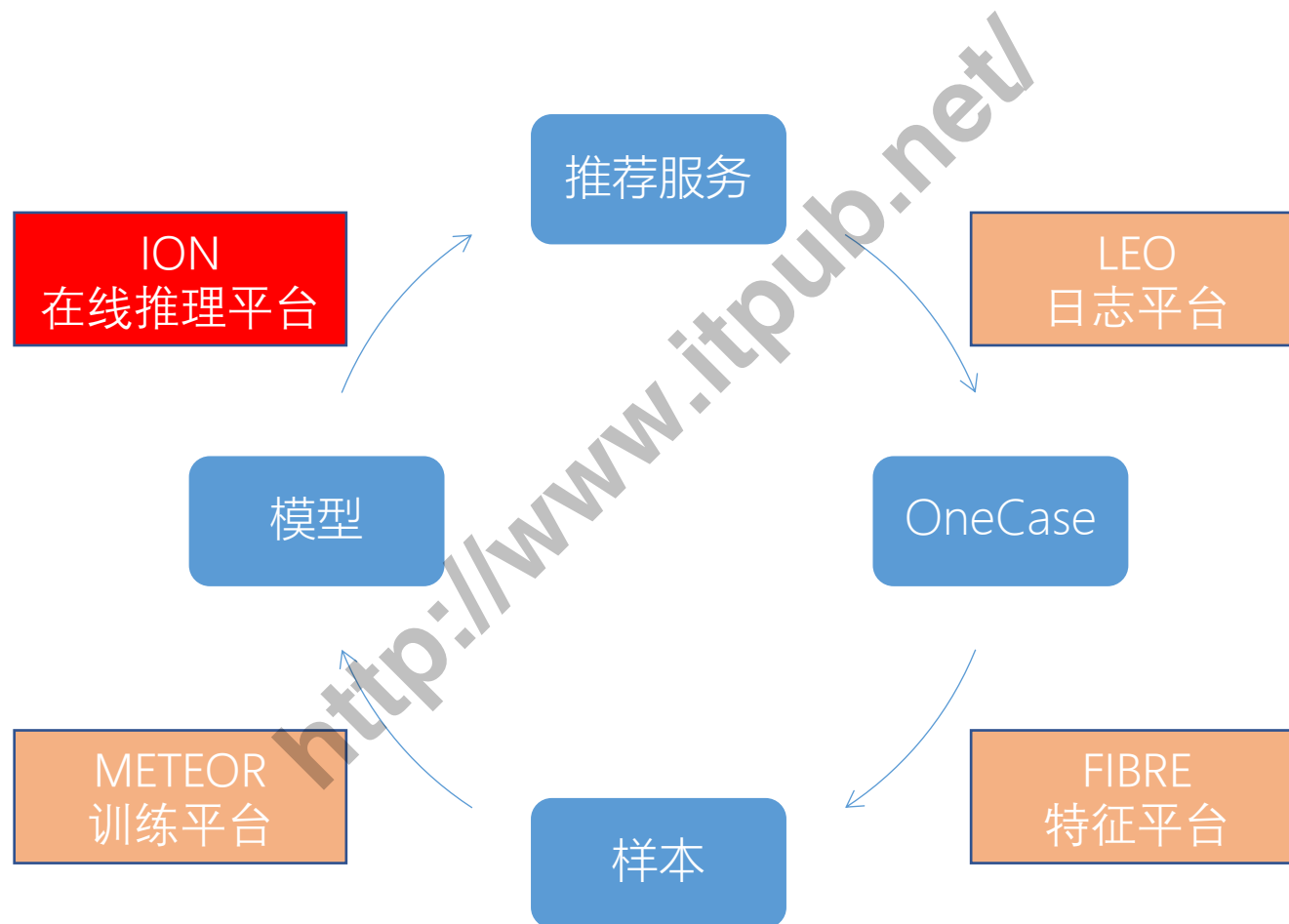
METEOR (训练平台) —— 参数服务器





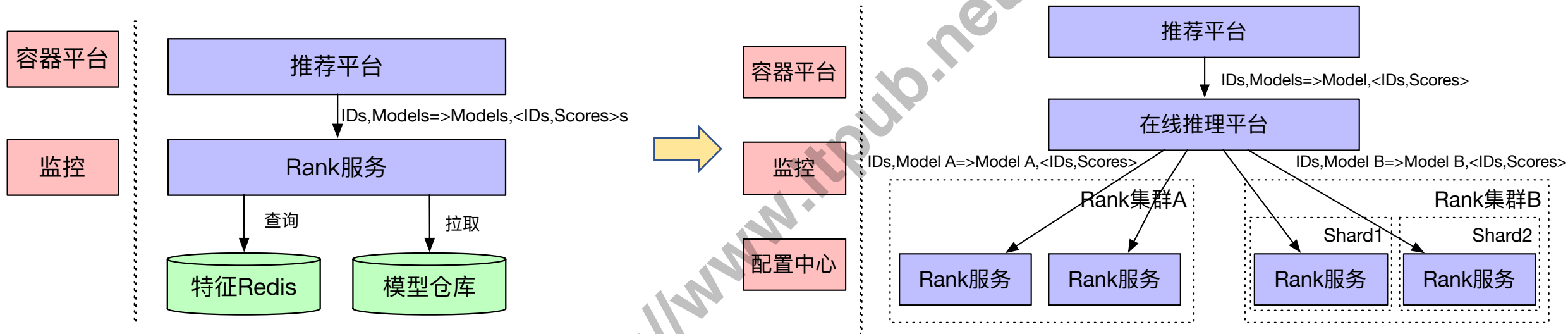
ION (在线推理平台)

- 模型规模增大
- 队列长度增加
- 迭代效率提升





ION（在线推理平台）——深层模型、千级队列



- 支持队列长度增加

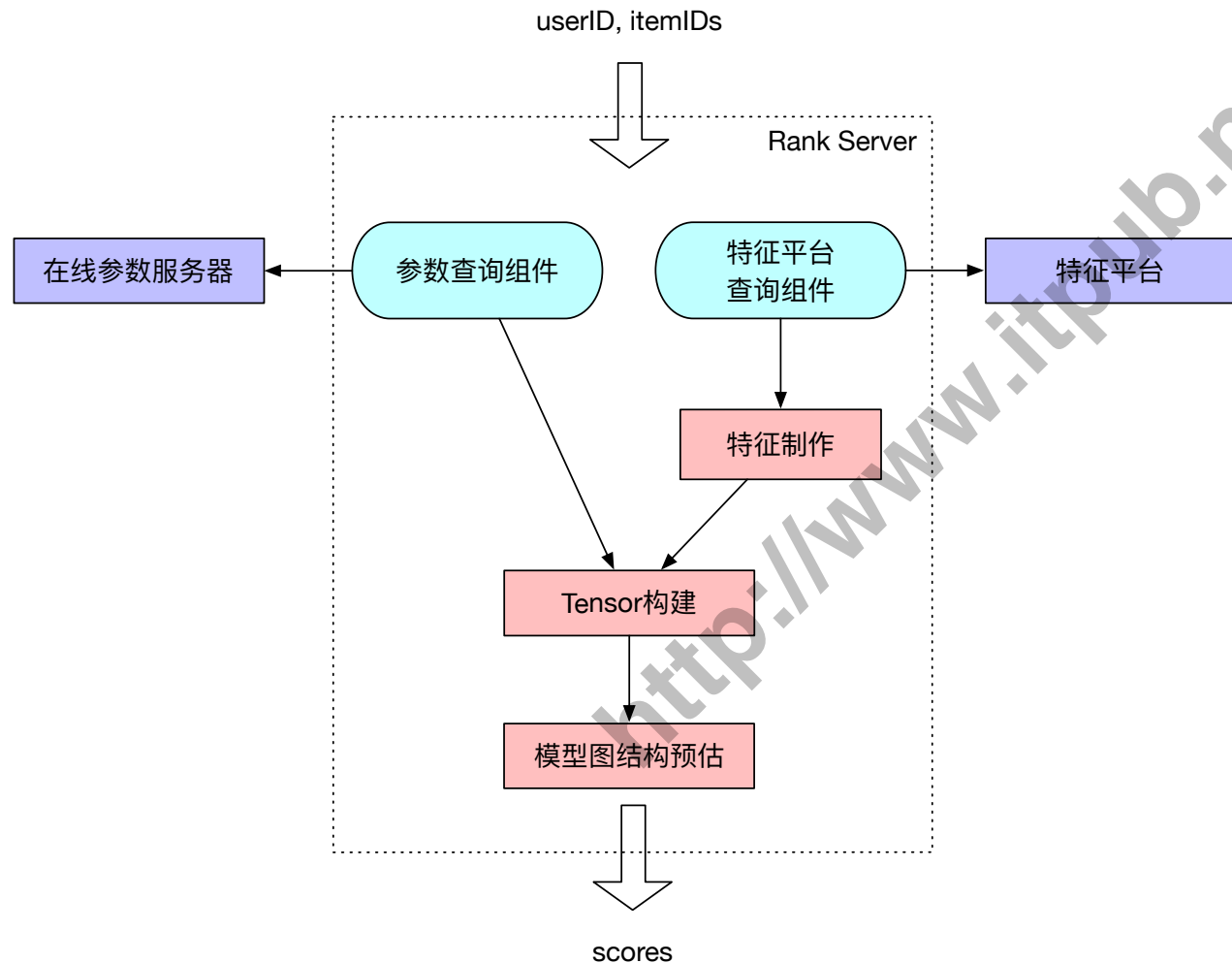
通过在线推理平台进行多维度拆分，将请求均匀拆分到下游的Rank服务，具备水平分布式能力；通过引入负载均衡策略，有效的降低长尾带来的影响。有效时间内Rank打分成功率在99.9%以上

- 支持10GB模型的深度预估

单机能够支持不超过10GB的模型预估。



ION (在线推理平台) —— 超大规模深度模型



- 模型和参数分离

DenseGraph+Sparse Parameter Server, 保证了预估服务的高性能

- DenseGraph

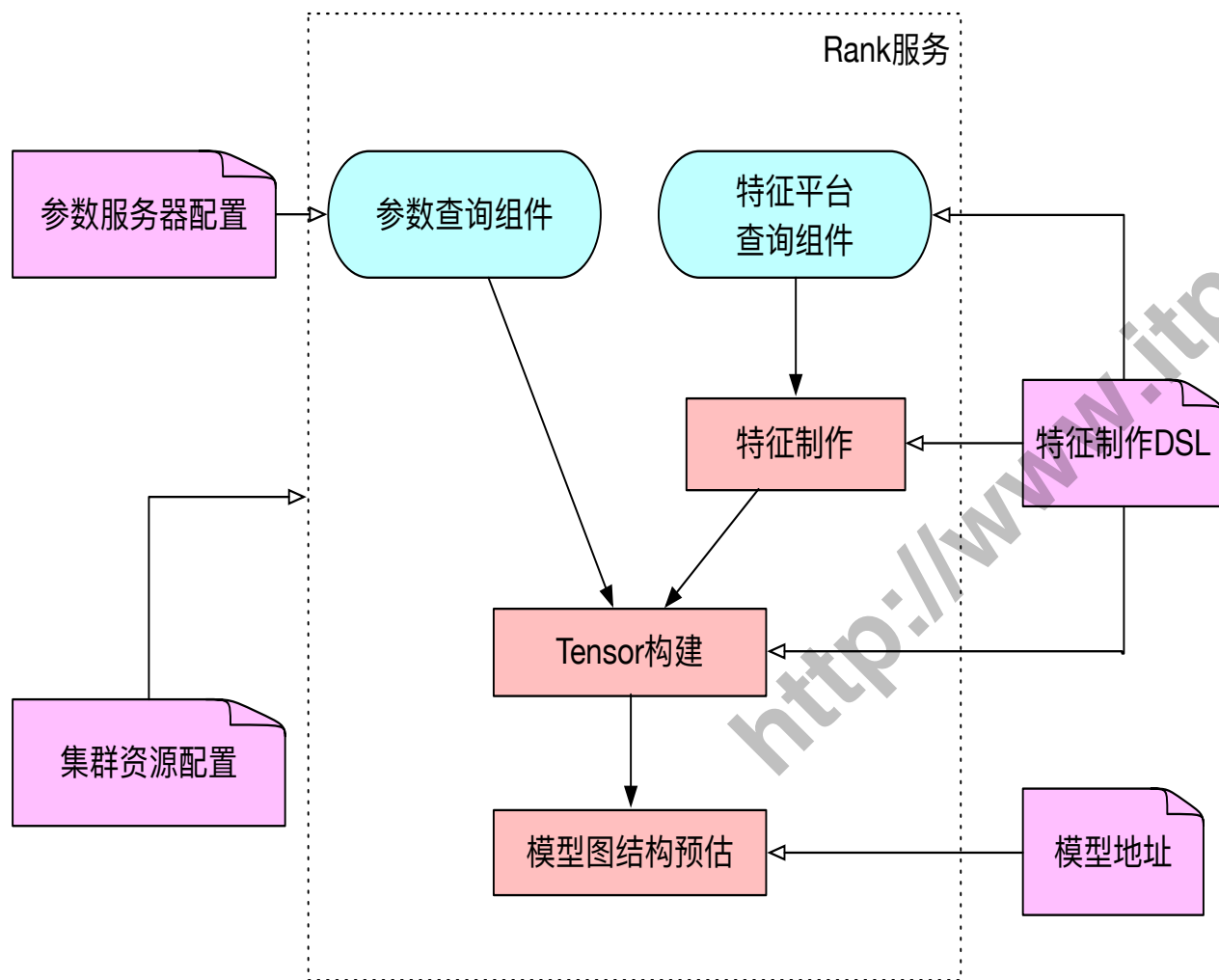
将大的Dense参数固化到模型, 内存加载, 解决了高维Dense参数存储在PS时的通信消耗

- Sparse Parameter Server

将低维Sparse参数存储在参数, 解决了亿级规模Sparse参数下的存储问题



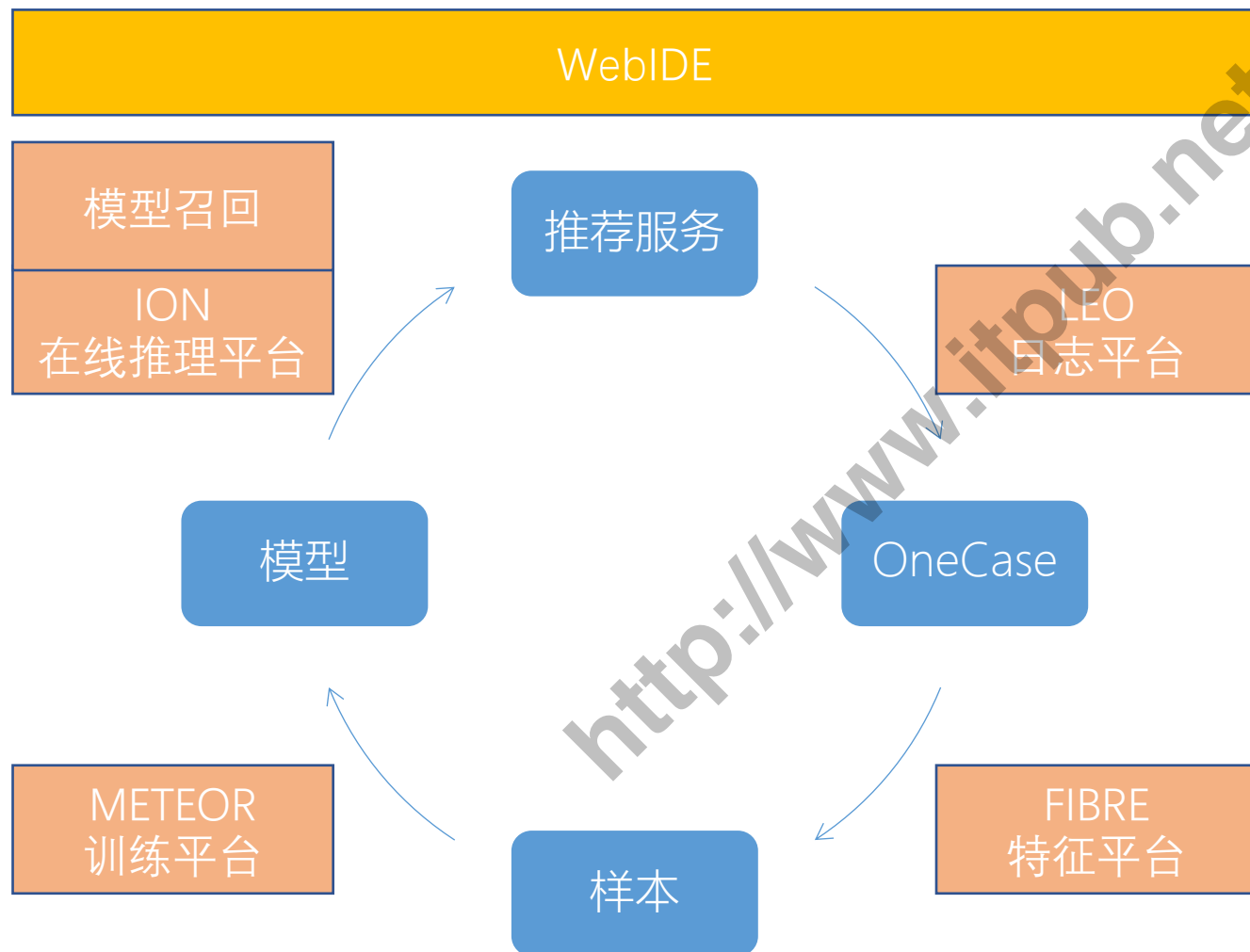
ION（在线推理平台）——流程配置化



```
1 FeatureMaker:
2   {age_u} = hash(bucket(luser.age1, [1,2,3,4]))
3   {level_u} = hash(bucket(luser.level1, [10, 20, 30]))
4   {perfer_s} = hash(match(luser.perfer_gender_map1, lstar.gender1))
5   {click_list_u} = hash(luser.click_list1)
6 Group:
7   DeserializeSparse_31:sparse:{click_list_u}
8   IteratorGetNext15:dense:{age_u},{level_u},{perfer_s}
9 Model:
10  hdfs://user/rank_v1_model.tar
11 PS:
12  v1_cluster
13 Resources:
14  CPU:8
15  Memory:8G
16  Cluster:16
```



未来规划



动态

直播

广告

...



Q&A

<http://www.mitpub.net/>