规模化的联邦学习系统设计

TOWARDS FEDERATED LEARNING AT SCALE: SYSTEM DESIGN

目录

Catalogue

背景

联邦学习是一种分布式机器学习算法，可以实现在大量分散数据的情况下进行模型训练并且能够有效保护数据的隐私。但是要让大规模的分散的数据参与训练产生价值离不开**性能强大，稳定的大规模联邦学习系统。**

背景

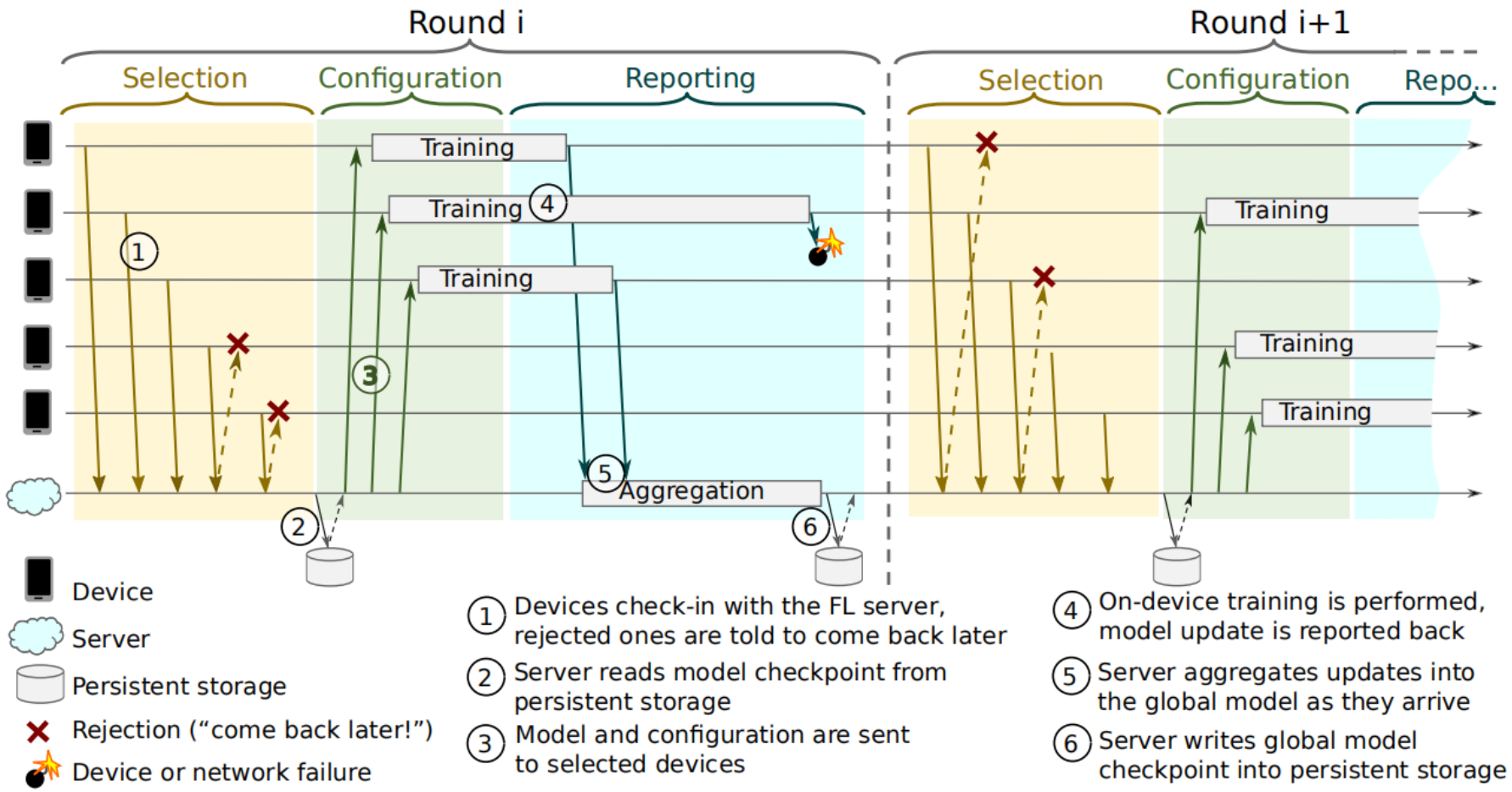


Figure 1: Federated Learning Protocol

基本架构--终端

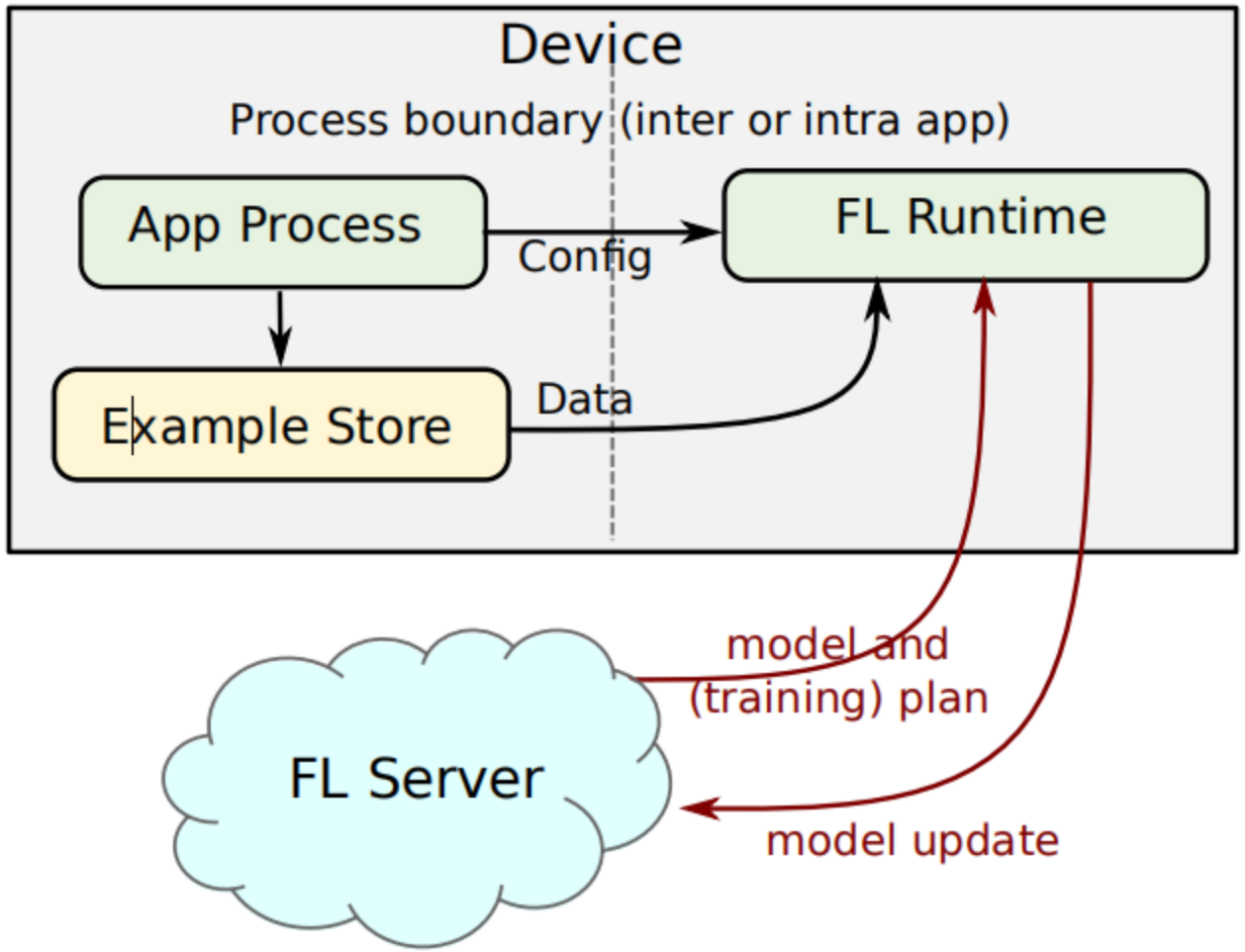


Figure 2: Device Architecture

FL Runtime是负责与Server进行交互，从服务器接收模型和训练计划。上传模型的更新梯度或者更新后的模型参数给服务器端

App Process负责将在终端设备存储的数据转化为FL Runtime可以访问的方式，比如将数据存储在SQLite里等方式，可以被FL Runtime访问的数据池称为Example Store。App Process也会将这个Example Store以Config的方式注册到FL Runtime可以访问到里面的数据

基本架构--服务端

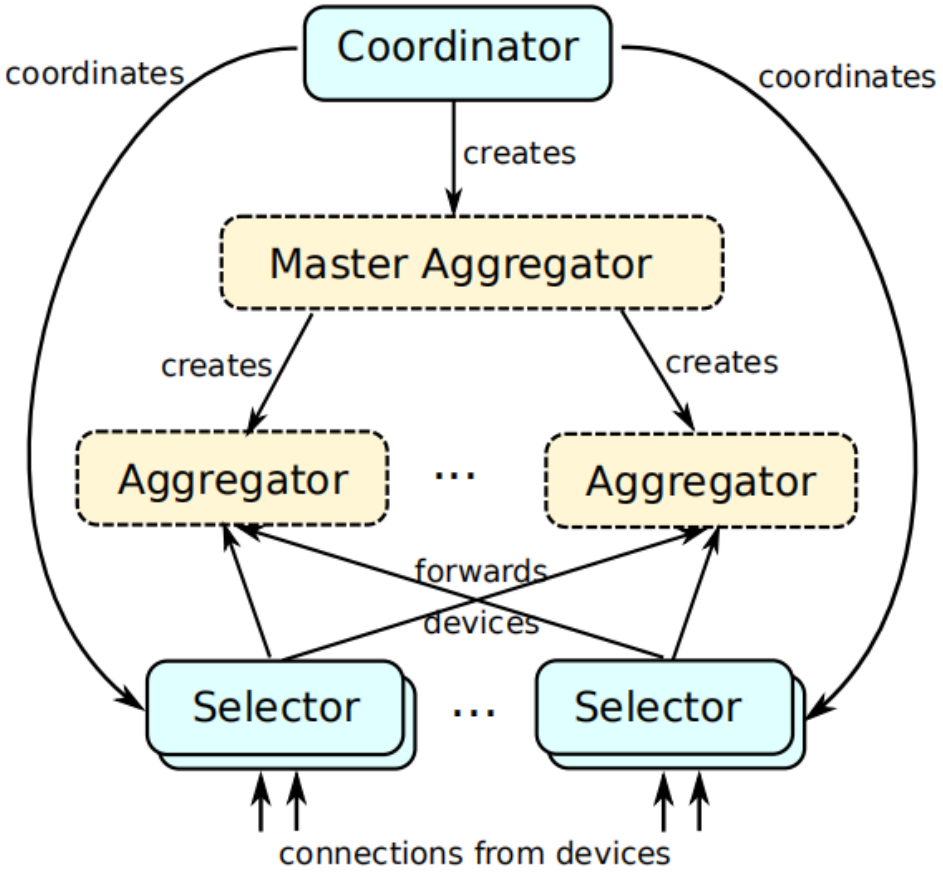


Figure 3: Actors in the FL Server Architecture

背景：服务端采用了角色编程模型来设计。在角色编程模型中万物皆为角色，不同的角色之间可以相互通信，或者创建新的角色等等。以上的操作都可以并行执行。

Coordinator是服务端的角色，一个服务端可以有多个Coordinator，用来管理每一个FL Job.每一个Coordinator的地址以及它所管理的FL Job会被注册到一个共享池中。

Master Aggregator也是服务端的角色，根据终端的连接个数以及模型更新的数据规模来调整Aggregator的个数

Selector也是服务端的角色，维持着和终端设备的连接，每一个Selector将已连接设备的子集发送给聚合器，这样协调器可以更方便的去将FL任务发送给各个终端

协议层

1. 终端选择机制：Server会根据一系列的指标例如终端的电量，是否处在充电状态以及数据质量和规模等信息来选择 符合条件的终端的子集。如果没有足够的终端进行这一轮的FL任务，那么会server会跳过这一轮并指导终端进行重连

2. 训练机制： server会将FL的训练计划以及最新的局和模型发送给终端。

3. 报告机制： 当有有足够的终端训练完成将更新梯度或者模型参数传递给Server的时候，Server会执行这一轮的模型聚合

4. 步态调整机制：步态调整机制就是根据FL训练的任务大小动态的调整连接的终端连接的数量 。 Server会根据当前FL训练任务的情况告诉终端设备一个重连的最佳窗口时间，终端设备就根据这个窗口时间进行重连。

设备层

1. 编程配置：App Progress 会将FL任务的名称以及Example Store 信息配置到FL Runtme进程，并且该进程会被安卓的任务管理器所调度。

2. 任务调度： FL Runtime会和Server联系声明自己的可调度状态，Server会根据查看自己这边是否有FL任务空闲，如果有就将FL计划传递给终端，否则就传递给终端最佳的等待窗口时间。

3. 任务执行： 当FL Runtime接受到FL计划后，就根据FL计划去向Example store读取数据然后执行训练操作。

4. 汇报：当FL Runtime执行训练操作后，将更新梯度或者模型参数传递给Server。

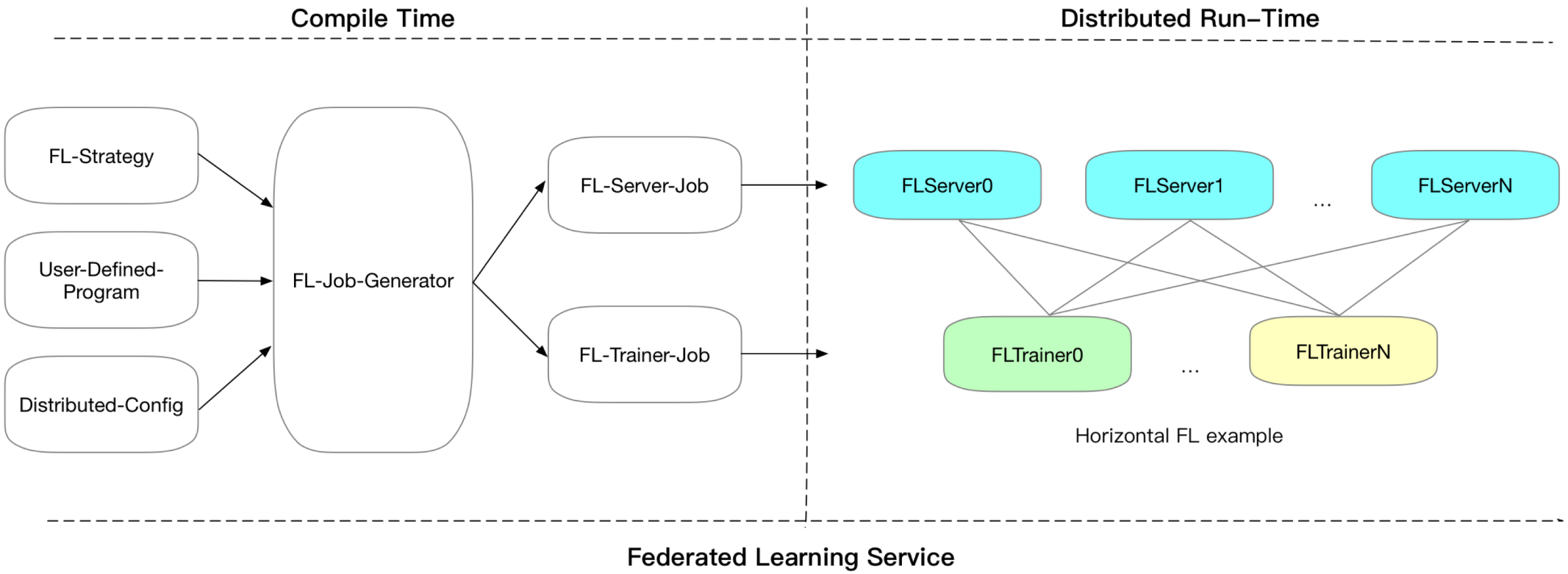
服务层

1. 流水线机制：Seletor阶段因为没有任何的依赖，因此党第一轮模型聚合的Selector结束之后可以马上进行下一轮的终端选择阶段，直到前一轮的配置和汇报阶段结束。

3. 恢复和容错机制：如果Selector或者Aggregator崩溃了，只是会丢到与终端设备的连接，不会对整个系统造成影响，当Master Aggregator崩溃的话，当前这一轮的FL任务会被终止，Coordinator会将它重启。如果Coordinator崩溃了，共享资源池会根据之前Coordinator在里面的信息重新生成一个新的Coordinator.

总结

PaddleFL联邦学习库



感谢观看

Thank you for
watching