

scala & storm 下流式计算

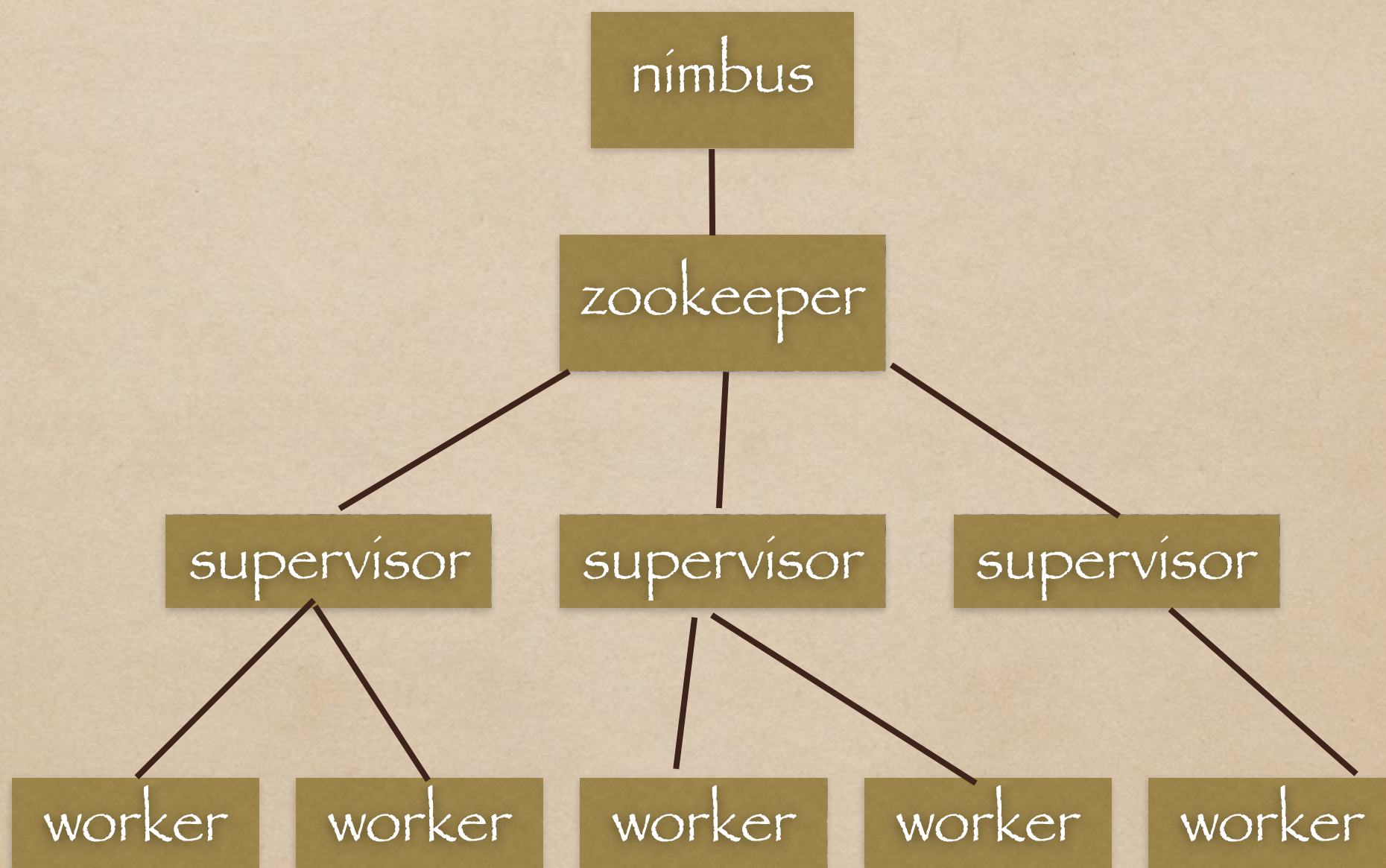
何首乌@挖财

目录

- ◆ storm 简介

- ◆ 实战

storm构成



容錯

集群的容错

- ◆ worker死亡
- ◆ supervisor宕机
- ◆ nimbus挂了

可靠的消息传递机制

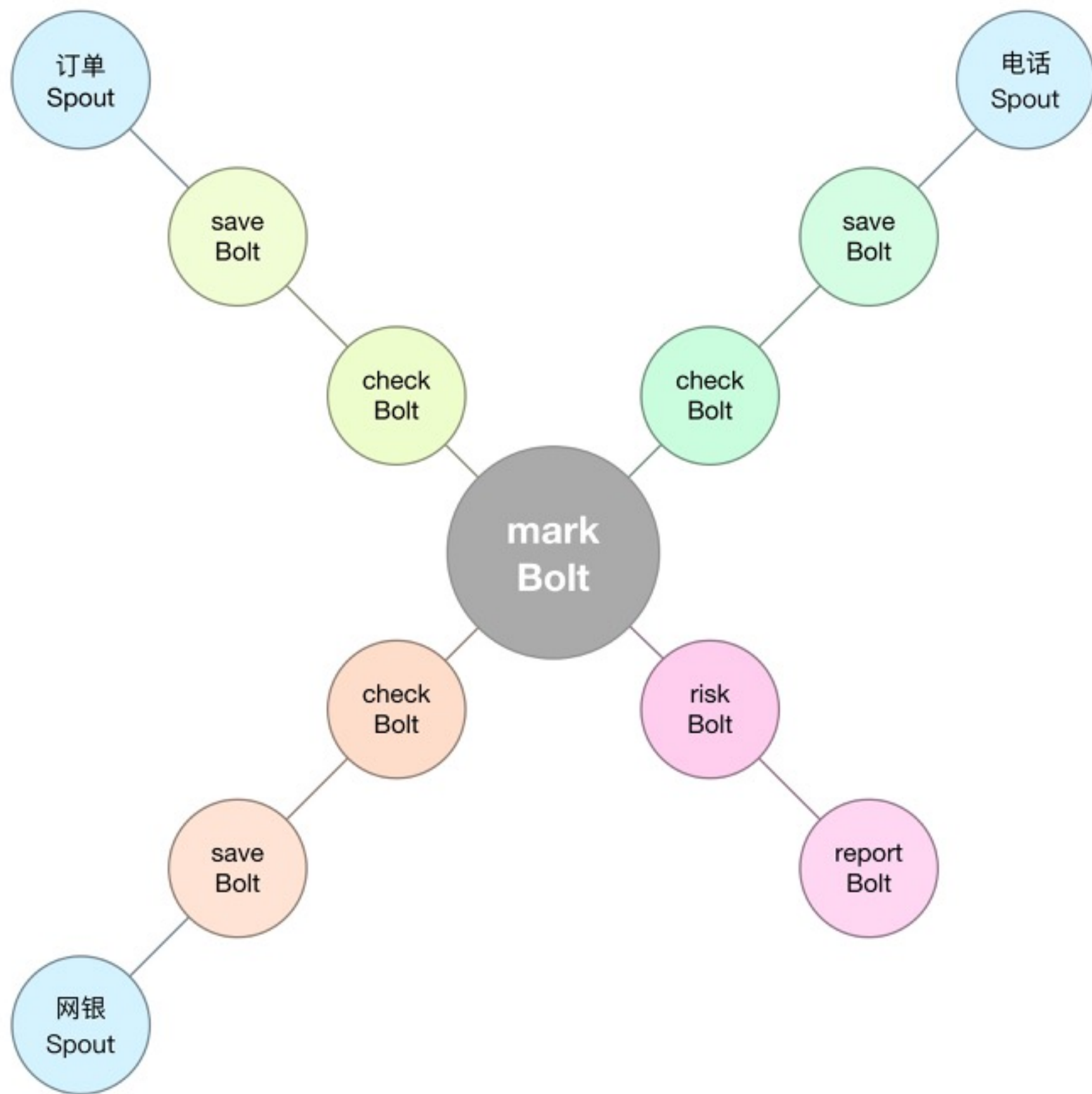
- ◆ 确保消息都被消费
- ◆ 失败的消息会进行重试

消息分发

- ◆ shuffle
- ◆ LocalAndShuffle
- ◆ fieldGroup
- ◆ CustomGroup

实战

- ◆ kafka
- ◆ storm
- ◆ codis
- ◆ 主要的逻辑使用scala编写



- ◆ storm 搭建起处理流程
- ◆ scala 实现具体逻辑

- ◆ 优点
- ◆ 从设计模式中解脱出来
- ◆ 专注于逻辑
- ◆ 更优雅更简洁



Scala

```
def products = orders.flatMap(o => o.products)
```

```
public List<Product>getProducts() {  
    List<Product> products = new ArrayList<Product>();  
    for (Order order : orders) {  
        products.addAll(order.getProducts());  
    }  
    return products;  
}
```




```
1 val ls = (1, 2) :: (2, 1) :: (1, 3) :: Nil
2 ls.groupBy(_._1).map{ case (id, id2valueLs) =>
3   id -> id2valueLs.map(_._2).reduce(_ + _)
4 }
```


- ◆ 缺点
- ◆ 各种高级语法,隐转,不知所云
- ◆ 大量生成临时对象

项目感悟

- ◆ 上下文
- ◆ 测试
- ◆ 监控

