

Sugon 瑞翼教育

YARN

报告人：曙光瑞翼教育品牌部

课程目标

- ✓ 了解YARN出现的背景，YARN的基本概念
- ✓ 掌握YARN的组成部分，各组成部分的作用
- ✓ MapReduce1与YARN的区别
- ✓ 掌握YARN三种调度器的特点、工作方法

目录

01

YARN产生的背景

02

初识YARN

03

YARN架构

04

YARN调度器

05

小结



YARN产生背景

思考现实生活的例子

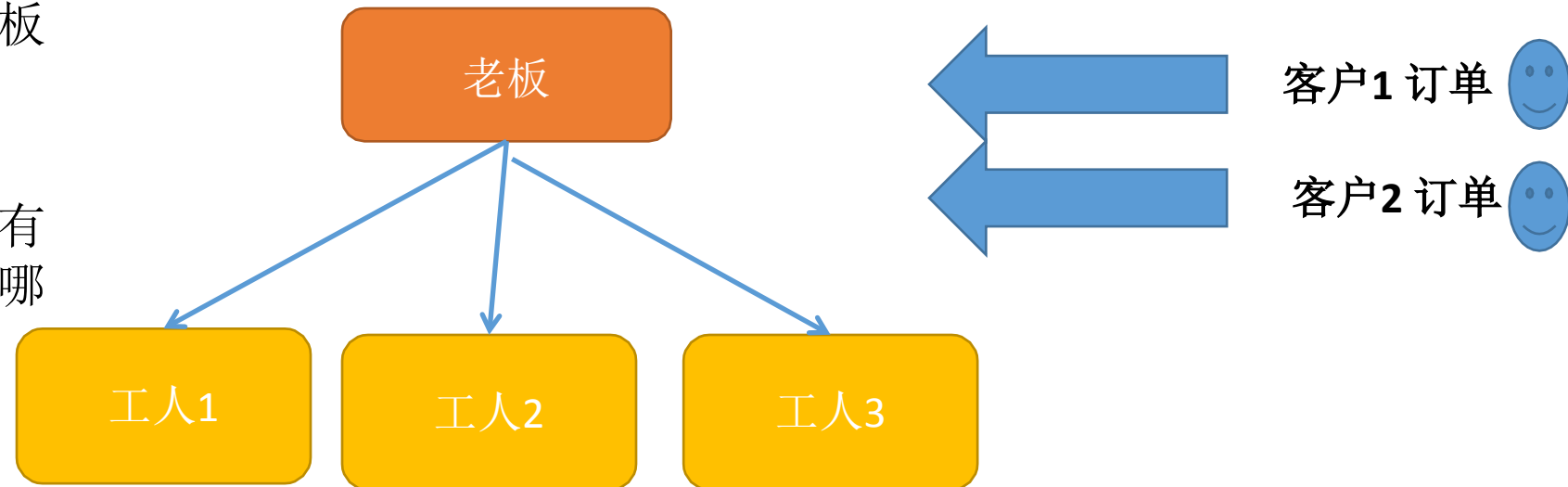
一个电子厂，刚开始是1老板+3个工人
思考：随着业务扩大，订单将增加，工人增多。
电子厂会遇到哪些问题？



问题一：
对于老板：订单进来，要分配任务。
工人增多，管理工作事情多。老板
很忙啊!!

问题二：订单进来，工人数量是有一个上限。如何安排协调资源，哪些订单先处理？

其他问题：



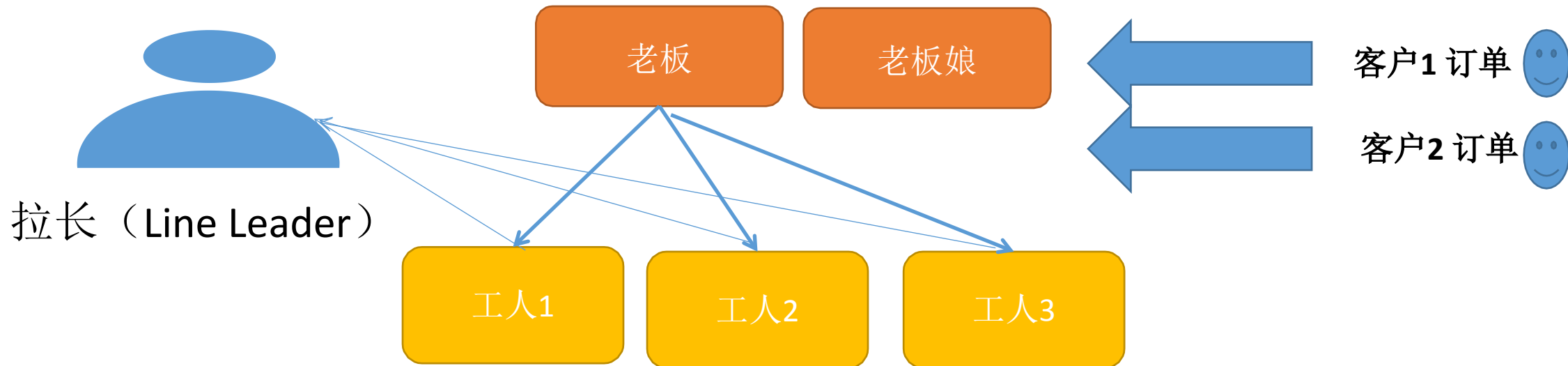
如何解决？

一个电子厂，刚开始是一个老板+3个工人
 随着业务扩大，订单将增加，工人增多。
 思考：电子厂会遇到哪些问题？



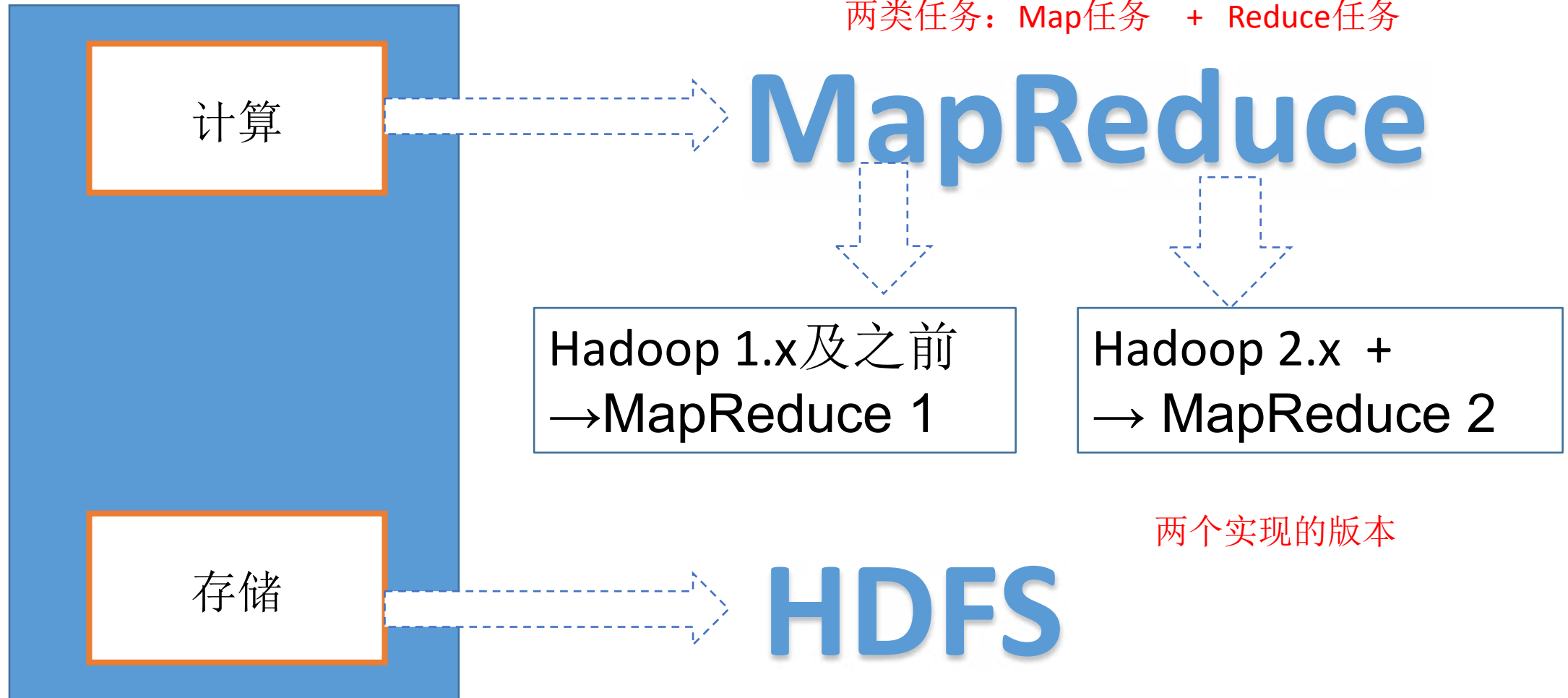
解决方法：

- (1)增加拉长，负责跟进进度
- (2)采用合适的调度器处理订单。



4.1.1 Hadoop主要组成

Hadoop主要的组成

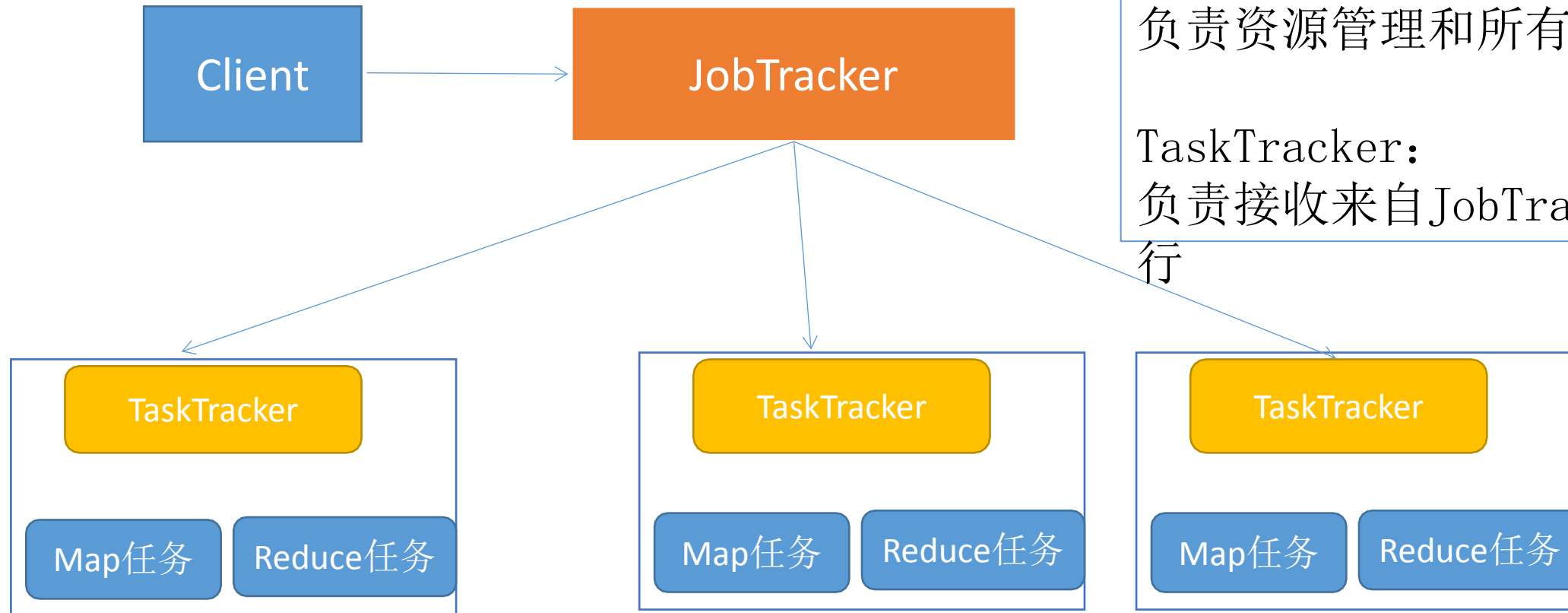


4.1.2 MapReduce 1 架构

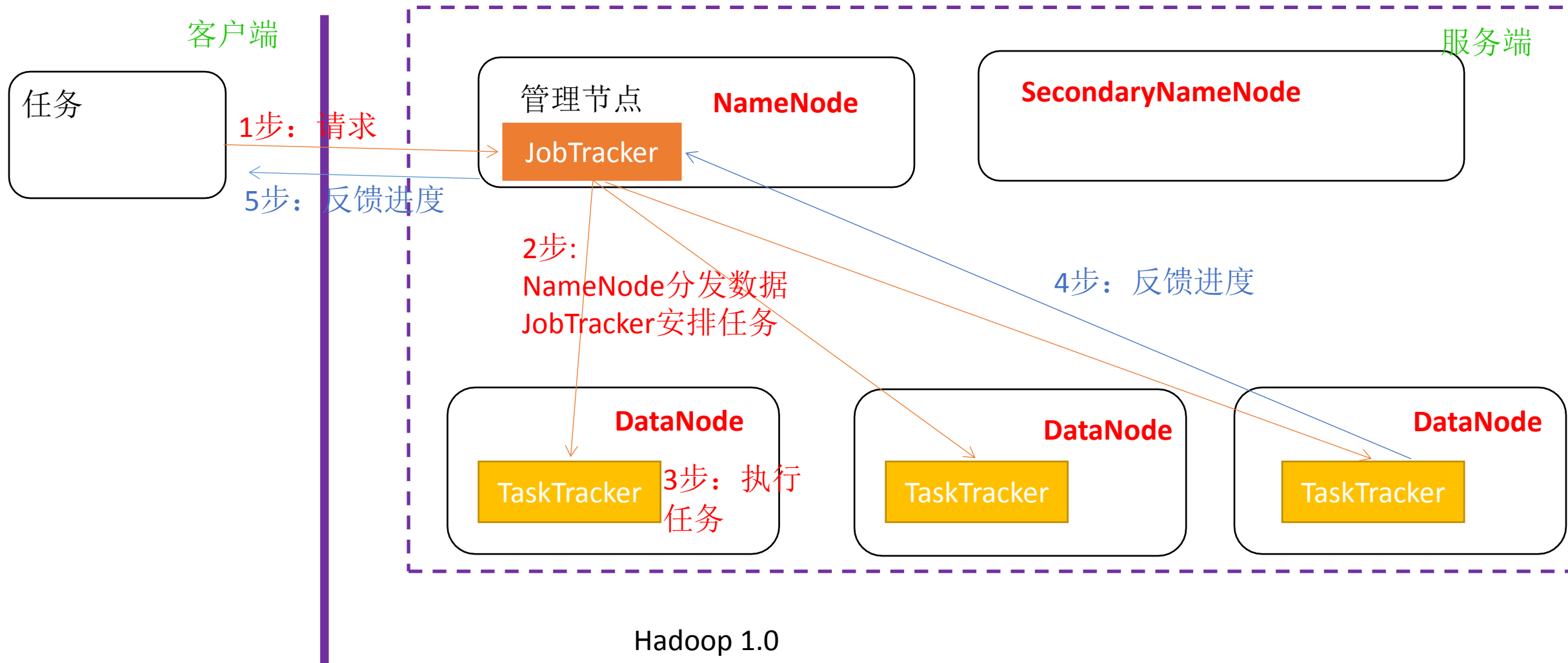
Client:
客户端

JobTracker:
负责资源管理和所有作业的控制

TaskTracker:
负责接收来自JobTracker的命令并执行



4.1.3 MapReduce 1 工作流程(了解即可)



4.1.4 MapReduce 1 局限性

MapReduce 1的JobTracker有多忙？

1、作业调度（把任务安排给TaskTracker）

2、任务进度监控（跟踪任务、重启失败的任务；记录任务流水）

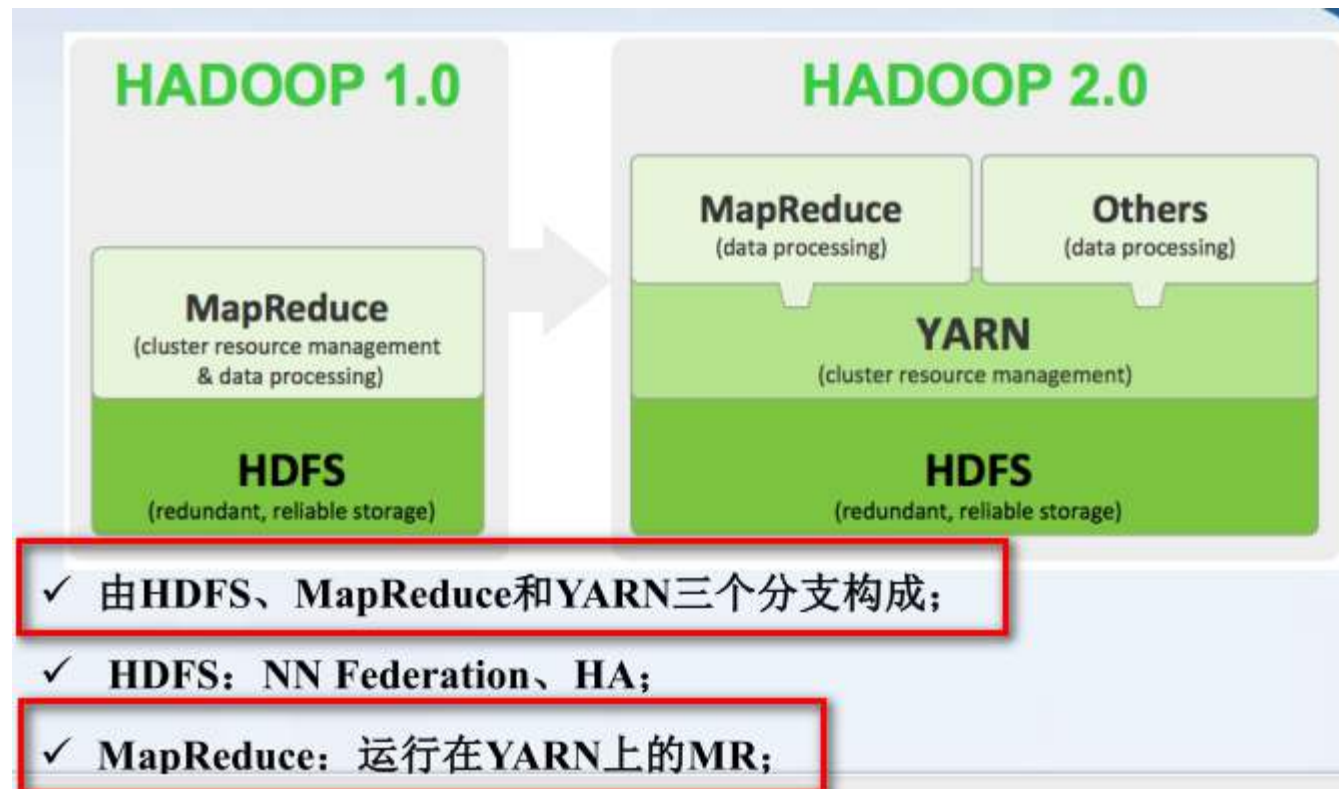
4.1.4 MapReduce 1 局限性

1. JobTracker访问压力大，影响系统扩展性，不适合所有大型计算。主要表现在大型集群上。官方称当节点数达到4000，任务数达到40000，MapReduce会遇到可扩展瓶颈。

其他：

2. 难以支持除MapReduce之外的框架，如Spark、Storm等。
3. JobTracker单点故障（因为Hadoop 1.0的本身就有单点故障）。

4.1.5 MapReduce 2



可以这样理解: MapReduce 2 \rightarrow MapReduce + YARN



初识YARN

4.2.1 YARN - 进程


```
hadoop@node1:~/hadoop/etc/hadoop$ jps
11477 SecondaryNameNode
32265 Jps
11258 NameNode
11631 ResourceManager
```

```
hadoop@node2:~$ jps
11520 DataNode
11654 NodeManager
16762 Jps
```

4.2.2 YARN - Web界面

← → ↻ ⓘ 不安全 | 192.168.1.51:8088/cluster 🔍 ☆ 👤 ⋮

Logged in as: dr.who



All Applications

▼ Cluster

- [About](#)
- [Nodes](#)
- [Node Labels](#)
- [Applications](#)
- [NEW](#)
- [NEW SAVING](#)
- [SUBMITTED](#)
- [ACCEPTED](#)
- [RUNNING](#)
- [FINISHED](#)
- [FAILED](#)
- [KILLED](#)
- [Scheduler](#)

► Tools

Cluster Metrics

| Apps Submitted | Apps Pending | Apps Running | Apps Completed | Containers Running | Memory Used | Memory Total | Memory Reserved | VCores Used | VCores Total | VCores Reserved | Active Nodes | Decommissioned Nodes | Lost Nodes | Unhealthy Nodes | Rebooted Nodes |
|----------------|--------------|--------------|----------------|--------------------|-------------|--------------|-----------------|-------------|--------------|-----------------|--------------|----------------------|------------|-----------------|----------------|
| 0 | 0 | 0 | 0 | 0 | 0 B | 16 GB | 0 B | 0 | 16 | 0 | 2 | 0 | 0 | 0 | 0 |

Scheduler Metrics

| Scheduler Type | Scheduling Resource Type | Minimum Allocation | Maximum Allocation |
|--------------------|--------------------------|-------------------------|--------------------------|
| Capacity Scheduler | [MEMORY] | <memory:1024, vCores:1> | <memory:8192, vCores:32> |

Show 20 ▼ entries Search:

| ID | User | Name | Application Type | Queue | StartTime | FinishTime | State | FinalStatus | Progress | Tracking UI | Blacklisted Nodes |
|----------------------------|------|------|------------------|-------|-----------|------------|-------|-------------|----------|-------------|-------------------|
| No data available in table | | | | | | | | | | | |

Showing 0 to 0 of 0 entries First Previous Next Last

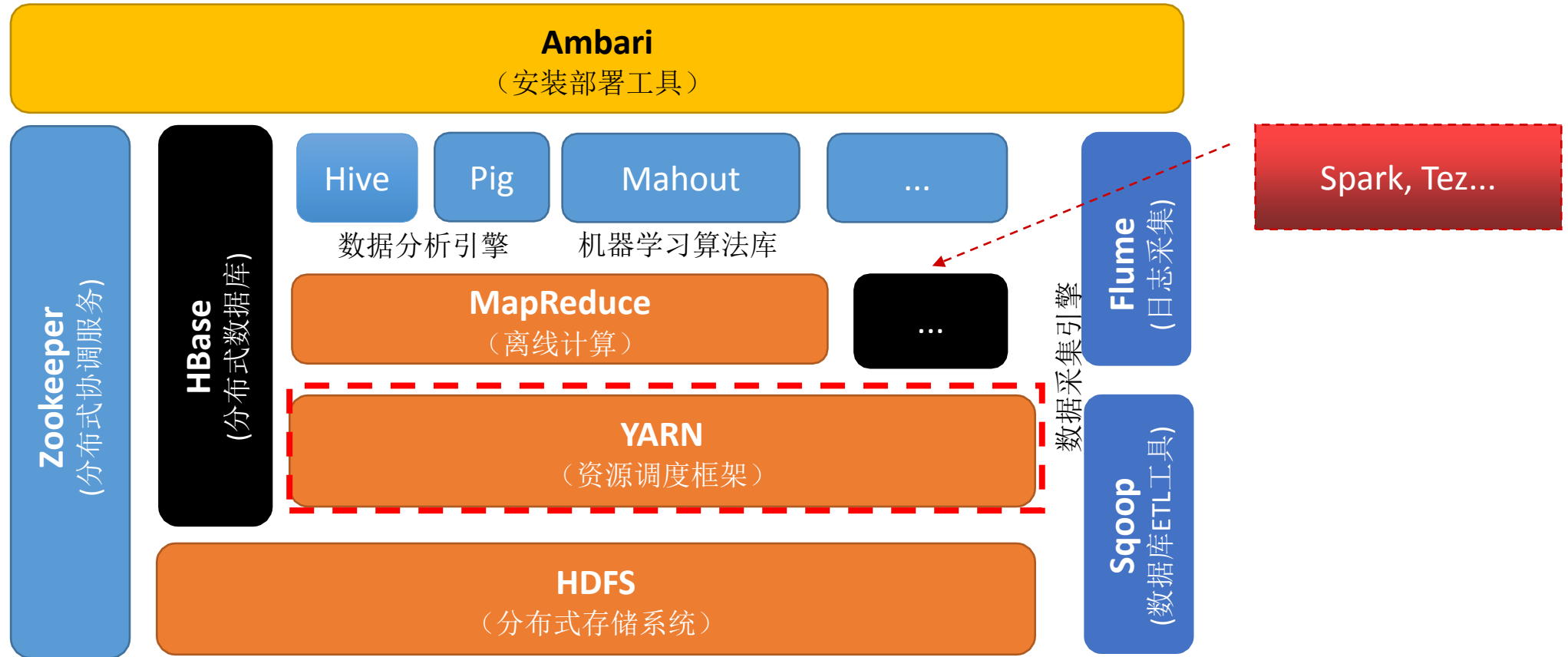
4.2.3 什么是YARN

- ✓ YARN : Yet Another Resource Negotiator, 另一种资源协调者。
- ✓ 最初是为了改善MapReduce的实现, 但也是一种资源调度框架, 具有**通用性**, 可为上层应用提供统一的资源管理和调度, 可以支持其他的分布式计算模式 (如Spark) 。
- ✓ 它的引入为集群在利用率、资源统一管理和数据共享等方面带来了巨大好处。

4.2.3 什么是YARN

YARN是一种通用资源调度框架，
不仅仅支持MapReduce

4.2.4 YARN在Hadoop生态圈中位置





YARN架构

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/768021065143006053>