

单元化架构 (Cell) 下的 高性能服务实践

梁宇鹏 @一乐

新浪微博通讯技术专家

- 服务器硬件
- 操作系统
- 共享库
- 算法
- 架构



单元化架构

一种优化的可能

- 我们在做什么
- 单元化是什么
- 为什么单元化
- 怎么做到的
- 困难与解决



924

关注

16232545

粉丝

10346

微博



陆琪 (设置备注)    <http://weibo.com/jdluqi>

工作邀约, 请联系: 18668077100。。知我爱我, 罪我笑我, 近我忘我

♂ | 浙江 杭州 | 标签 ▾

✓ 已关注 | 取消

☐ 私信

求关注

@他

更多 ▾

他的主页

微博


星店

个人资料

博客

今天也许是中国自媒体、新浪微博微博以及社交网络跨时代的一天。16120000条私信发送, 几十分钟内六十万条私信回馈。。。无数人同时在线做情感测试。从来没有感受到新浪微博有如此大的力量啊。迎接新时代吧, 各位。——陆琪

10月24日 21:47 来自新浪微博 | 举报

 (666) | 转发(12268) | 收藏 | 评论(1751)



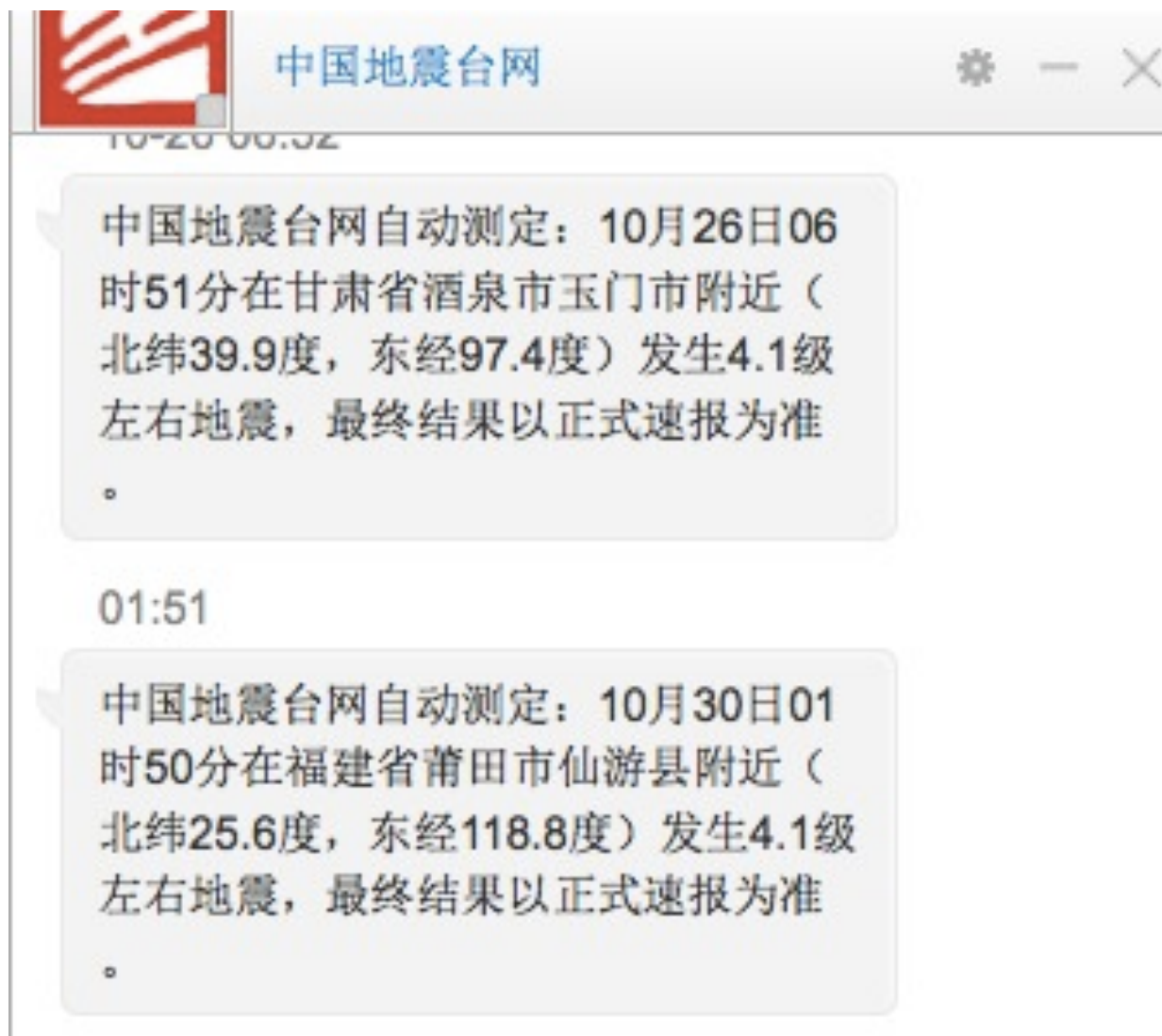
中国地震台网



中国地震台网自动测定：10月26日06时51分在甘肃省酒泉市玉门市附近（北纬39.9度，东经97.4度）发生4.1级左右地震，最终结果以正式速报为准。

01:51

中国地震台网自动测定：10月30日01时50分在福建省莆田市仙游县附近（北纬25.6度，东经118.8度）发生4.1级左右地震，最终结果以正式速报为准。



- 内容推送系统 Castalia
 - 用户筛选
 - 消息群发
 - 百万条每秒

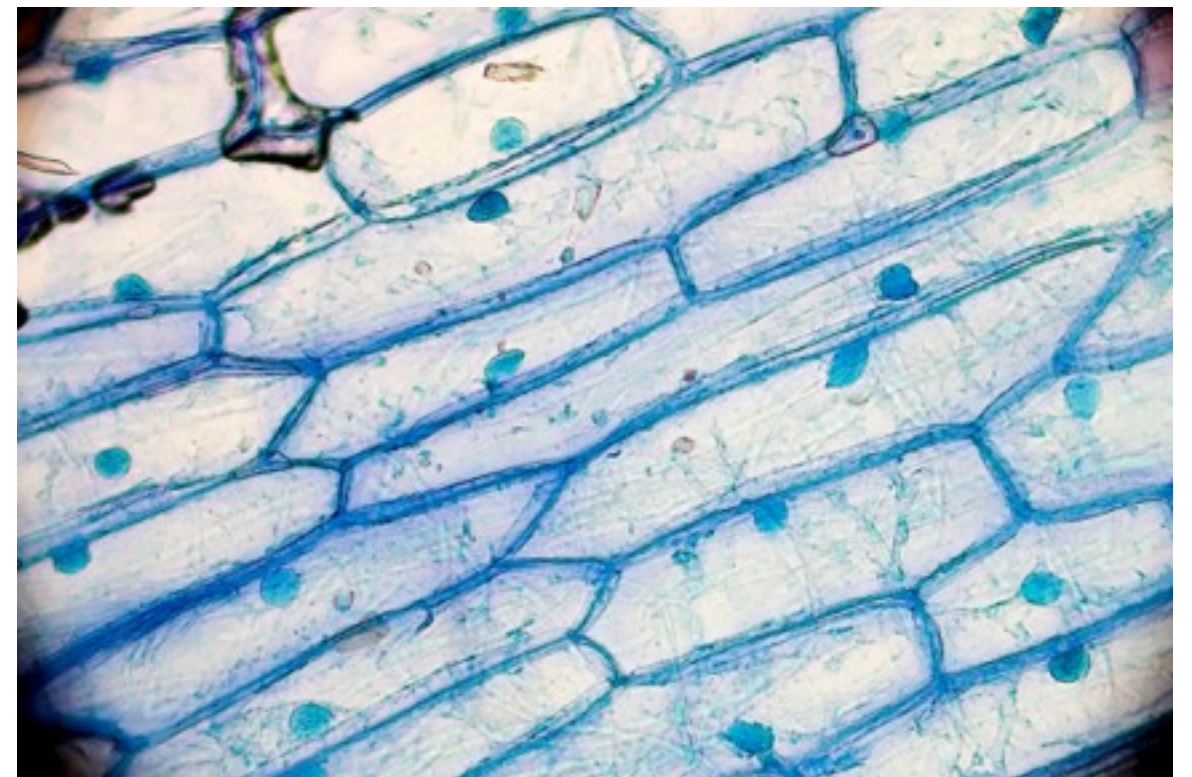




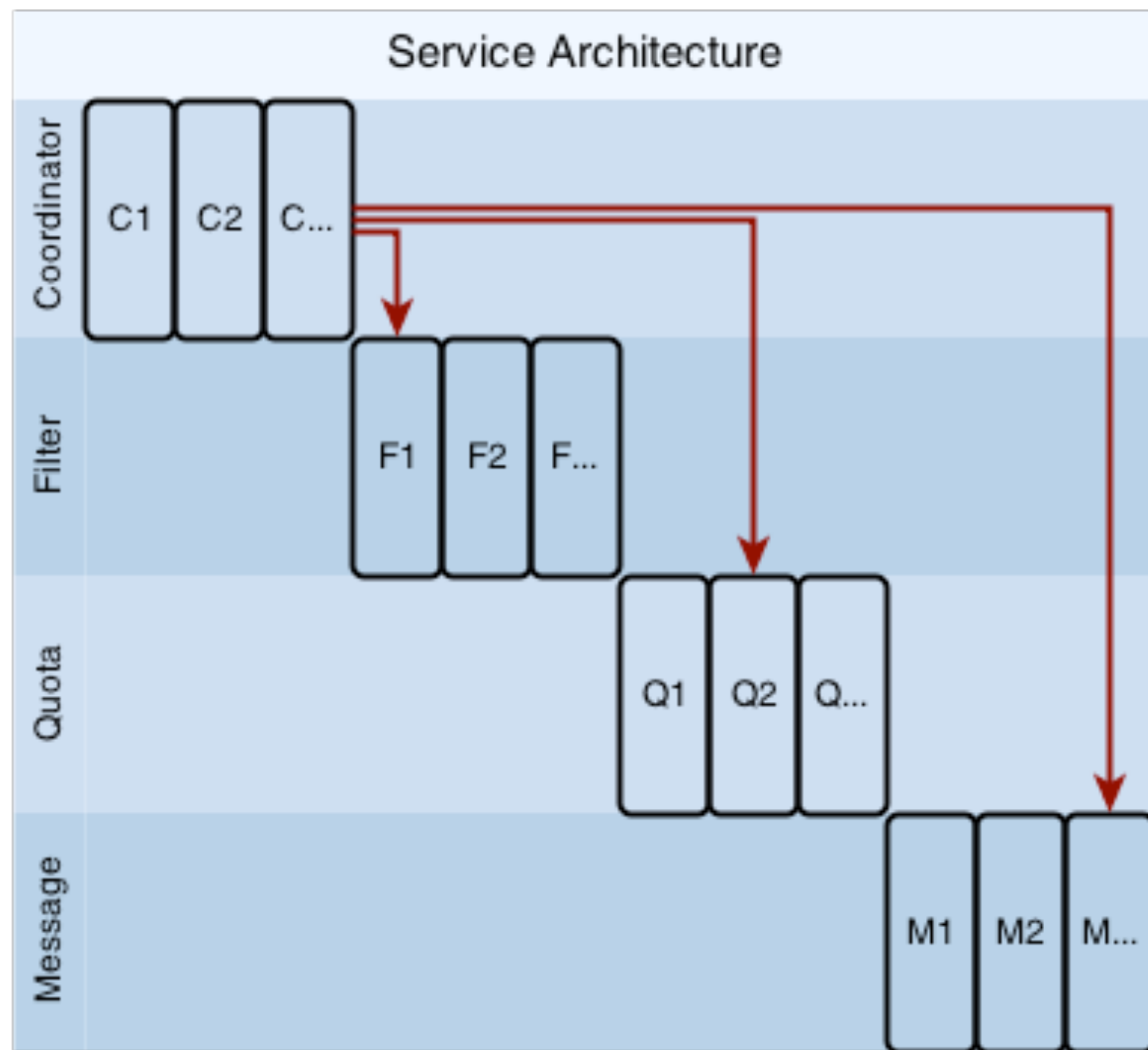
没有不合理的需求
只有不合理的设计

- 我们在做什么
- 单元化是什么
- 为什么单元化
- 怎么做到的
- 困难与解决

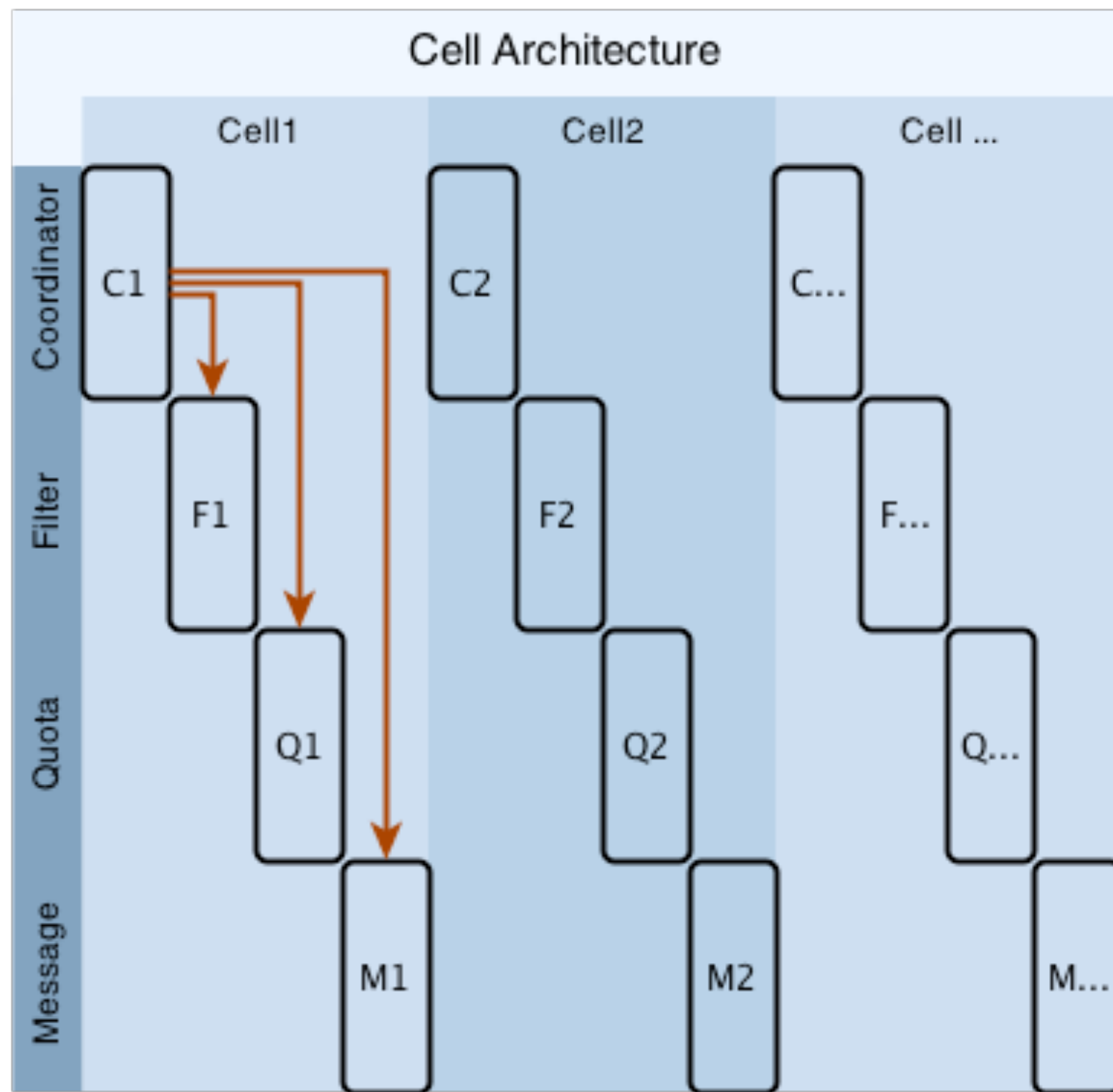
- 单元 (Cell)
 - 满足某个分区所有业务操作
 - 自包含的安装
- 分区 (Shard)
 - 数据集的一个子集
 - 一部分用户



- 分层服务
 - 不同数量节点
 - 不同分区算法
- 前端**随机**选择下层节点



- 分层服务
 - 相同数量节点
 - 相同分区算法
- 前端访问指定下层节点

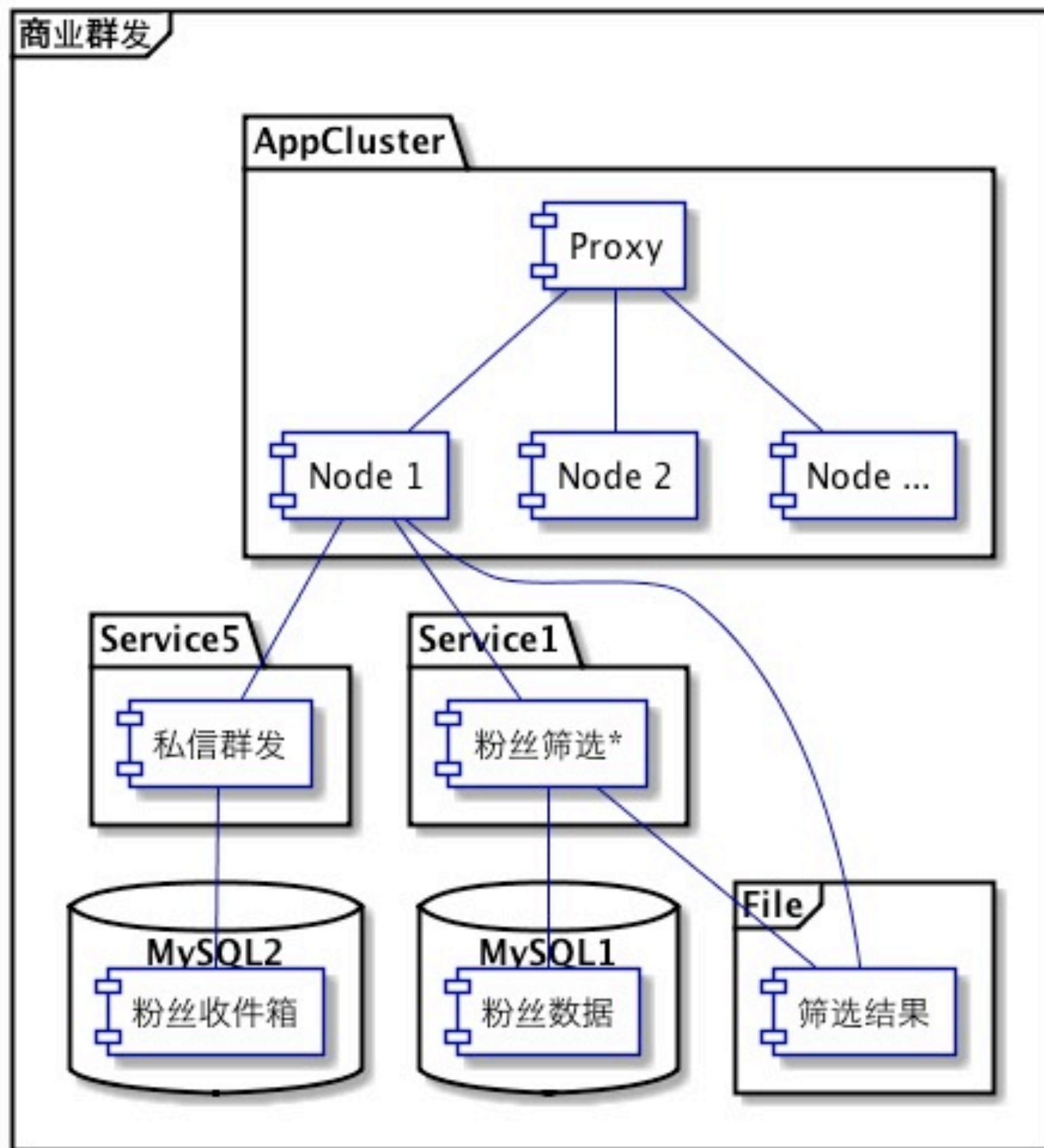


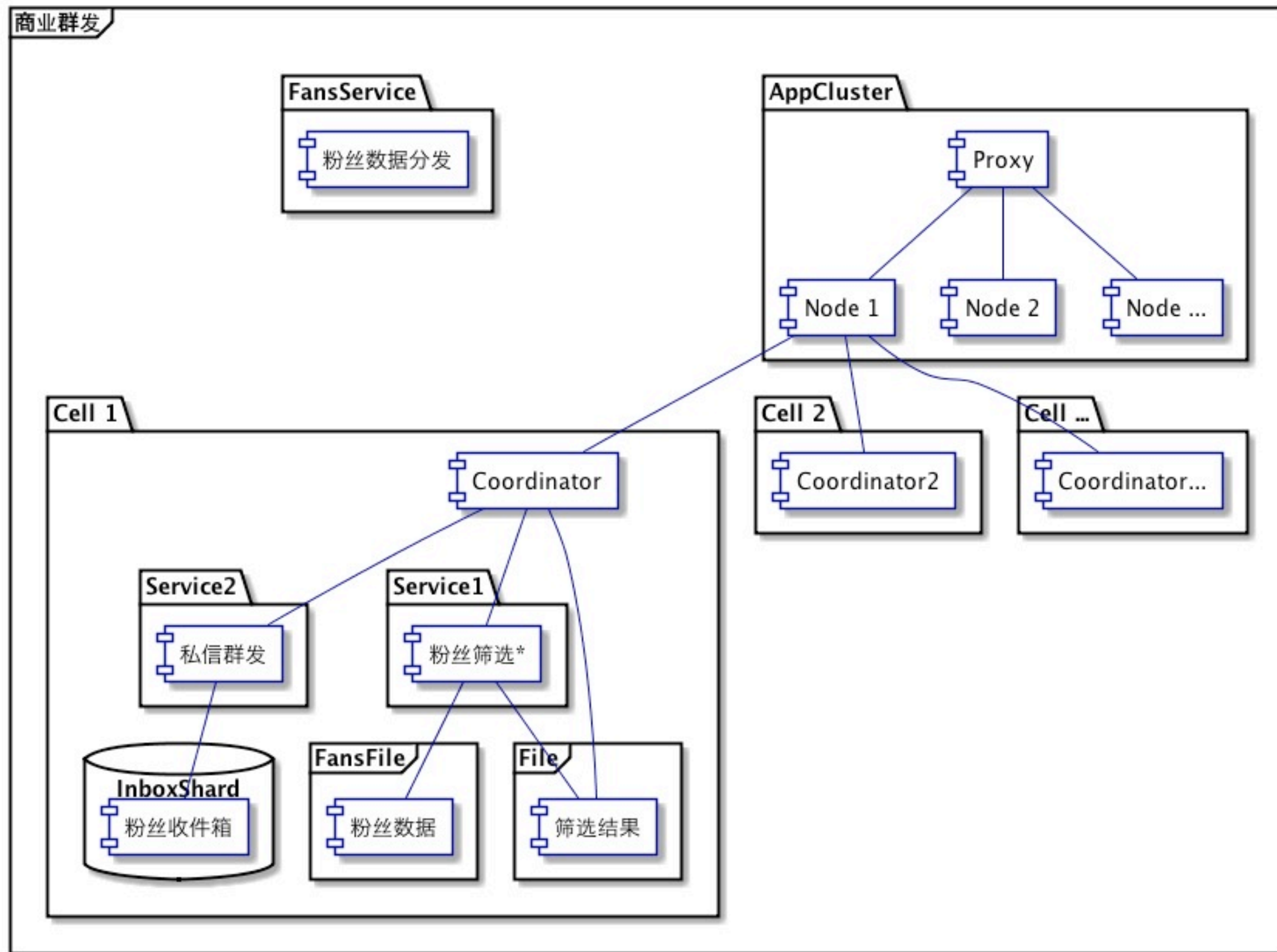
- 我们在做什么
- 单元化是什么
- 为什么单元化
- 怎么做到的
- 困难与解决



- 性能需求
 - 每秒百万级别
- 成本压力
 - 水平扩展 Scale out -> 垂直扩展 Scale up
 - 用复杂性来换吞吐

- 用户筛选
 - 读取粉丝，筛选用户特征，写筛选结果
- 消息群发
 - 写粉丝收件箱
- 发送计费、屏蔽检查、限流控制*





- 筛选服务（百万粉丝为例）
 - 获取粉丝关系
 - 筛选结果传输
- 数据库查询 vs 文件读取
 - 数据库资源 vs 本地缓存 = 10/s vs ~50/s
- 远程网络 vs 本地磁盘
 - 带宽：8M vs 0
 - 延时：启动和传输，400ms vs 0

- 消息群发
 - 写粉丝收件箱
- MySQL vs Redis
 - 队列+处理机 vs 0
- Redis 远程 vs 本地
 - QPS : 80K/s vs 100K/s
 - 带宽 : 20M vs 0

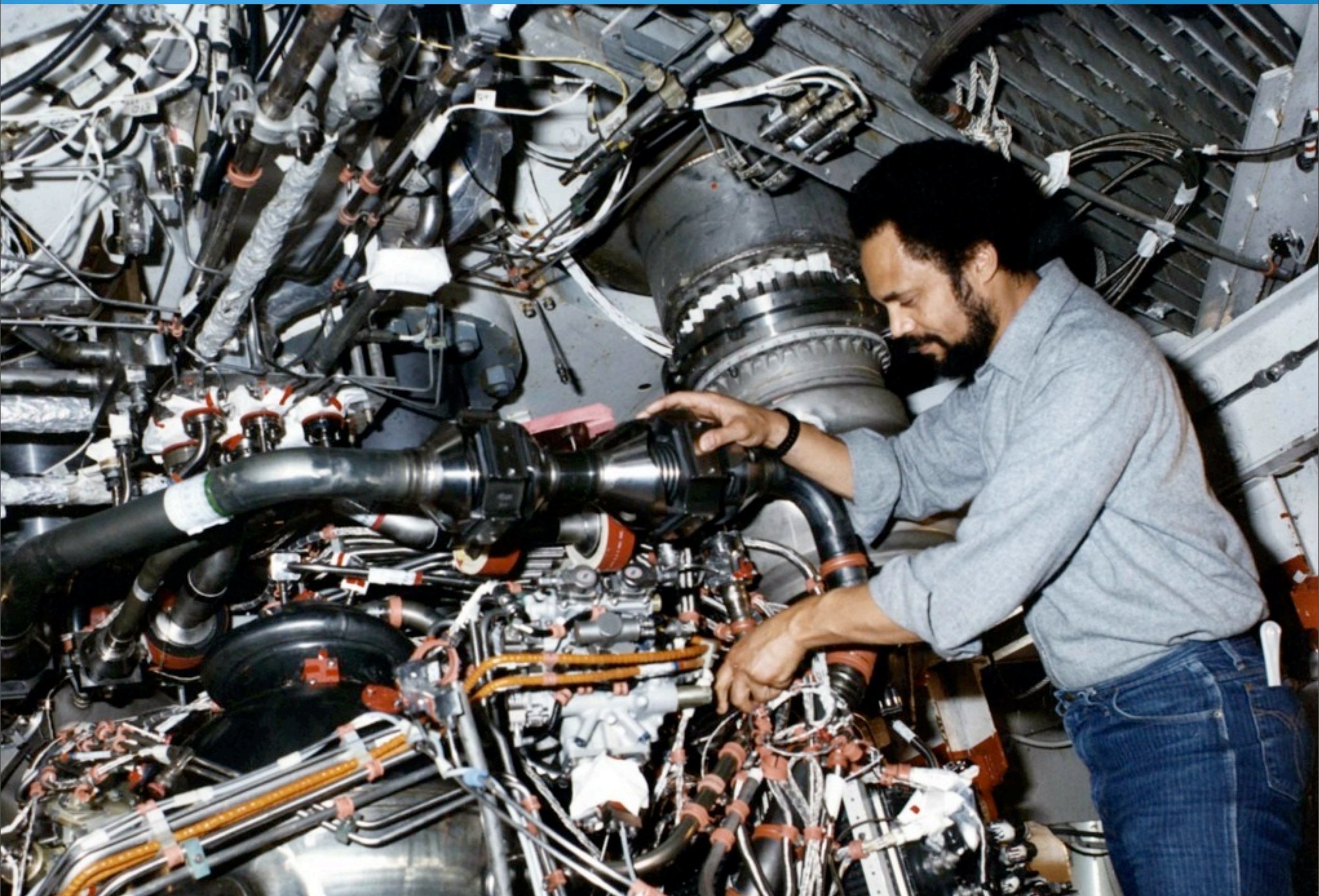
更少的机器1/2
更高的性能100

集成部署 + 数据本地化

- 请求隔离
 - Cell之间
- 独立升级
 - 灰度发布
- 扩容方便
- 资源隔离

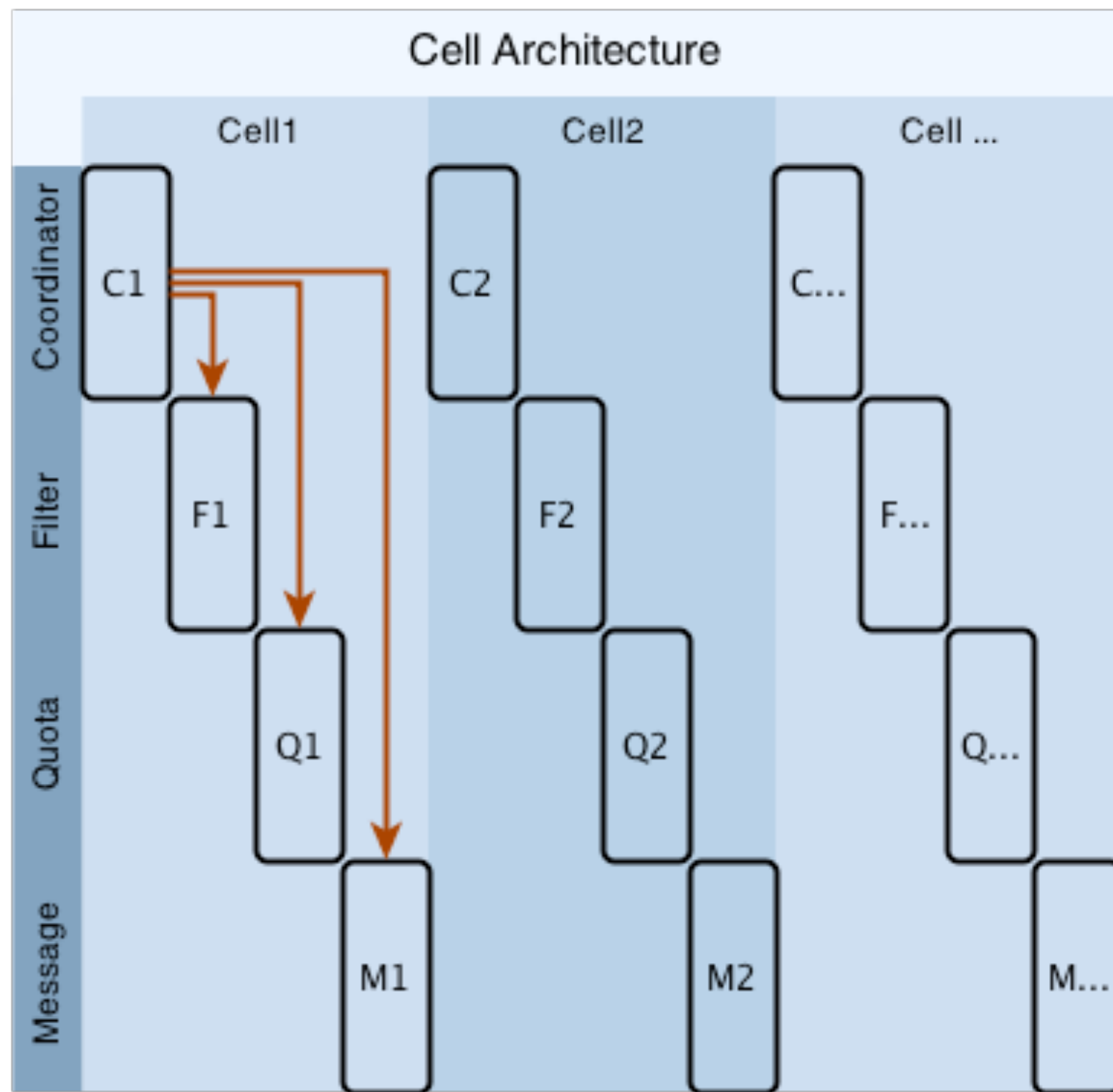
- 我们在做什么
- 单元化是什么
- 为什么单元化
- 怎么做到的
- 困难与解决

怎么做到的



- 彻底单元化
 - 从前端到存储
- 分区方式
 - 按接收者ID分区
 - 不是所有的服务都需要分区
 - 不是所有的数据都需要分区

- 分层服务
 - 相同数量节点
 - 相同分区算法
- 前端访问指定下层节点

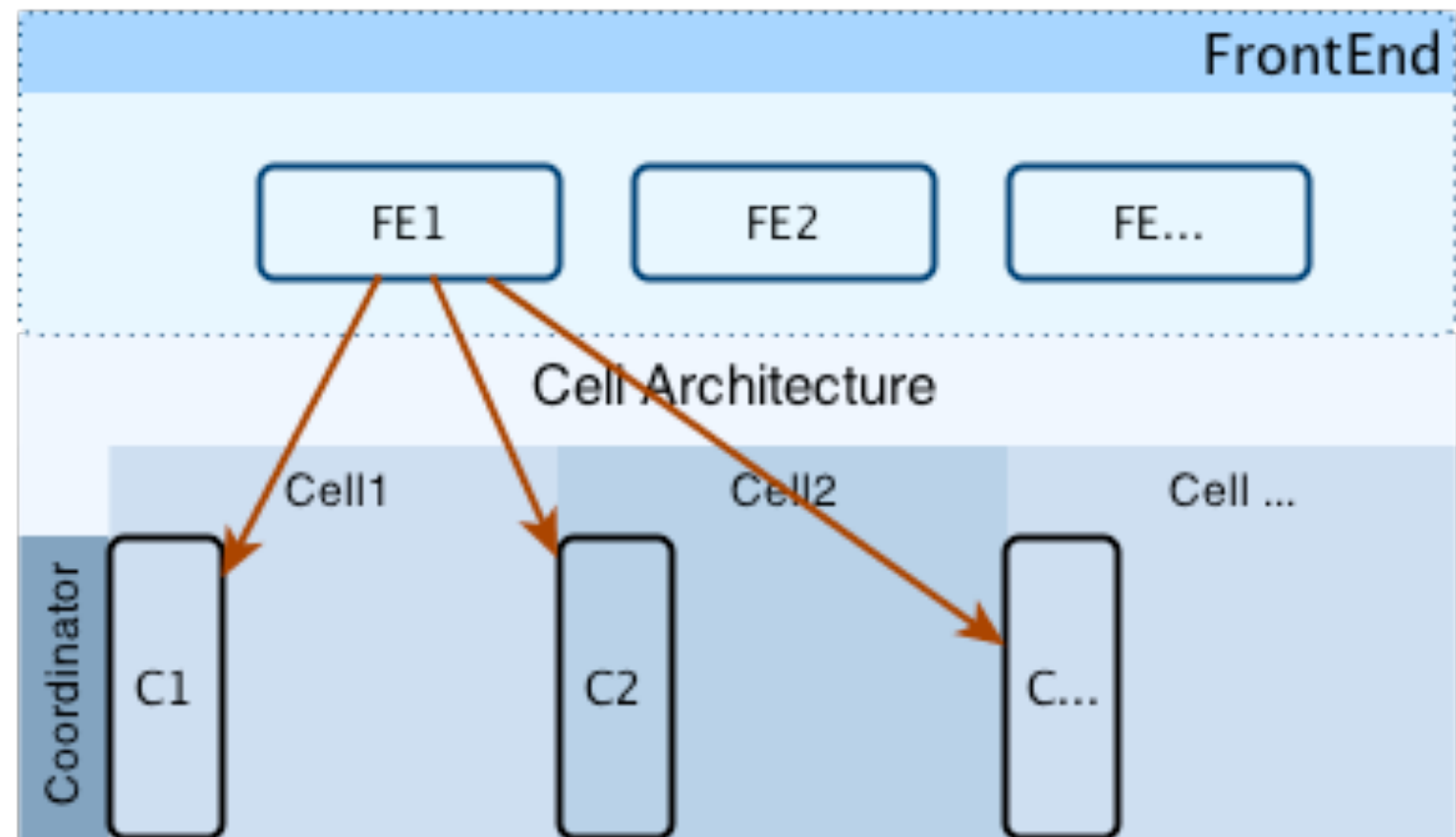


- 作业管理

- 请求分发
- 结果收集
- 超时控制
- ! Scatter-Gather

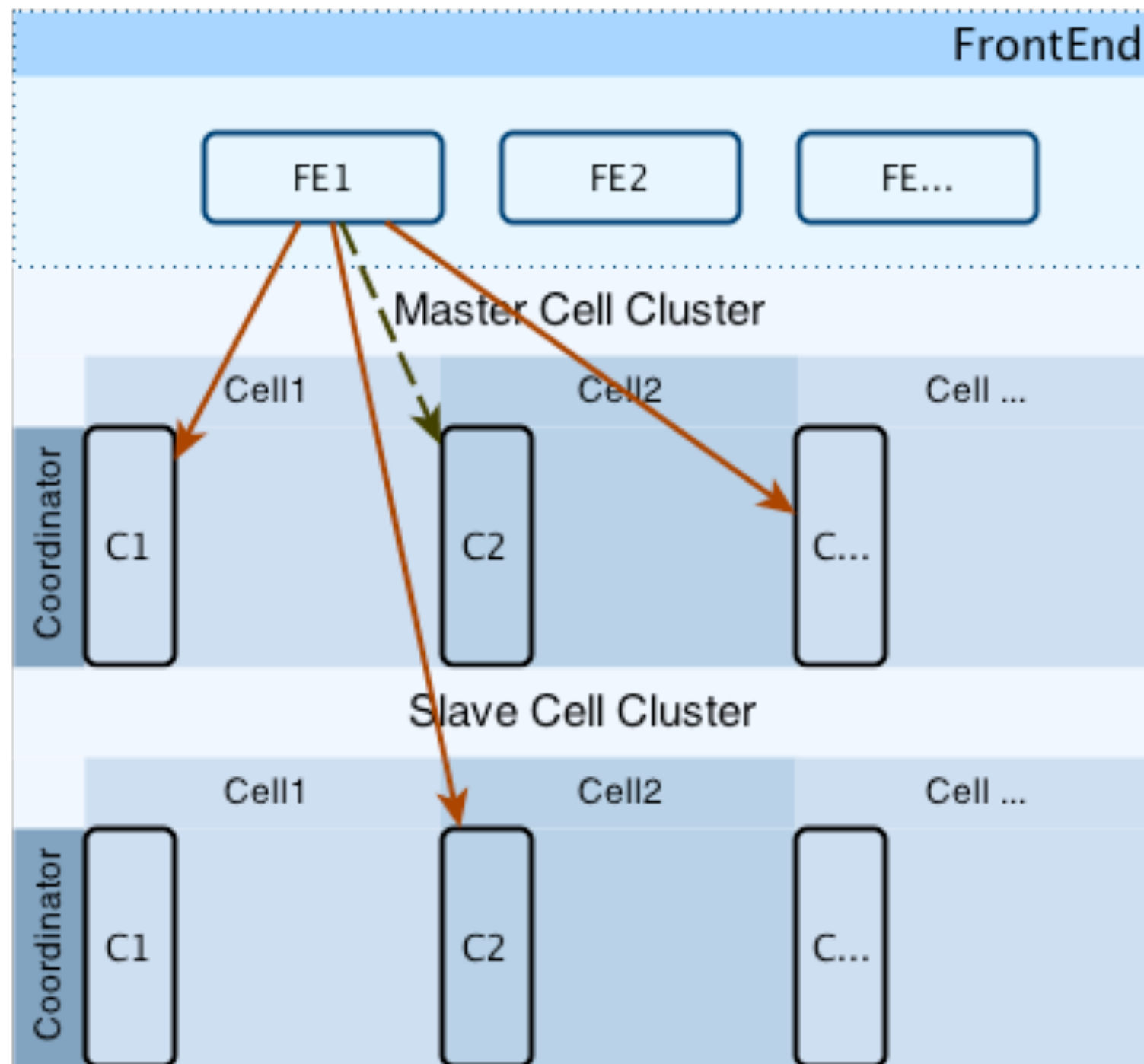
- CQRS

- fire-and-forget

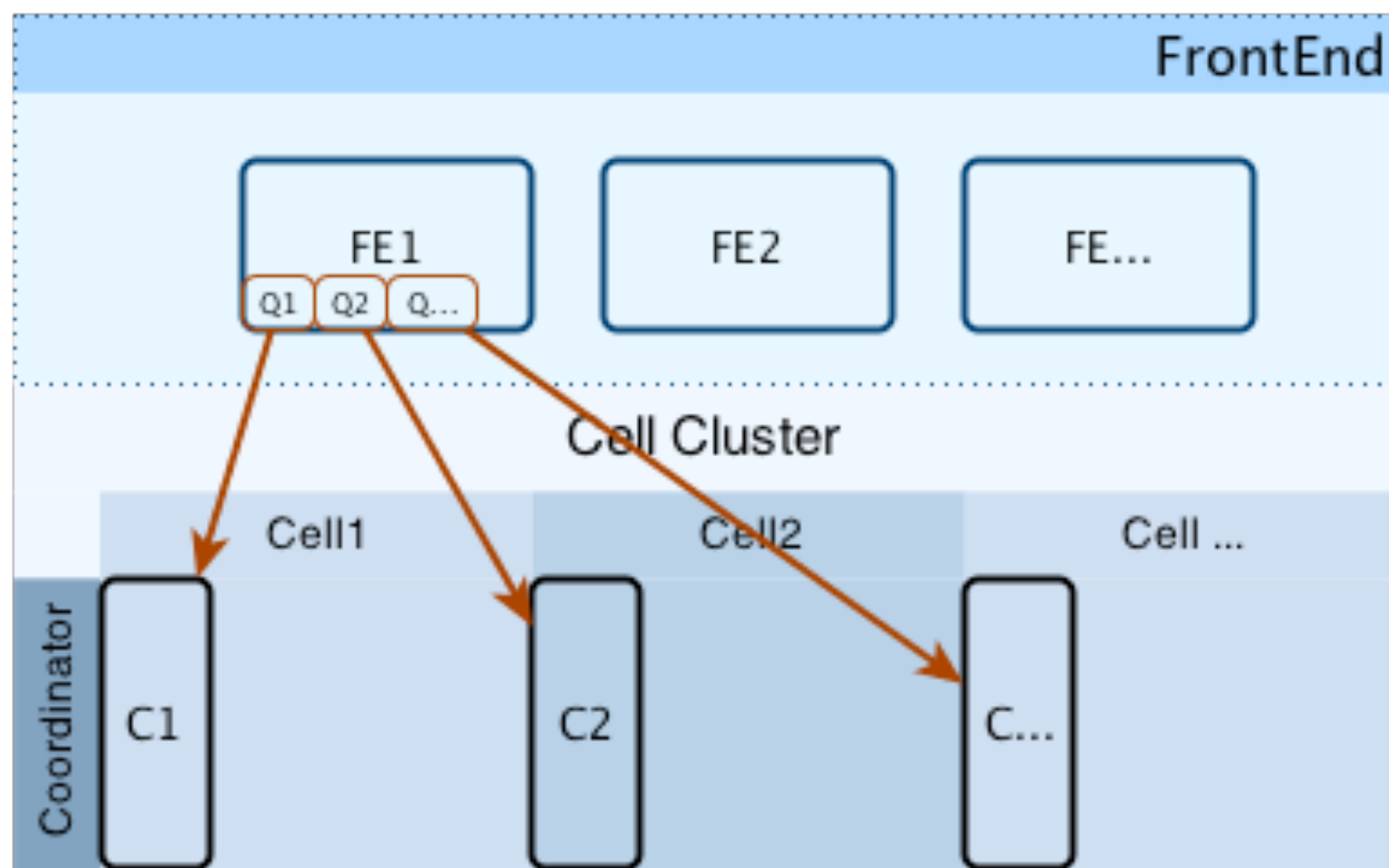


- 容错设计

- 公共资源故障
- 关键资源故障
- Cell切换



- 如何升级
 - 作业重做
 - 持久化队列



- 精细化控制
 - 资源规划：CPU、内存、硬盘
 - CPU绑定
 - 多硬盘设计
 - 主从读写分离
- 如何伸缩
 - Cell多实例

- 我们在做什么
- 单元化是什么
- 为什么单元化
- 怎么做到的
- 困难与解决



- 遇到失败怎么办
- 问题解决四步
 - 发现问题？
 - 定位原因 &
 - 修复数据 *
 - 重做任务！

- 工具集Muses

- Clio

- 日志审计
- 故障报警
- 每日报表

- Uranian

- 检测与修复



- 如何重做任务
- 作业需要满足
 - 幂等性
 - 可重入

Move Fast and Break Things

- 服务如何运维
- 异构服务
 - 资源监控：主从切换
 - 多团队：权限管理
 - 服务配合：Dump操作

单元化架构 一次值得的尝试

谢谢

@一乐

@微博平台架构

以微博之力 让世界更美！
weibo.com