1. Redis复习总结
   1. 缓存击穿、雪崩、穿透等场景及解决
   2. 布隆过滤器及实现(布隆过滤器需要预先知道取值空间？ 核心思想:hash碰撞的概率小)
   3. 分布式事务、分布式锁 -redis 相关的 lua、multi机制
   4. 分布式协调 -> raft协议

Pubsub没有保障消息投递机制(at most once)

Redis slot

1. 流式应用总结

Apache beam 模型

Event time

Processing time

Watermark

Trigger

Window

Flink: stateful :

Flink fault tolerance:

State backend;

State清理：ttl， timeService, state partition?

Keyd state

Operator state:

Operator State (or non-keyed state) is state that is is bound to one parallel operator instance. The [Kafka Connector](https://ci.apache.org/projects/flink/flink-docs-release-1.12/dev/connectors/kafka.html) is a good motivating example for the use of Operator State in Flink. Each parallel instance of the Kafka consumer maintains a map of topic partitions and offsets as its Operator State.

The Operator State interfaces support redistributing state among parallel operator instances when the parallelism is changed. There are different schemes for doing this redistribution.

In a typical stateful Flink Application you don’t need operators state. It is mostly a special type of state that is used in source/sink implementations and scenarios where you don’t have a key by which state can be partitioned.

Broadcast state

Exactly-once

两阶段提交

Flink 架构 flink on yarn yarn

Jobmanager、slot、resourcemanager、

在failover恢复后 timer注册的fire时间已过，会如何处理。

Fault tolerance: checkpoint、state如何恢复， checkpoint何时如何触发？能否由此保证exactly-once和最终一致性。

Kafka+ flink: partition与parallelism的关系？

Flink parallelism?? Task chain , operator parallelism? What is task??? JobGraph ExecutionGraph

Checkoutpoint barrier。。

Handling late data

反压

* 1. 大数据 shuffle??? Data skew ？ flatmap

1. Spring boot
2. Kafka
3. 设计模式
   1. 观察模式与发布订阅
   2. **代理模式：典型应用skywalking 接口埋点增强**
   3. 责任链



1. Lru spark rdd

工作项目总结

1. 客服坐席系统

系统结构： 用户前端->中台-》问答引擎 || 坐席系统

会话管理：流程图和核心模块讲清楚

redis事务处理： 分布式资源竞争 + redis分布式锁：redLock(备选).

主从、哨兵模式事务： watch, multi, exec. 实现原理

集群模式事务：lua脚本，

持久化: lua脚本中的随机数持久化与其他的区别： aof,通常是命令，而随机数命令应当将值持久化 否则将造成从持久化数据恢复后不一致的现象

Skywalking总结

Skywaling 架构， 整体数据处理流程

Skywalking instrumentation agent 反射 增强 代理 plugin pluginfinder spi

Open tracing： TraceContext, traceId的生成， span。 链路还原

Google Dapper

Lambda 架构

全链路监控总结

业务监控流总结

1. 语音导航监控 ：Kafka +spark streaming + hbase = 数据落库;
   1. Spark streaming 利用hbase rowid幂等写入去重 来实现exactly-once语义
2. CMS运营监控： kafka + flink +es 监控。 业务数据dashboard
3. CMS现场监控: kafka + flink + es

图：

BFS(BreadthFirstSearch) ： FIFO队列， 结点分类， 未发现(WHITE), 已发现并添加至待搜索队列(GREY), 已发现并已完成其相邻结点搜索(BLACK)

DFS(DepthFirstSearch): 回溯法