9 月 16 日，FATE 开源社区第 13 期圆桌会圆满落幕。本次圆桌会，由 FATE 团队的资深架构师邓凯老师，为大家介绍 FATE 的在线组件 FATE-Serving2.1.0 版本。

接下来带大家回顾经典**问答环节**，为各位朋友答疑解惑。

#问答环节

serving-admin 的 service 那个页面的 weight 是什么?

weight 就是权重，是我们在服务治理的时候需要关心的一些问题，比方说一个模型有可能分布在三台机器上，但是三机器的资源是不一样的，可能两台机器资源非常好，一台机器 CPU 非常弱，这时候需要对流量进行一些分配，可以把那两台的 weight 分成 100，然后这边分成 50，那么，比较弱的这台机器分的流量就是 1/5。

不用 deploy 模型吗?

deploy 模型有。就是我刚才说的 FATE-Flow 推模型的时候分成两个步骤，一个是 load，一个是 bind。

能否借助 exchange?

exchange 是双方不同 party 之间的一个中间节点，实际上我们会有其他的比这个功能更完善的组件。比方说，需要考虑到计费、流控、鉴权、路由相关的功能，比现在的 exchange 要更复杂，是因为本身在线业务就更复杂，它不像是离线传输的都是训练过程中产生的一些数据，而我们在线预测的时候传过去的请求可能每一笔都要计费的。

看代码，guest 是会遍历 host，把相关信息发给 host，是修改了 guest 合并多个 host 结果那个阶段吗?

实际上我们在做 2.0.0 的时候就已经为多 host 预测预留了，所以之前的代码会遍历 host，但是模型离线推给在线的时候，模型数据是不支持多 host 预测的。所以，这次我们是在模型的结构上进行了改动，在线和离线也都进行了一些相关的改动，使其能相互适配。由于我们之前在 2.0.0 的时候，在机制上设置上已经支持了多 host 的预测，所以你会在代码上看到会遍历 host，把相关信息发给 host，就是这个意思。

想问一下，比如有 x1,x2,x3；guest 发起在线推理时，有 id，x1，x2；host 那边是提前上传 x3 特征值么，如果 host 没做对应上传，预测时会有什么提示么?

需要分成两种情况：1、在 host 方 ID 匹配不到；2、能够匹配到 ID，但是特征不完整，无特征或者特征很少。第一种情况建议由 host 方报错，判断逻辑可以在 host 的 adaptor 中去自定义开发，需要根据自己需求决定。第二种情况一般不报错。也可以加入自定义逻辑，比如：拿不到特征或者特征命中率低于某个数值就认为这次预测请求失败。

鉴权这块是怎么设计呢？看默认是没有开启。

我们之前的投产的业务鉴权是放在了中间节点这一块，在两端的鉴权是做的比较弱的一块。鉴权现在是支持了 https，有颁发证书，双方安装各自相应的证书再进行一个鉴权。鉴权的重点是我们放在了中间节点的设计，就是 exchange 那一块的设计。

只部署了 2 台服务器 host、guest 能否做联邦学习，还是只能做离线联邦?

是可以的。因为在线比离线的需要的资源少多了，如果是离线都能训练的话，在线就一点问题都没有。

这个支持计费的加强 exchange 什么时候发布?

我们现在做的一些 FATE-Cloud 的组件里面有，这些组件开源出去会有计费的一些功能，我们自己也有一些非开源的项目，计费相关的逻辑是比较完善的。

我理解 host 方的特征系统接入第三方的服务，在 serving 时就写好相关的接口。如果我想要更新这个第三方的特征服务，是可以不中断相关的服务么，也就是可以对第三方的特征系统热更新不?

FATE-Serving 不关心它的更新逻辑，只要保证你那边设计好接口，我通过这个接口能够访问能够正常就可以。比方说，更新特征的时候，你这个接口至少还是能够访问的，能不能支持热更新是在那个系统去决定，而不是 Fate-serving 决定，因为这个也没办法决定。

有没有单纯在在线预测节点做路由转发的 exchange 节点?

deploy 你现在用离线的 exchange 也是可以进行转发的，只不过只是一个转发功能，没有其他功能。

我想问一下多 host 的话，是不是也是要打包多个 host 的 extension 的 jar 包去替换。

如果你自己做实验，你是可以这么做的。正常情况下，如果投产的话，多 host 意味着是多个公司或者多个部门，获取特征的接口有可能是由不同的系统提供，然后这是需要各公司自行开发的。

纵向场景里面，进行在线的联邦推理两边数据不需要对齐吗。在实际的业务场景中，如果一方拿到了一条用户的实时数据，另一方并没有该用户的数据怎么办?

一般逻辑是对方没有数据、拿不到特征，这时候是报错的。（补充：报错逻辑可以在 host 的 adaptor 中去自定义开发 ，比如：拿不到特征或者特征命中率低于某个数值就认为这次预测请求失败，也可以做成 ID 匹配不到也不报错，需要根据自己需求决定。）

如何做多节点布署（至少 3 个节点）?

只有在你自己去做实验的时候，你可能搞一个节点，实际上生产环境是至少三个节点，因为使用了 zookeeper，zookeeper 建议部署奇数个节点 1 个节点、3 个节点、5 个节点、7 个节点这样子。所以怎么部署多节点应该是比较简单一个问题，取决于你是怎么规划的。好比说，给你三台机器，你想怎么分配哪台机器部署什么组件，其实你怎么部署都可以，因为它通过了 zookeeper 寻址，所以只要端口不冲突部署在哪都无所谓。

麻烦问下 exchange 是什么角色，之前没有遇到过。

我想应该在离线的时候已经用过 rollsite 了，对吧？之前我们经常使用 rollsite 来进行做进行一个转发的。在不同公司有可能它是不同的网络，如果两个公司并不知道对方的存在，它需要一个中间角色进行转发，如果是双方都知道的话，没有必要做这个事情的，就没有必要 exchange 这个角色，它就是一个路由转换。（补充说明：在生产环境中，有可能不同的参与方属于不同的公司， 公司与公司之间的网络打通是一个比较繁重的事情，涉及到各个公司的网络策略、带宽分配、安全审计等等事物，exchange 就是把整个网络变成一个星型网络，由负责 exchange 的公司负责对接各个合作方，完成上述工作，这样就减轻了各合作方自行互联的负担。）

多节点布署（至少 3 个节点）的相关资料望能提供一下!

其实在文档上我们之前有一个类似的图，可以看一下，如果说觉得不够，可以在群里面给我们提一些建议，我们这边会安排同学去更新文档。

生产环境部署，更推荐 exchange 还是非 exchange？还没部署过 exchange 方式?

需要知道生产环境是你们公司内部不同的部门之间进行一个交互，还是不同的公司之间进行一个交互。如果是不同公司，倾向于有 exchange；如果是在内部的话是没必要的。如果在内部交互中使用，只是多一点交互，不能带来特别多的好处。

老师，请问进行一次在线预测的时间大约是多少?

这个是不定的。可以回答一下我们目前线上的一些情况，大概是在 100 多毫秒，这是跨了两个公司的预测。因为跨了公网，有可能会耗时严重一点。如果是在公司内部不同部门之间进行内网的交互，应该会好很多。在线预测的耗时很大部分是放在 host 那边获取特征那一步，获取特征时候，如果系统做的不够好的话，它的延迟高，整条链路的延迟就很高。

一套集群如何加载多个模型？起不同的服务 ID 可以吗?

服务 ID 是跟模型是一一对应的，如果是多个模型肯定是不同的服务 ID，如果是相同的服务 ID 就是直接覆盖了。不同的模型，相同的服务 ID，就可能会导致不正确的选择模型，所以不同的模型一定是不同的服务 ID。

最新版本 FATE-Serving 对 FATE 的版本有要求吗？最低版本是多少呢？还是不限版本?

近一年发布的版本都是兼容的，最早以前的（如，两年前的）不一定，因为没有测过。

直观想，两个纵向联邦参与方能够同时拿到一个用户数据的可能性很小，如果一方拿到数据，另一方没有就报错的话，那在线推理的成功率会不会很低?

这是在你选择合作方的时候是需要考虑的问题。一般来说需要 host 那一方的数据比较全面，因为如果你拿不到特征，训练的模型只有一半起作用，这个效果也不会很好。如果说双方的数据非常少，传来传去都找不到自己的那些 ID 相关的特征，做联邦学习也没有什么太大的意义。一定是有一方的数据非常全，做这个才有意义。

FATE-Serving2.1.0 更新到 kubefate 上了么?

目前没有。

支持 dsl v2 训练出来的模型吗?

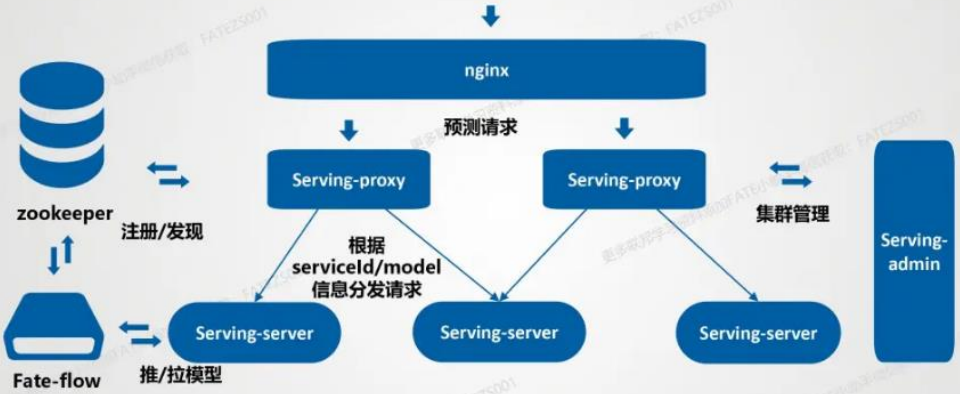
FATE-Serving 版本 >=2.0.4，FATE 版本>=1.5.1 的话，dsl v2 训练是支持 Serving 调用的，但 dsl v2 推到 Serving 前，需要先 deploy 训练模型去指定生成的预测推理工作流。FATE dsl v1 和 dsl v2 的一个比较大的区别是 v1 会帮你自动推导预测工作流，但导致的问题是当模型训练完之后，预测工作流就不可变了，所以 dsl v2 通过去掉自动推导、增加 deploy 操作由用户灵活去决定预测的工作流。

为什么不支持横向模型的预测?

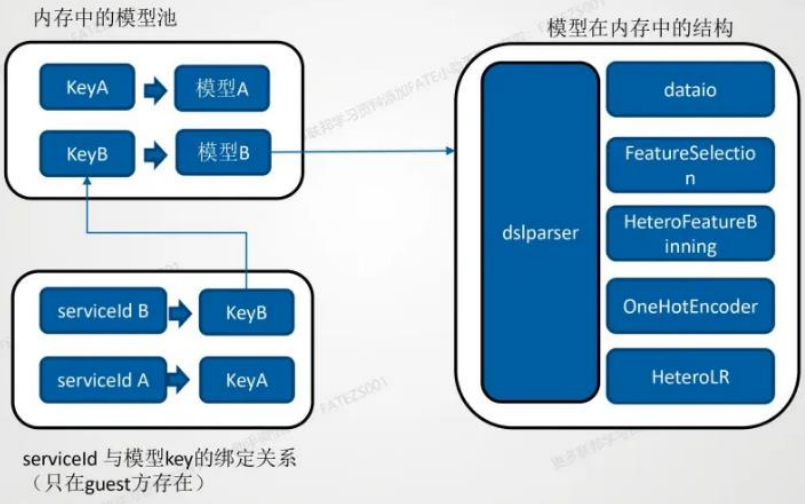
横向模型训练完成后，每一方都会得到完成的模型，这个和纵向逻辑是不一样的。纵向联邦无论是离线训练还是在线推理阶段，都是需要双方协助的，而横向联邦则是离线训练完成后，在线阶段每一方拿到完整模型，可以单独用自己的数据来直接进行推理。而 FATE 里面横向联邦逻辑回归、GBDT 或者 NN，在推理阶段，可以使用第三方引擎来进行推理。基于这种考虑，FATE-v1.7 会推出横向模型转成第三方可用的模型文件功能，届时用户可以使用该模型文件，再导入到第三方引擎去做推理即可。

以下为本次圆桌会的部分内容介绍，添加小助手（FATEZS001）可获取详细资料：

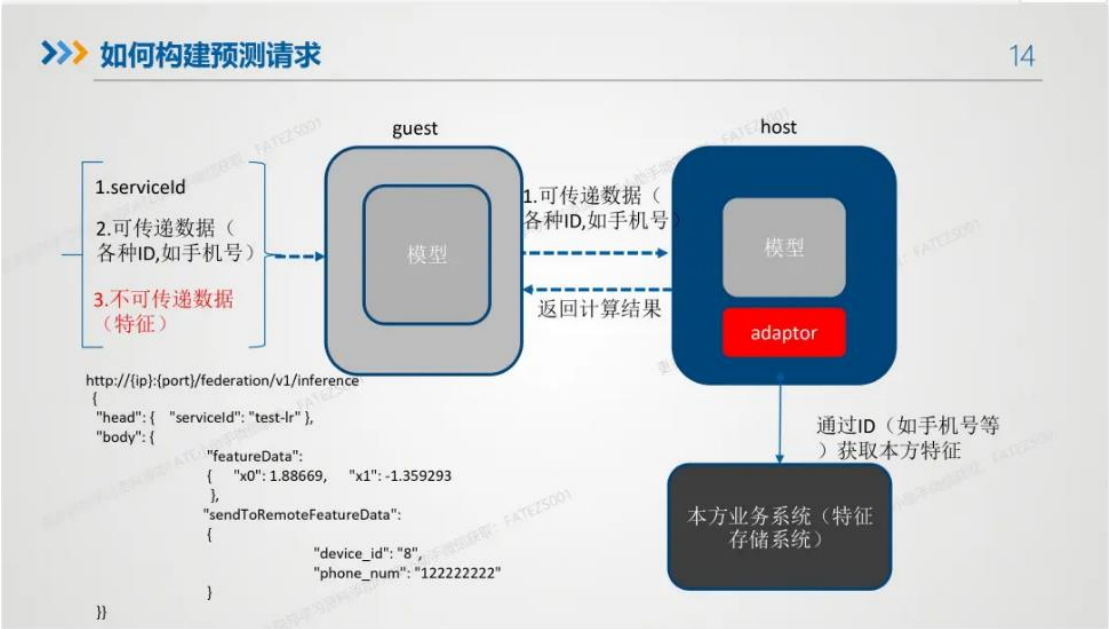
部署架构



模型的初始化



serviceId 与模型key的绑定关系
（只在guest方存在）



获取会议 PPT，或对圆桌会还有别的疑问？欢迎联系 FATE 开源社区助手获得帮助。

原文链接：https://mp.weixin.qq.com/s/dDkZS-wyc_Btk1ATBbLqXA