



SDCC 2016

中国软件开发者大会

SOFTWARE DEVELOPER CONFERENCE CHINA

一统三国：

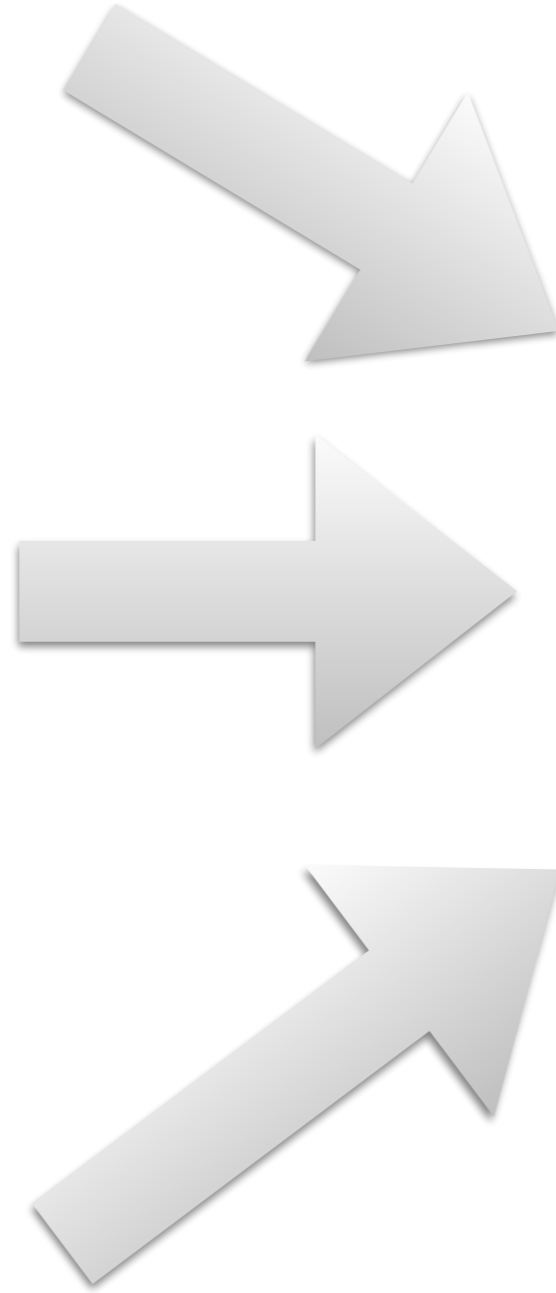
阅文集团内容系统架构整合

大嘴

xuhaifeng@yuewen.com

现状

SDCC 2016
中国软件开发者大会
SOFTWARE DEVELOPER CONFERENCE CHINA



腾讯文学: java + mysql

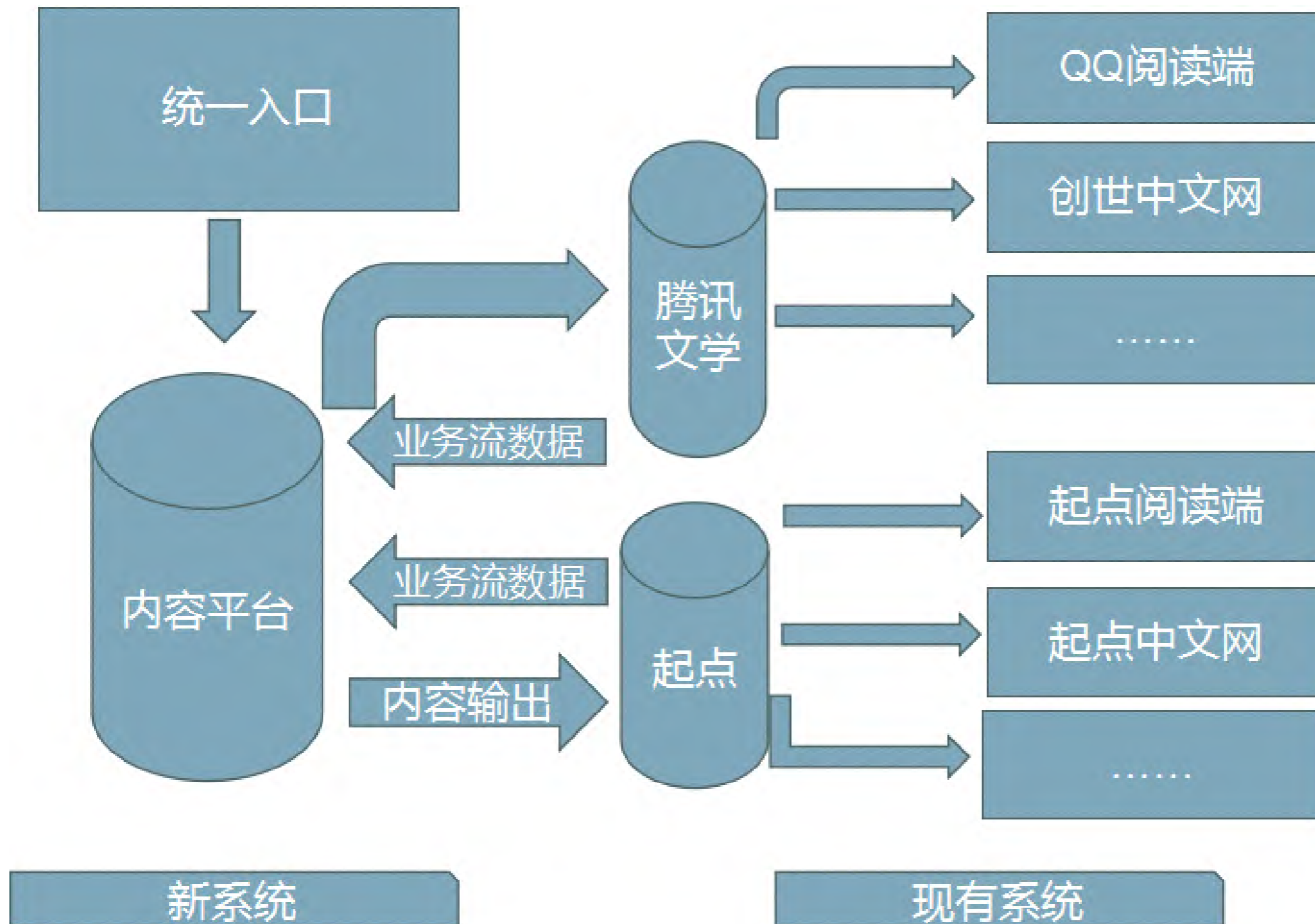
创世: java + php + mysql

盛大文学: c# + mysql + oracle
+ sqlserver

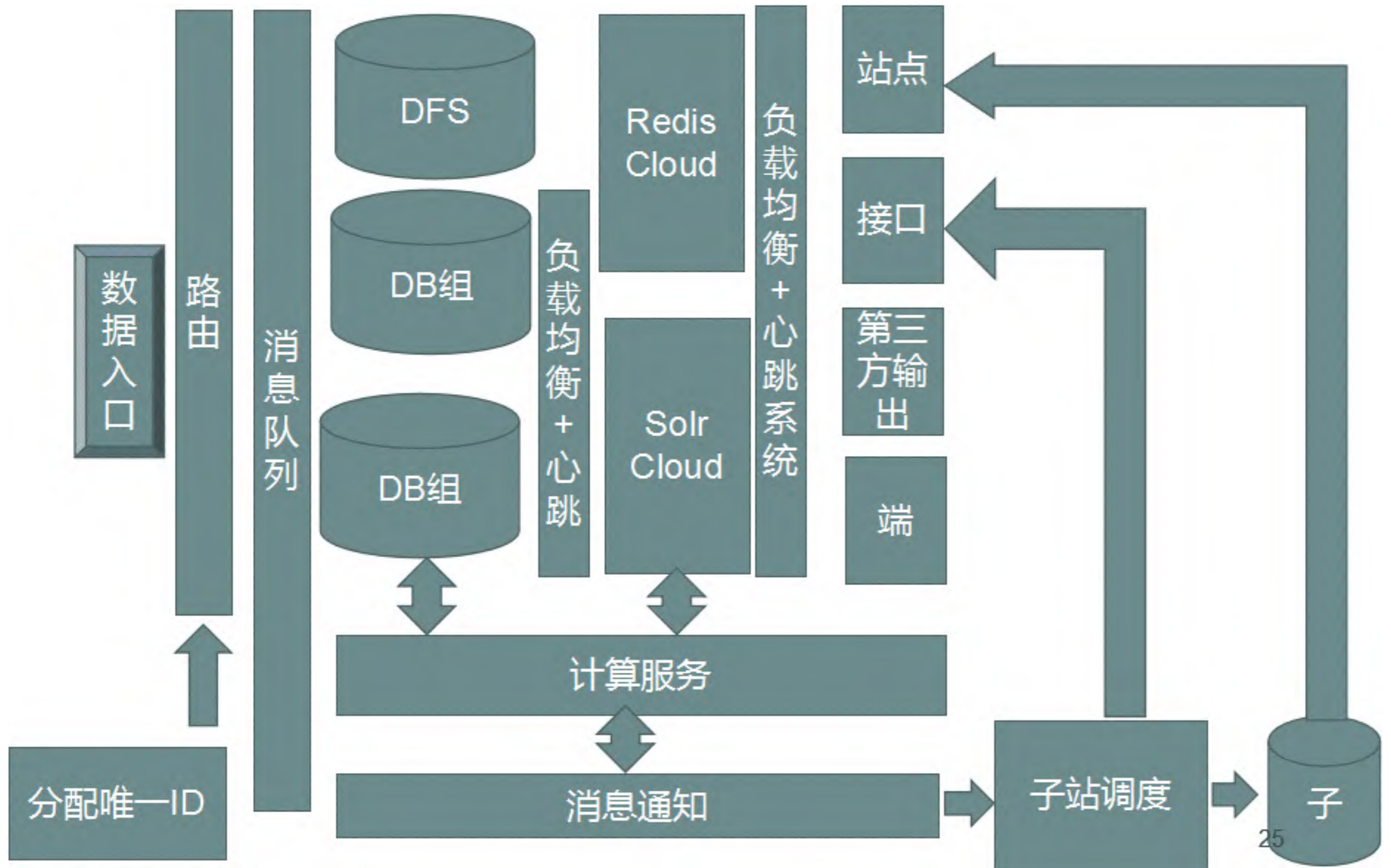
创立内容中心

- 书：X百万自主 + X百万合作商
- 章节：X亿自主 + X亿合作商
- 电子书：mp3（低音质、高音质）
- 流水：打赏、月票、推荐票……
- 作者：定时发布、草稿、日志（无法估量）
- 稿酬计算

业务架构



技术架构



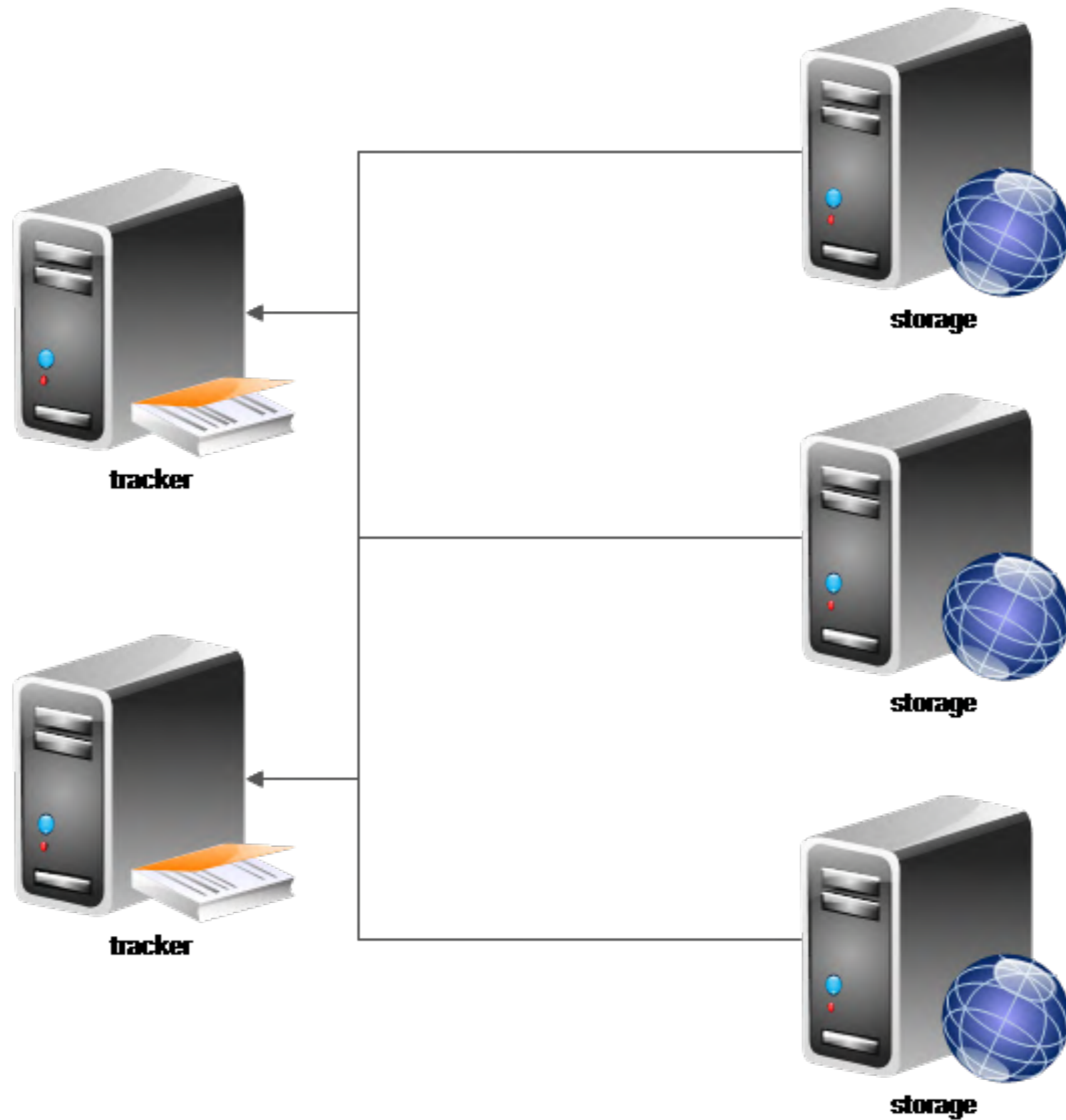
- 应用程序：java开发框架（自研）
- 队列服务：完成双向消息通知（自研）
- 调度服务：完成定时计算等逻辑（自研）
- ID服务：根据模块编号提供唯一ID（自研）
- DFS：分布式内容存储（自研）
- MySql：基本数据存储，采用主从结构
- Redis：缓存
- Solr：索引
- Nginx服务：提供http服务及负载均衡

- 横向扩展
- 自动化：数据平衡、数据恢复
- 尽可能让每台机器提供服务：多主模式，没有slaver
- 支持频繁的更改
- 数据一致性

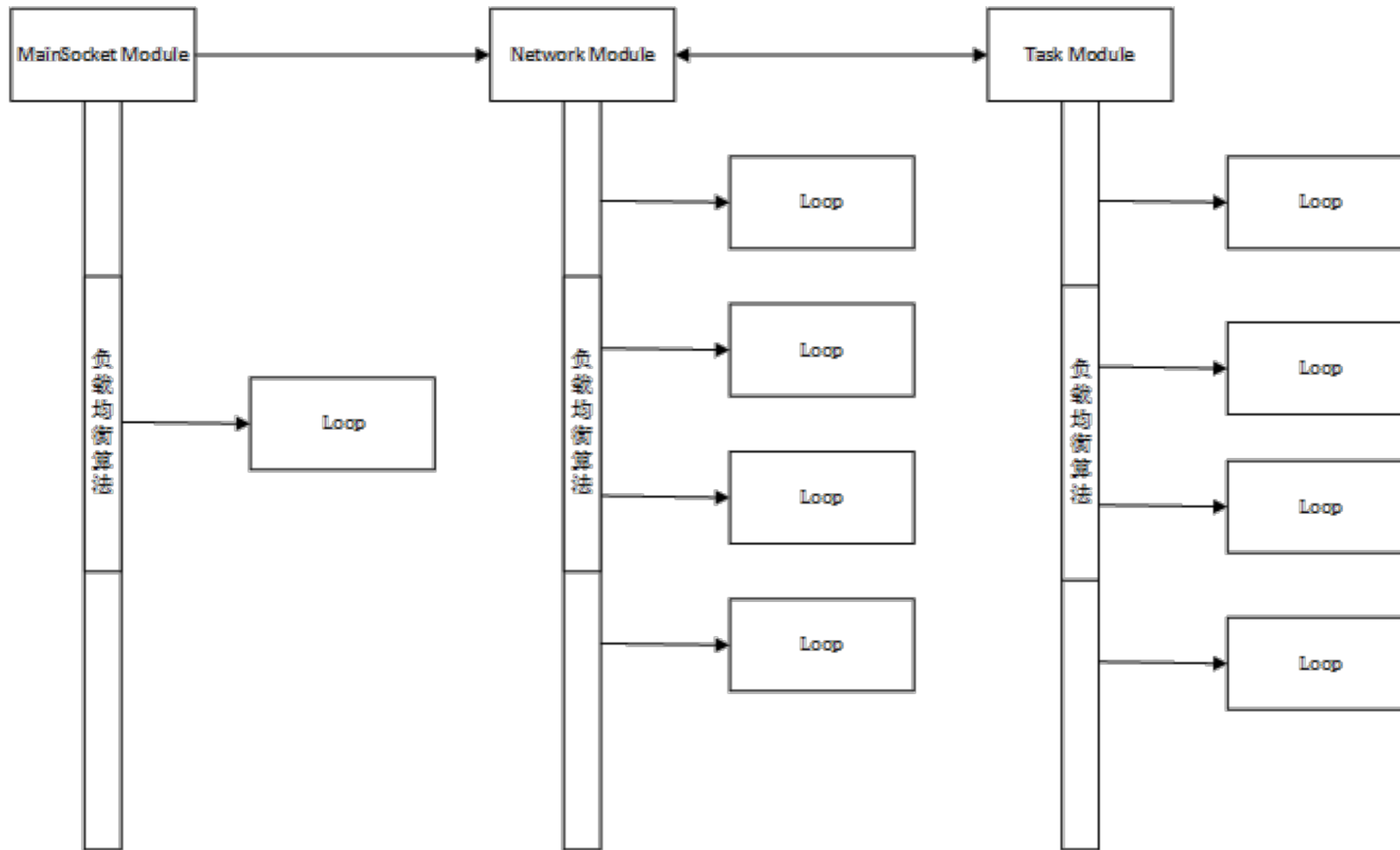
- Fastdfs :
 - chunkfile的版本控制不满意
 - 同group镜像粒度太大
- CFS (腾讯内部) :
 - 小文件就是灾难
 - 挂载运行, 没有管理层
- 剩下的 :
 - 就可以不看了

但是还有一个纠结的: 都TM对更改支持的不友好

DFS架构



进程逻辑图



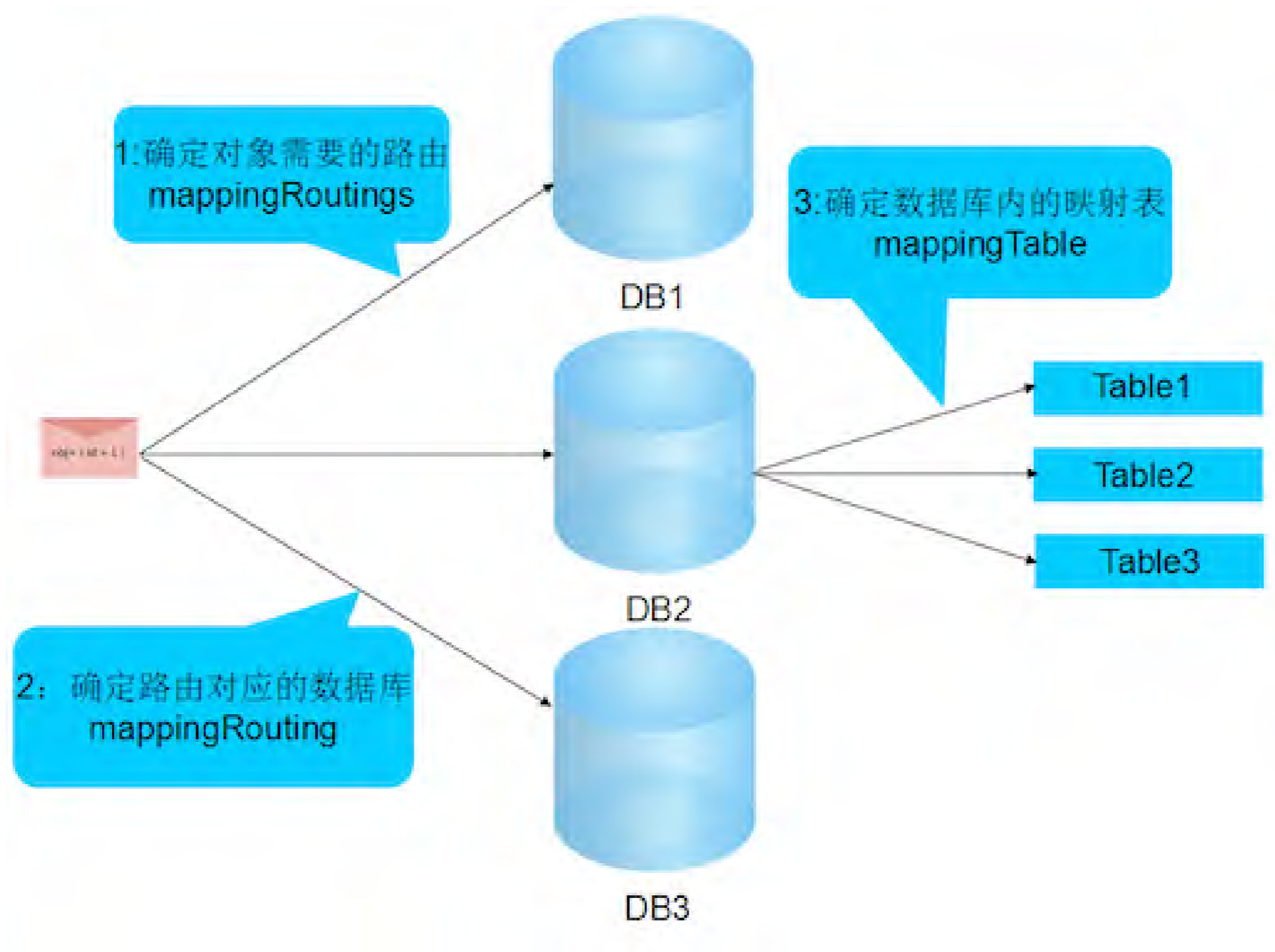
- 清楚的逻辑分层：职责清楚
- 线程调度算法（无锁编程算法）

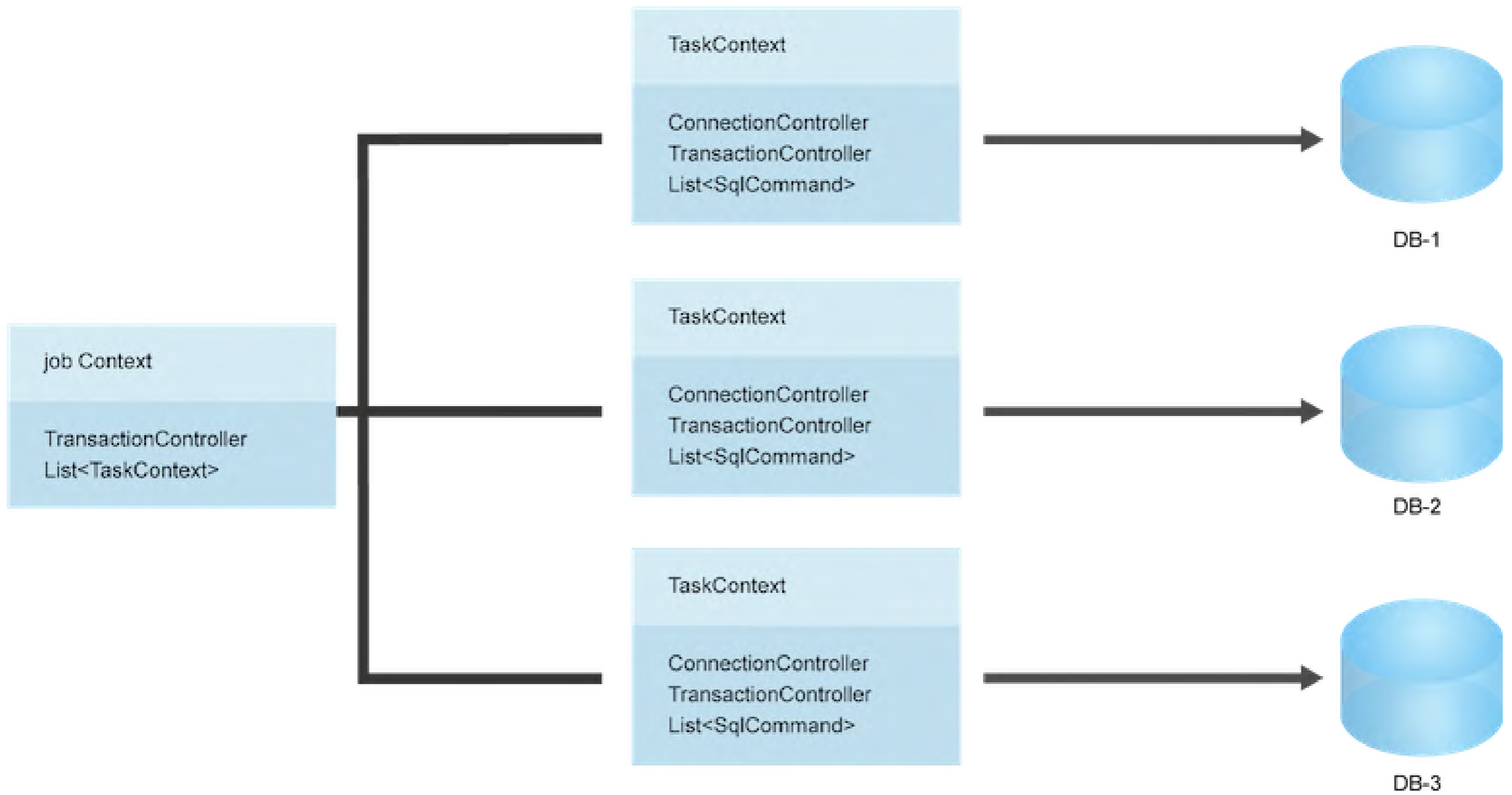
- 时间向量算法控制MVC
- 实际长度 * 120 % = 总长度
- hashcode控制签名

0		4		8
	是否删除		后缀名	
	操作版本		服务器版本	
	创建时间			
	最后更新时间			
	总长度			
	实际使用长度			
	Hashcode			
	保留长度			

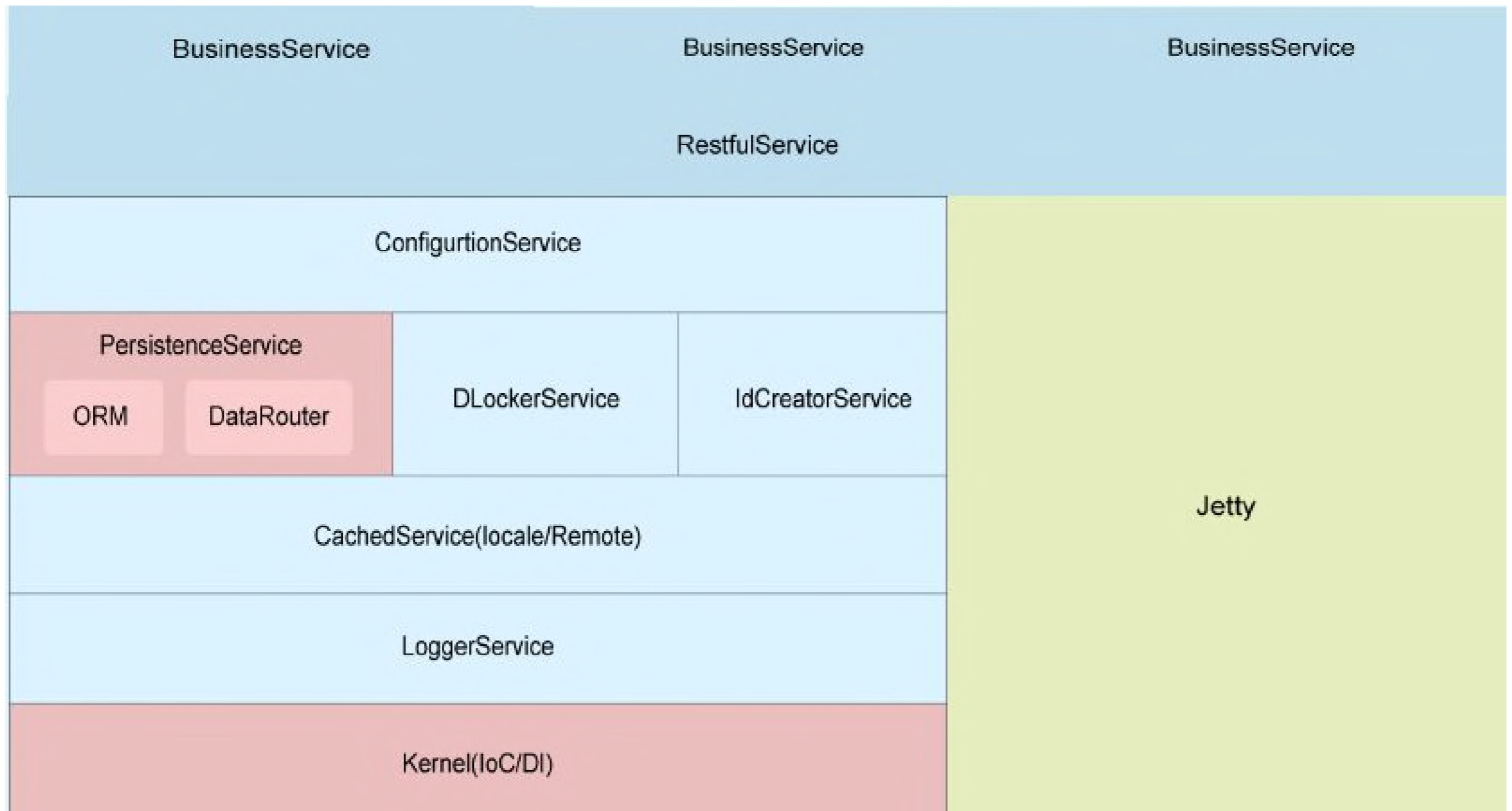
章节信息—数据路由

路由过程





编程框架系统



路由标识-id生成器

- int型, 自增
- int自增, 步长自定义
- GUID / UUID
- 自定义String

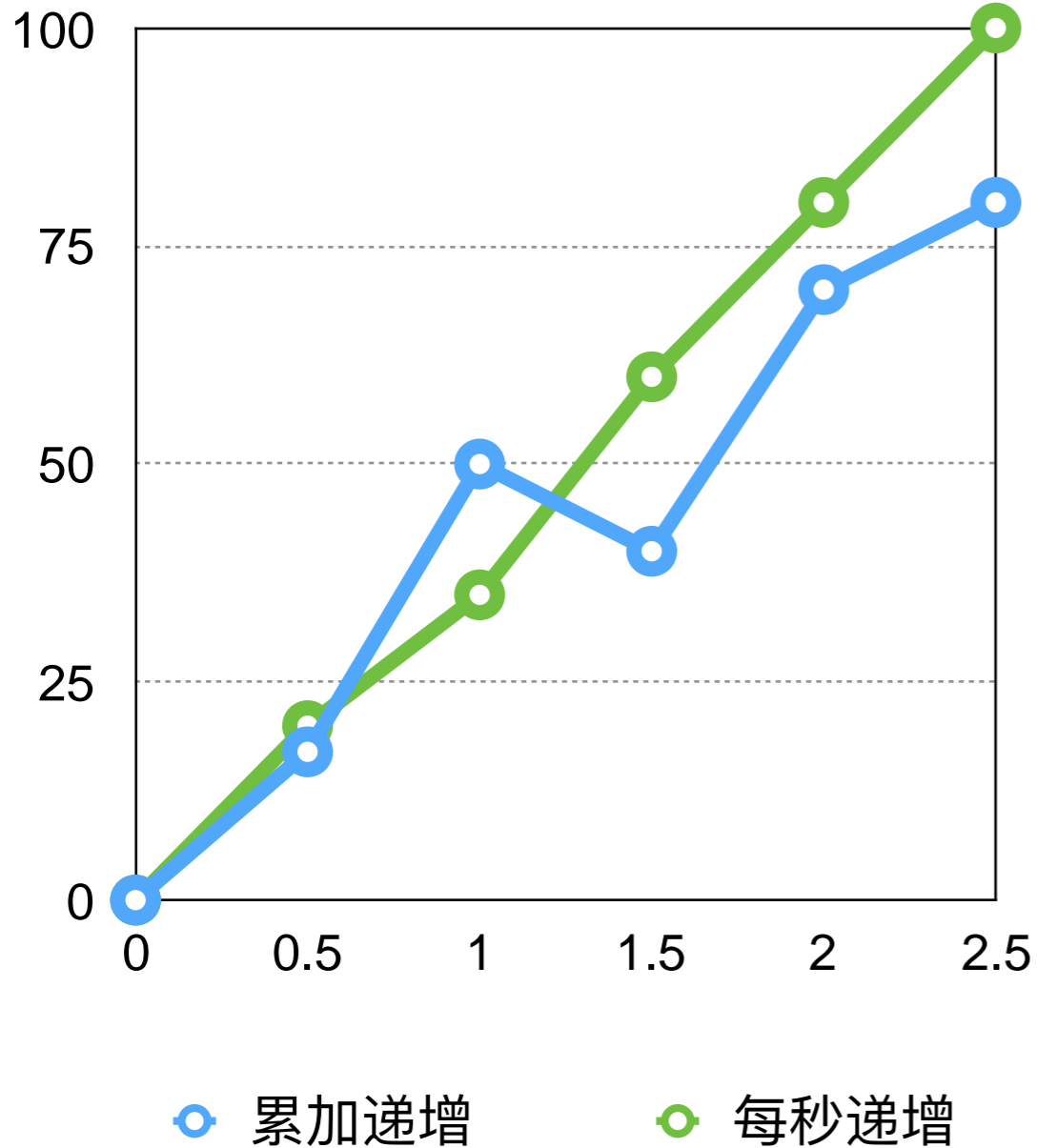
- 唯一，必须唯一
- 短，尽可能的短
- 生成速度足够快
- 运算足够简单，快速
- 附带实体业务信息，比如时间、类型等
- 部分信息可以自定义，比如路由信息
- 不仅机器能识别，人类也可以识别
- 对索引友好
- 根据业务规则，能自定义排序等业务规则

snowflake算法—twitter

- 0xFFFFFFFF, uint64的最大值
- 组成：时间戳-机器位-随机数
- 41位 + 10位 + 12位
- 9223 3720 3257 7650 688
- 0111 1111 1111 1111 1111 1111 1111 1111
- 0000 0001 0001 0000 0100 0000 0000 0000
- 2199023251472-264-0

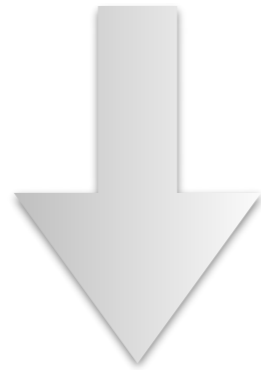
- 1844 6744 0737 0955 1656 , uint64的最大值
- 4294967295-0000-01-1-00
- 算法：时间戳-随机数位-类型位-机器位-库位
- 具体位数：10位 + 4位 + 2位 + 1位 + 2位
- 为什么会比最大值少了一位？

id生成趋势

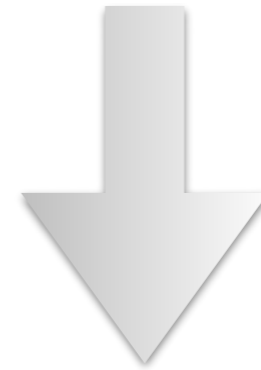


- 累加递增：长时间（2s内）内保证单调递增，短时间（1s内）内不保证单调递增
- 每秒递增：它肯定是递增的，因为每秒都会从0开始，单位时间内都是单调递增

按照时间分



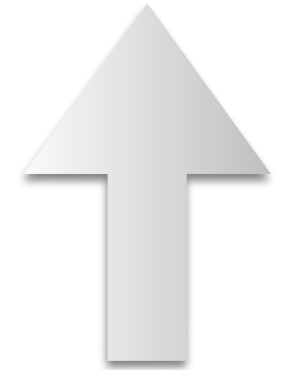
类型



4294967295-0000-01-1-00



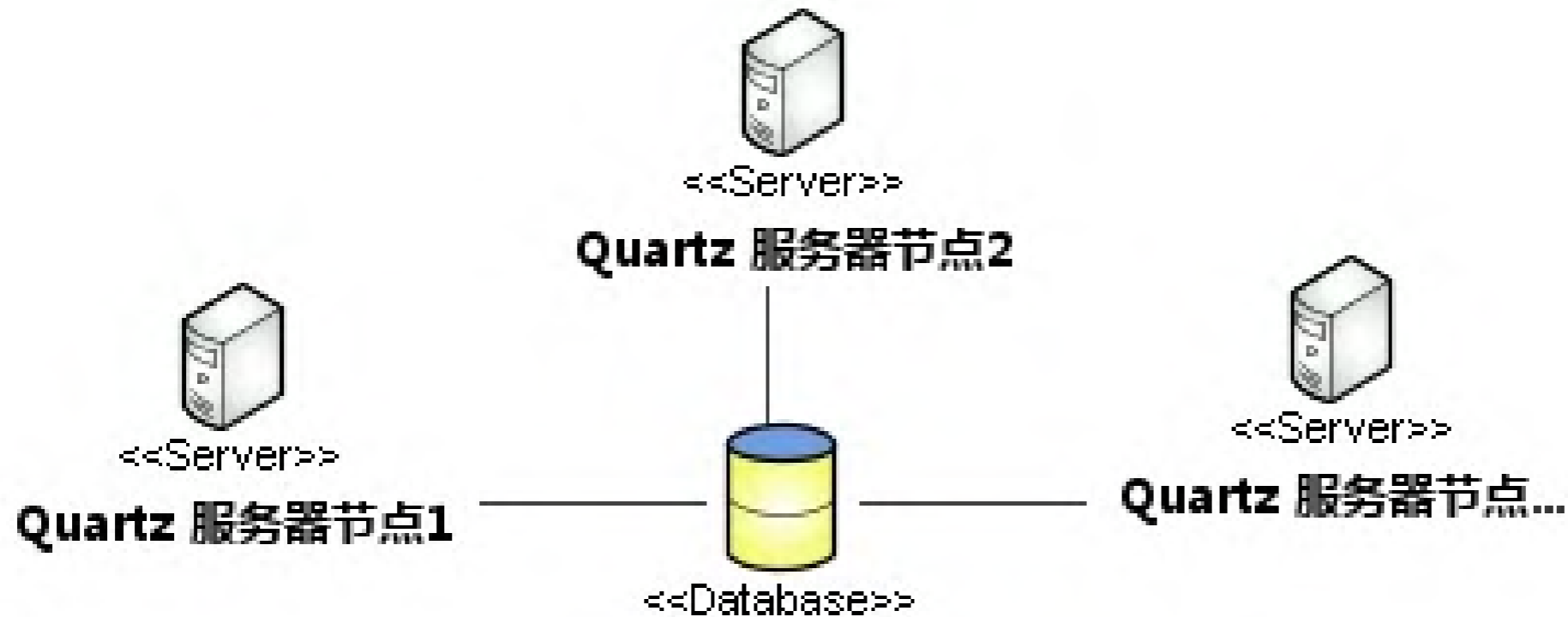
分表位



分库位

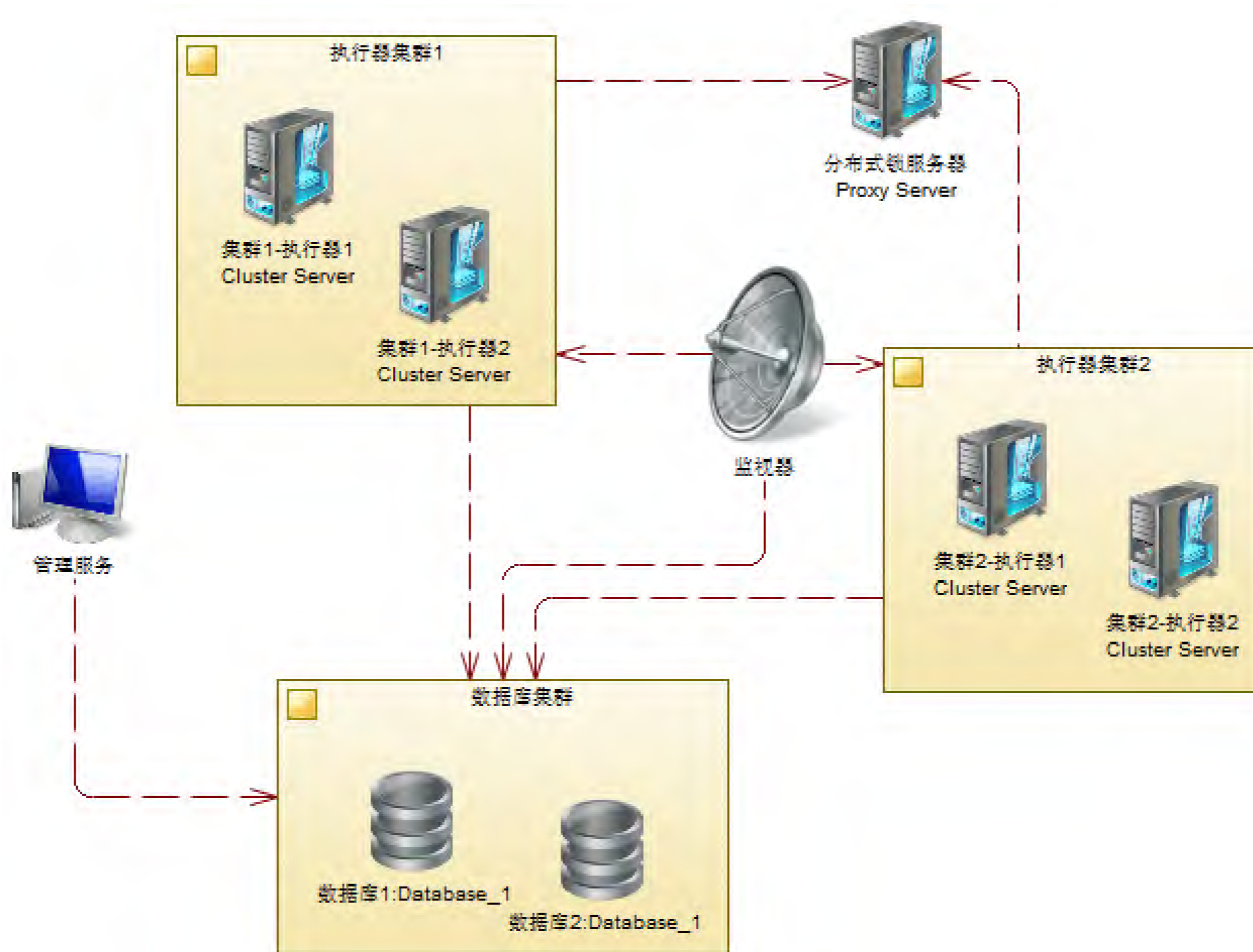
任务调度

V1-quartz



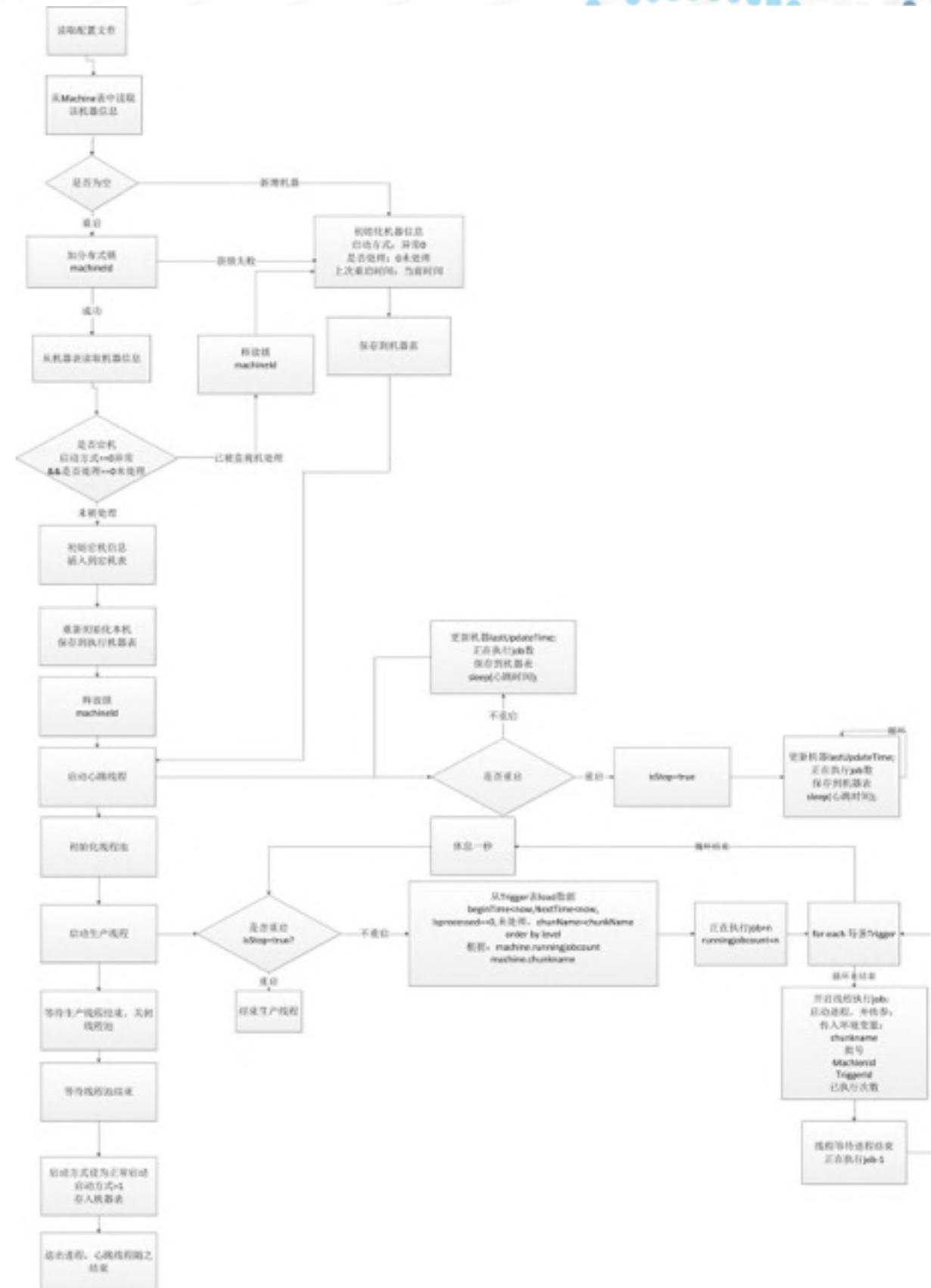
- 改进为进程模式
- 严重依赖数据库
- 自定义功能性扩展真的太难了

V2-自主实现



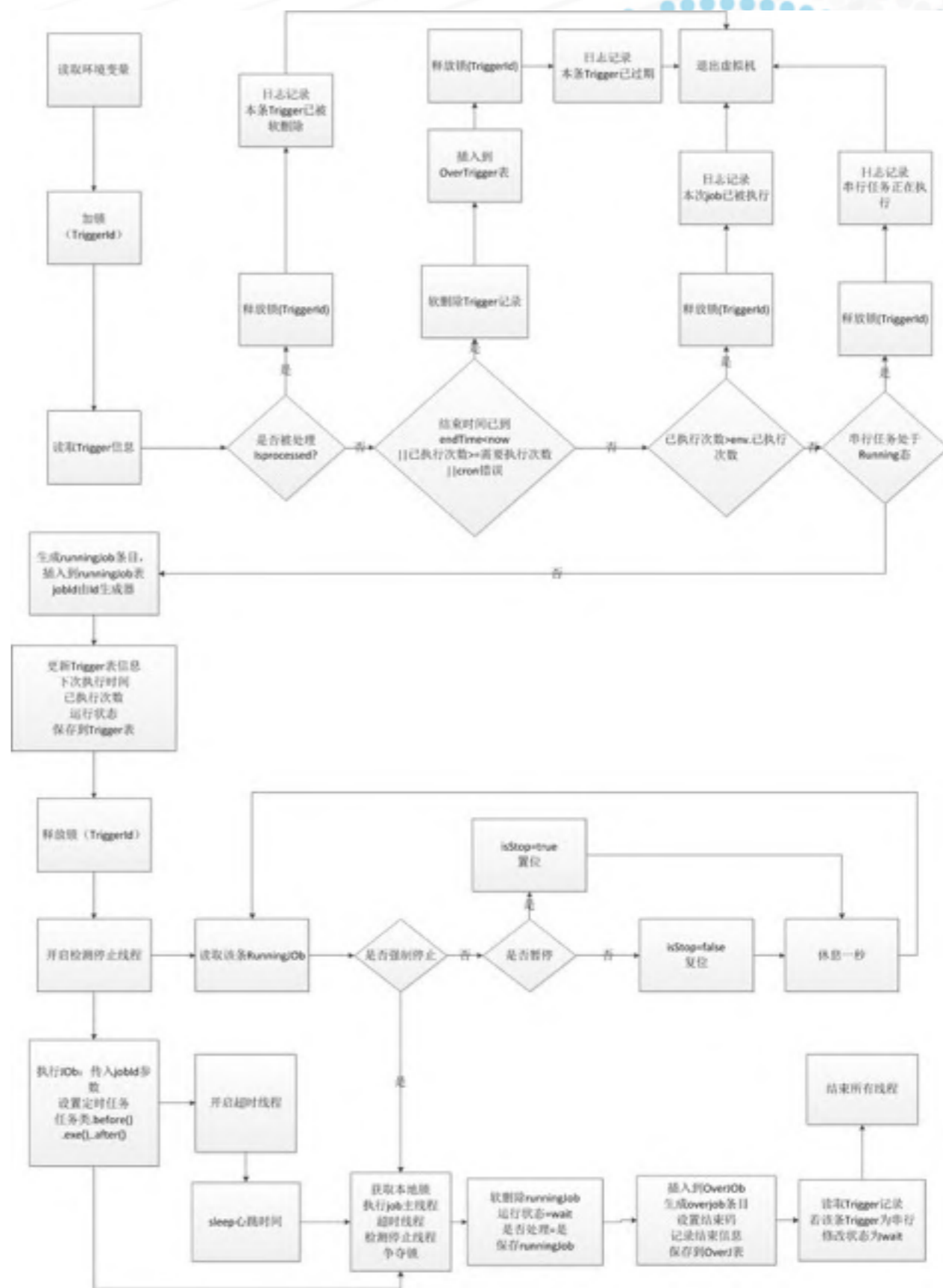
任务调度器 - scher

- batchid, 每次生命周期的批号, 判断是否被重启
- scher db, 获取需要需要启动的job
- 线程池 -> 进程池
- 执行container

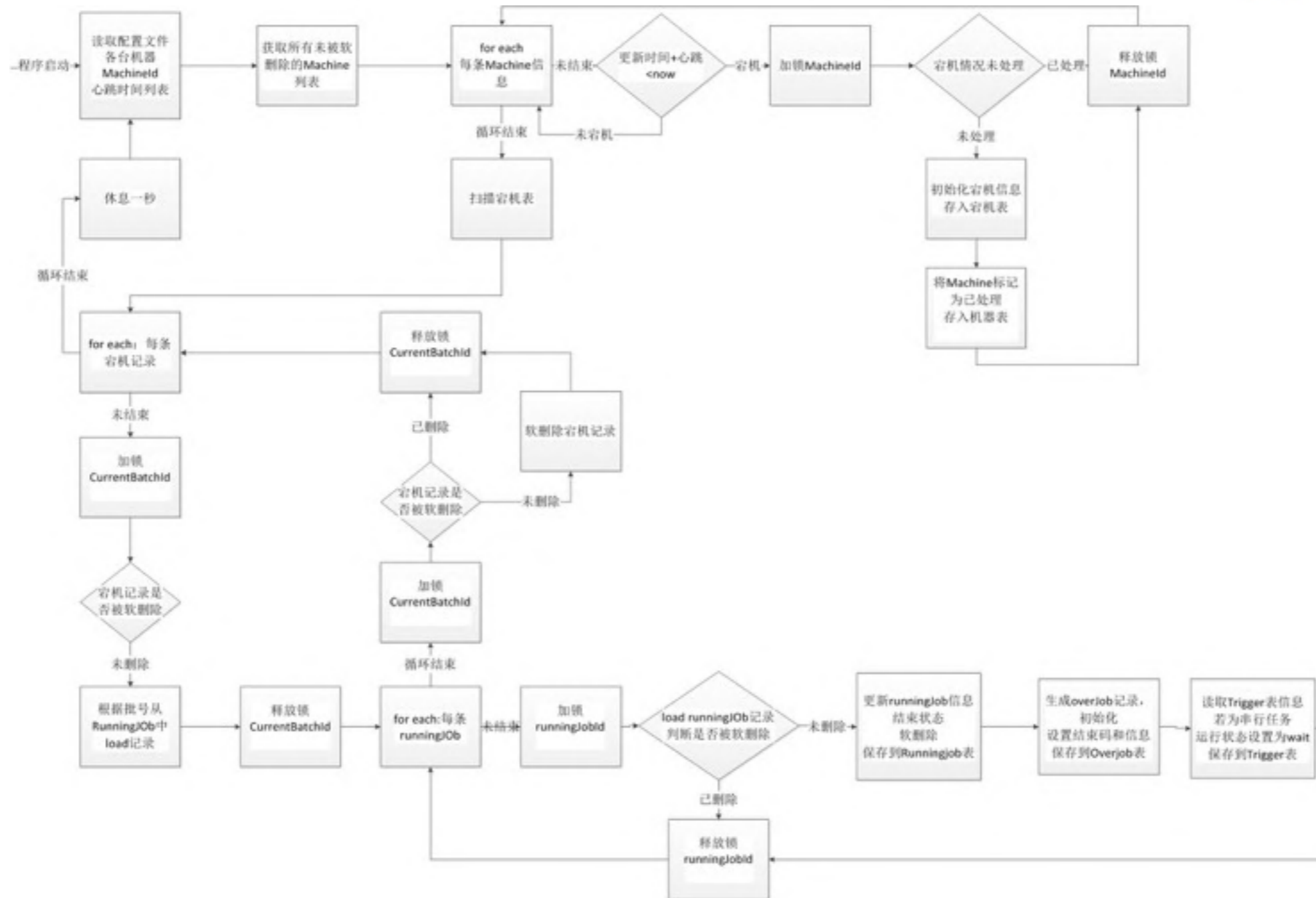


任务调度器 - container

- 每个job的包装类
- job分为可并行和不可并行
- job可超时, 超时直接kill自己



任务调度器 - arbiter



- 获取被kill的scher
- 清理正在执行的job



SDCC 2016

中国软件开发者大会

SOFTWARE DEVELOPER CONFERENCE CHINA

谢谢！

