淘宝网
Taobao.com

淘宝分布式并行计算框架 fourinone

Stone.Peng

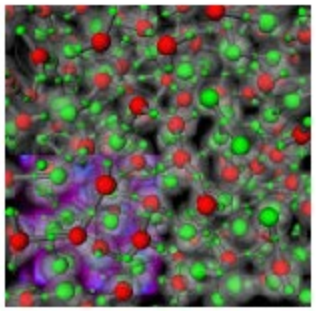
提纲

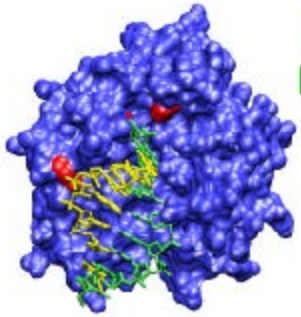
- 背景:我们需要解决的问题
- 分布式计算*并行计算*云计算
- Hadoop*Zookeeper*Hbase概述
- Fourinone介绍
- Fourinone应用场景:上亿数据排序
- Fourinone 2.0 新功能介绍

我们需要解决的问题

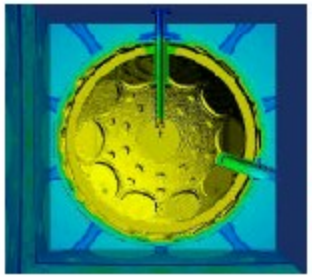
科学发现：万亿次计算实例

Electronic structure calculations



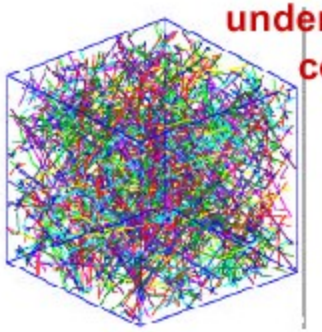


Modeling of DNA-protein binding

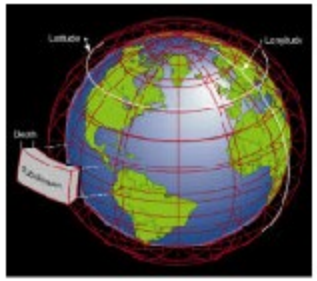


Largest-ever neutron transport simulations

Simulation of materials properties under extreme conditions



Highest resolution climate and carbon cycle modeling



我们需要解决的问题

需要非常巨大的计算能力才能解决的问题。这类问题很多还是跨学科的、极富挑战性的、人类急待解决的科研课题:

1.解决较为复杂的数学问题，例如：GIMPS（寻找最大的梅森素数）。 2.研究寻找最为安全的密码系统，例如：RC-72（密码破解）。

3.生物病理研究，例如：Folding@home（研究蛋白质折叠，误解，聚合及由此引起的相关疾病）。

4.各种各样疾病的药物研究，例如：United Devices（寻找对抗癌症的有效的药物）。

5.信号处理，例如：SETI@Home（在家寻找地外文明）。

从这些实际的例子可以看出，这些项目都很庞大，需要惊人的计算量，仅仅由单个的电脑或是个人在一个能让人接受的时间内计算完成是决不可能的。在以前，这些问题都应该由超级计算机来解决。但是, 超级计算机的造价和维护非常的昂贵，这不是一个普通的科研组织所能承受的。随着科学的发展，一种廉价的、高效的、维护方便的计算方法应运而生——分布式计算！

提纲

- 背景:我们需要解决的问题
- 分布式计算*并行计算*云计算
- Hadoop*Zookeeper*Hbase概述
- Fourinone介绍
- Fourinone应用场景:上亿数据排序
- Fourinone 2.0 新功能介绍

分布式计算

- **所谓分布式计算是一门计算机科学，它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算机进行处理，最后把这些计算结果综合起来得到最终的结果。**
  最近的分布式计算项目已经被用于使用世界各地成千上万位志愿者的计算机的闲置计算能力，通过因特网，您可以分析来自外太空的电讯号，寻找隐蔽的黑洞，并探索可能存在的外星智慧生命；您可以寻找超过1000万位数字的梅森质数；您也可以寻找并发现对抗艾滋病病毒的更为有效的药物。这些项目都很庞大，需要惊人的计算量，仅仅由单个的电脑或是个人在一个能让人接受的时间内计算完成是决不可能的。

思考:

我们能否将访问淘宝网的几千万个用户电脑利用做一次分布式计算?

什么是并行计算

并行计算（Parallel Computing）是指同时使用多种计算资源解决计算问题的过程。并行计算的主要目的是快速解决大型且复杂的计算问题。此外还包括：利用非本地资源，节约成本 — 使用多个“廉价”计算资源取代大型计算机，同时克服单个计算机上存在的存储器限制。

传统地，串行计算是指在单个计算机（具有单个中央处理单元）上执行软件写操作。CPU 逐个使用一系列指令解决问题，但其中只有一种指令可提供随时并及时的使用。并行计算是在串行计算的基础上演变而来，它努力仿真自然世界中的事务状态：一个序列中众多同时发生的、复杂且相关的事件。

并行计算的特点

为利用并行计算，通常计算问题表现为以下特征：

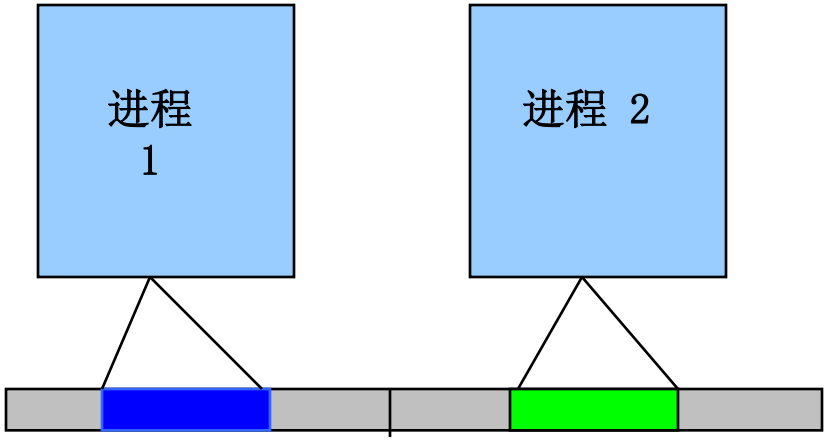
（1）将工作分离成离散部分，有助于同时解决；

（2）随时并及时地执行多个程序指令；

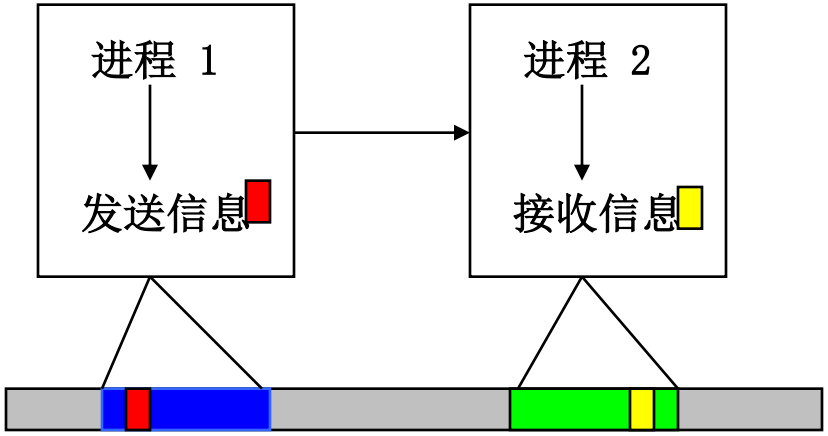
（3）多计算资源下解决问题的耗时要少于单个计算资源下的耗时。

并行计算是相对于串行计算来说的，所谓并行计算分为时间上的并行和空间上的并行。 时间上的并行就是指流水线技术，而空间上的并行则是指用多个处理器并发的执行计算。

并行计算与串行计算



传统的**串行计算**，分为“指令”和“数据”两个部分，并在程序执行时“独立地申请和占有”内存空间，且所有计算均局限于该内存空间。

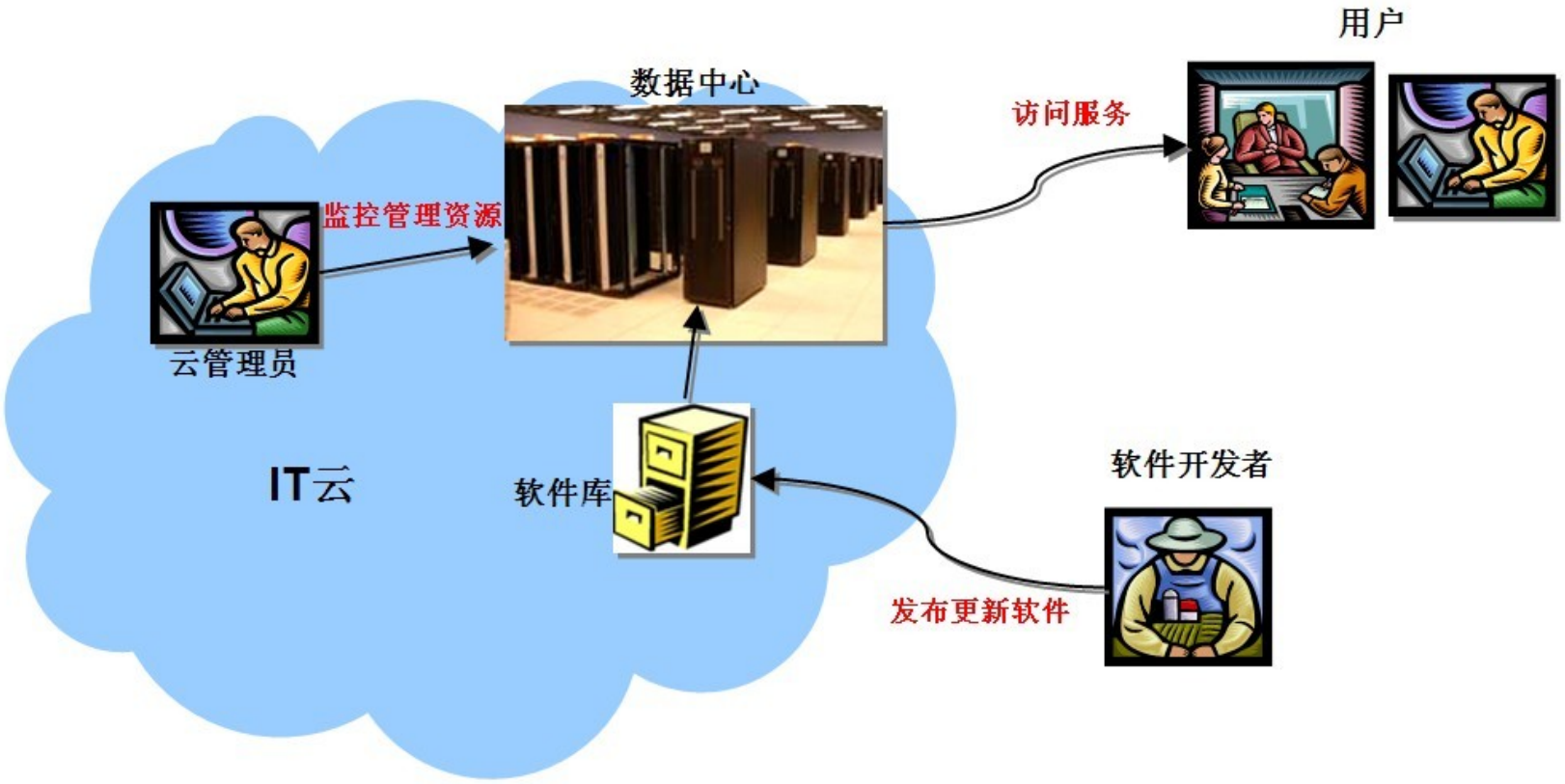


并行计算将进程相对独立的分配于不同的节点上，由各自独立的操作系统调度，享有独立的CPU和内存资源（内存可以共享）；进程间相互信息交换通过消息传递；

什么是云计算

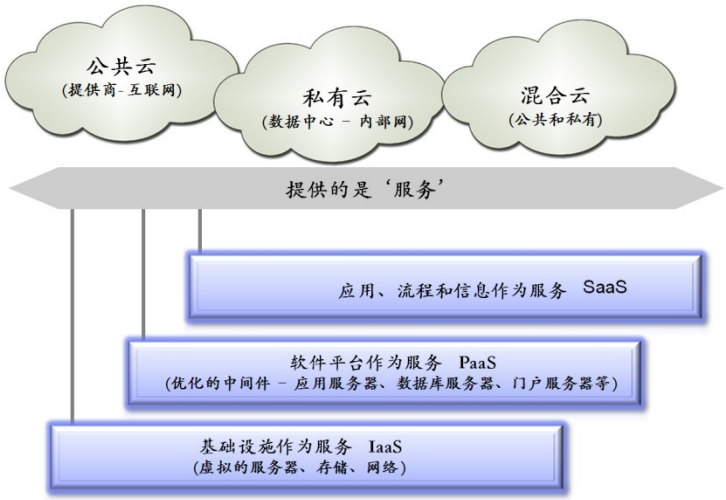
- 一种计算模式：把IT资源、数据、应用作为服务通过网络提供给用户（IBM）
- 一种基础架构管理方法论：把大量的高度虚拟化的资源管理起来，组成一个大的资源池，用来统一提供服务（IBM）
- 以公开的标准和服务为基础，以互联网为中心，提供安全、快速、便捷的数据存储和网络计算服务（Google）

云计算实例图



云计算的资源与服务

- 虚拟化技术：资源虚拟化、统一分配监测资源、向资源池中添加资源
- 服务思想
  - 软件即服务
    (Software-as-a-Service)
  - 平台即服务
    (Platform-as-a-Service)
  - 基础设施作为服务
    (Infrastructure as a Service



思考:分布式并行计算跟云计算的关系?

提纲

- 背景:我们需要解决的问题
- 分布式计算*并行计算*云计算
- Hadoop*Zookeeper*Hbase概述
- Fourinone介绍
- Fourinone上手Demo

Google的两篇论文

- 2002-2004: Apache Nutch
- 2004-2006:
  - Google 发表 GFS 和 MapReduce相关论文
  - Apache 在Nutch中实现HDFS和MapReduce
- 2006-2008:
  - Hadoop 项目从Nutch中分离
  - 2008年7月，Hadoop赢得Terabyte Sort Benchmark

Hadoop作者

Doug Cutting

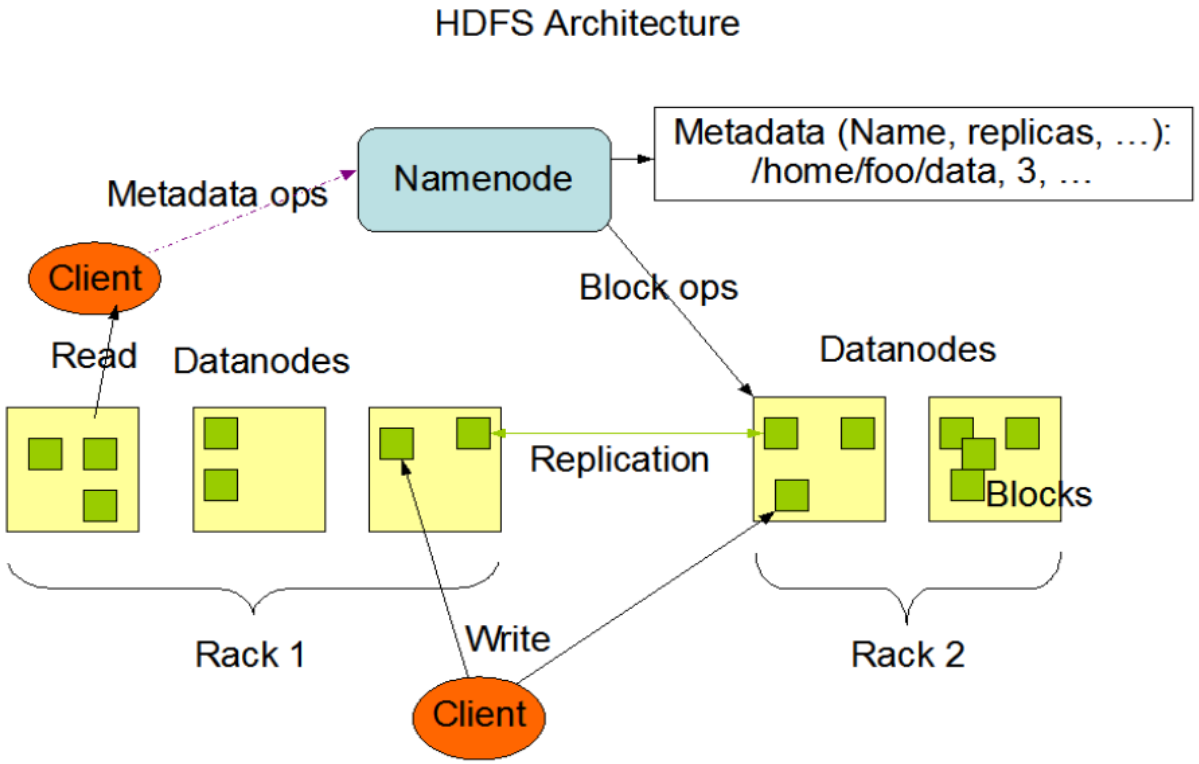
Hadoop和

Nutch和Lucene之父

早年供职yahoo

HDFS

- Fault-tolerant, 容错性
- Run on commodity hardware，在通用的机器上运行
- Scalable 可扩缩的



1个 namenode

多个 datanodes

http://hadoop.apache.org/hdfs/docs/current/hdfs_design.html

http://labs.google.com/papers/gfs.html

HDFS

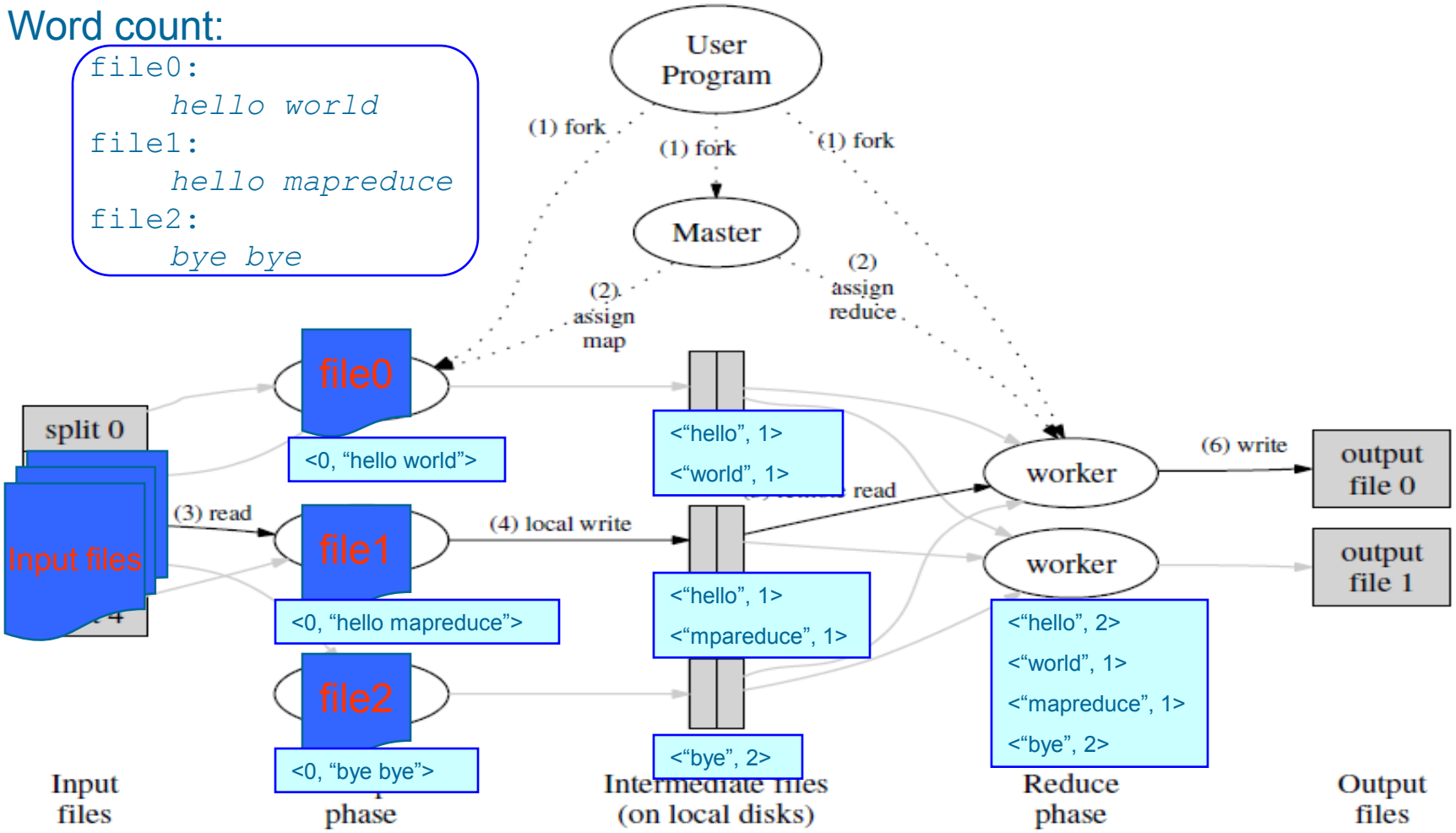
- NameNode
  - 存贮HDFS的元数据(metadata)
  - 管理文件系统的命名空间（namespace）
    - 创建、删除、移动、重命名文件和文件夹
  - 接收从DataNode来的Heartbeat 和 Blockreport
- DataNode
  - 存贮数据块
  - 执行从Namenode来的文件操作命令
  - 定时向NameNode发送Heartbeat和Blockreport

MapReduce

- Jobtraker (Master)
  - 接收任务（job）的提交
  - 提供任务的监控(monitoring)和控制(control)
  - 把job划分成多个tasks，交给Tasktracker执行，并管理这些tasks的执行
- Tasktracker (Worker)
  - 管理单个task的map任务和reduce任务的执行

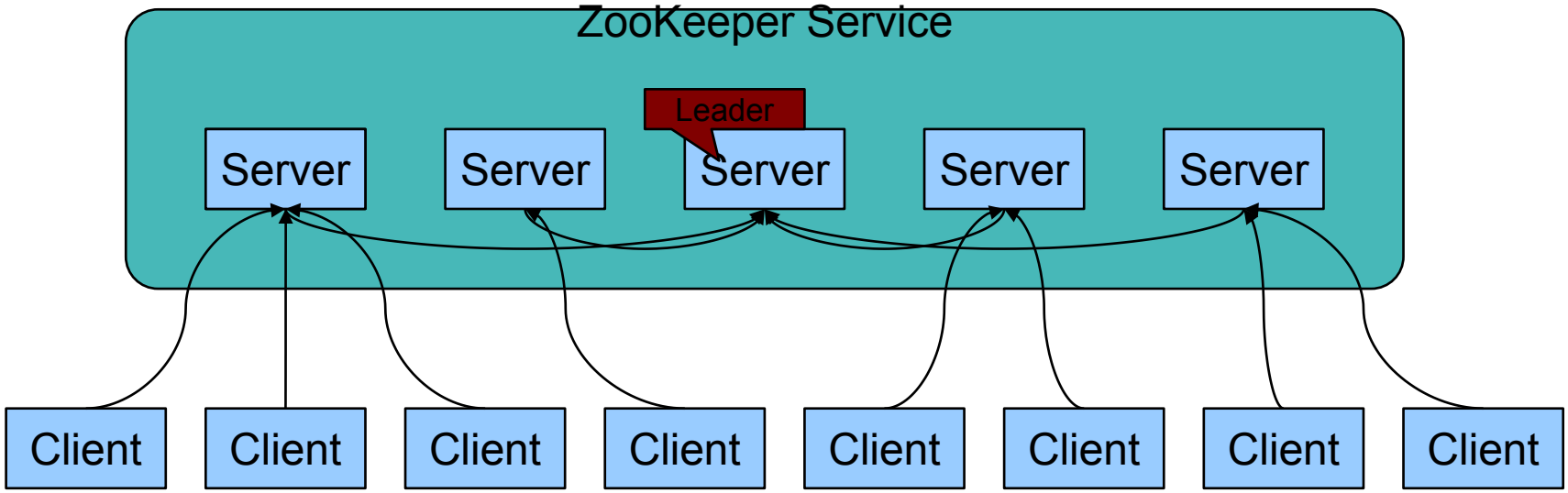
Word count:

```
file0:
    hello world
file1:
    hello mapreduce
file2:
    bye bye
```



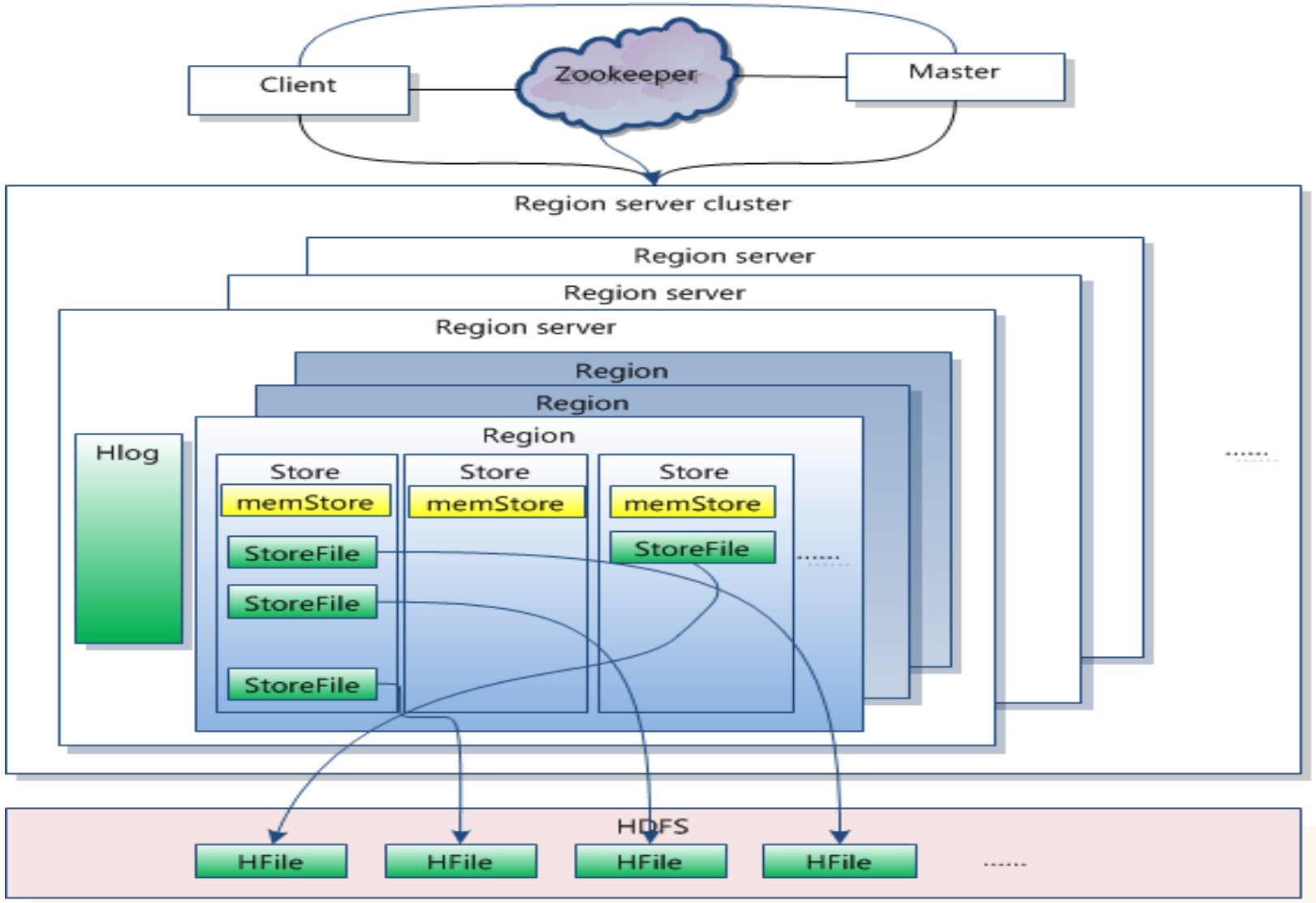
files → <line offset, line content> → <word, count> → <word, count> → files

ZooKeeper



- 启动Zookeeper服务器集群环境后，多个Zookeeper服务器在工作前会选举出一个Leader，在接下来的工作中这个被选举出来的Leader死了，而剩下的Zookeeper服务器会知道这个Leader死掉了，在活着的Zookeeper集群中会继续选出一个Leader，选举出leader的目的是为了可以在分布式的环境中保证数据的一致性。
- 另外，ZooKeeper 支持watch(观察)的概念。客户端可以在每个znode结点上设置一个观察。如果被观察服务端的znode结点有变更，那么watch就会被触发，这个watch所属的客户端将接收到一个通知包被告知结点已经发生变化。若客户端和所连接的ZooKeeper服务器断开连接时，其他客户端也会收到一个通知，也就说一个Zookeeper服务器端可以对于多个客户端，当然也可以多个Zookeeper服务器端可以对于多个客户端

Hbase


Client
Zookeeper
Master
Region server cluster
Region server
Region server
Region server
Region
Region
Region
Hlog
Store
memStore
StoreFile
StoreFile
StoreFile
Store
memStore
Store
memStore
StoreFile
HDFS
HFile
HFile
HFile
HFile

Hbase

- 上面是hbase的架构，里面的hdfs也就是hadoop中的分布式文件系统。里面主要的核心组件简单介绍如下：
- Client
- 包含访问hbase的接口，client维护着一些cache来加快对hbase的访问，比如regione的位置信息。
-
- Zookeeper
- 1 保证任何时候，集群中只有一个master
- 2 存贮所有Region的寻址入口。
- 3 实时监控Region Server的状态，将Region server的上线和下线信息实时通知给Master
- 4 存储Hbase的schema,包括有哪些table，每个table有哪些column family
-
- Master
- 1 为Region server分配region
- 2 负责region server的负载均衡
- 3 发现失效的region server并重新分配其上的region
- 4 GFS上的垃圾文件回收
- 5 处理schema更新请求
-
- Region Server
- 1 Region server维护Master分配给它的region，处理对这些region的IO请求
- 2 Region server负责切分在运行过程中变得过大的region

提纲

- 背景:我们需要解决的问题
- 分布式计算*并行计算*云计算
- Hadoop*Zookeeper*Hbase概述
- Fourinone介绍
- Fourinone应用场景:上亿数据排序
- Fourinone 2.0 新功能介绍

使用Hadoop碰到的问题

尝试套用map/reduce的问题：

1、**http协议报文一个请求占多行，各行之前有一定逻辑关系，不能简单以行拆分和合并**

2、**复杂的中间过程的计算套用m/r不容易构思，如大数据的组合，多机计算并不只有拆分合并的需求（举例）**

3、**hadoop实现的太复杂，api枯燥难懂，不利于程序员迅速上手并有驾驭感**

4、**m/r容易将逻辑思维框住，业务逻辑不连贯，容易让程序员使用过程中总是花大量时间去搞懂框架本身的实现**

5、**在一台机器上未能很直接看出并行计算优势**

6、**没有一个简单易用的window版，需要模仿linux环境，安装配置复杂**

“以前，拥有博士学位背景的人才能使用Hadoop。但是例如医院和银行这样的机构，并没有这样的人员。 Hadoop的配置和管理的确很让人痛苦。但是现在我们提供了更容易的可以让“普通人”使用的Hadoop，”Hammerbacher说，他曾经在 Facebook创建并领导了一个Hadoop数据工作组。

从秒杀作弊分析延伸的想法

抽取一个简化的并行计算框架用于业务场景需要：

1、对并行计算map/reduce和forkjoin等思想的对比和研究。

2、定位在侧重小型灵活的对业务逻辑进行并行编程的框架，基于该框架可以简单迅速建立并行的处理方式，较大程度提高计算效率，可以在本地使用，也可以在多机使用，使用公共内存和脚本语言配置，能嵌入式使用。

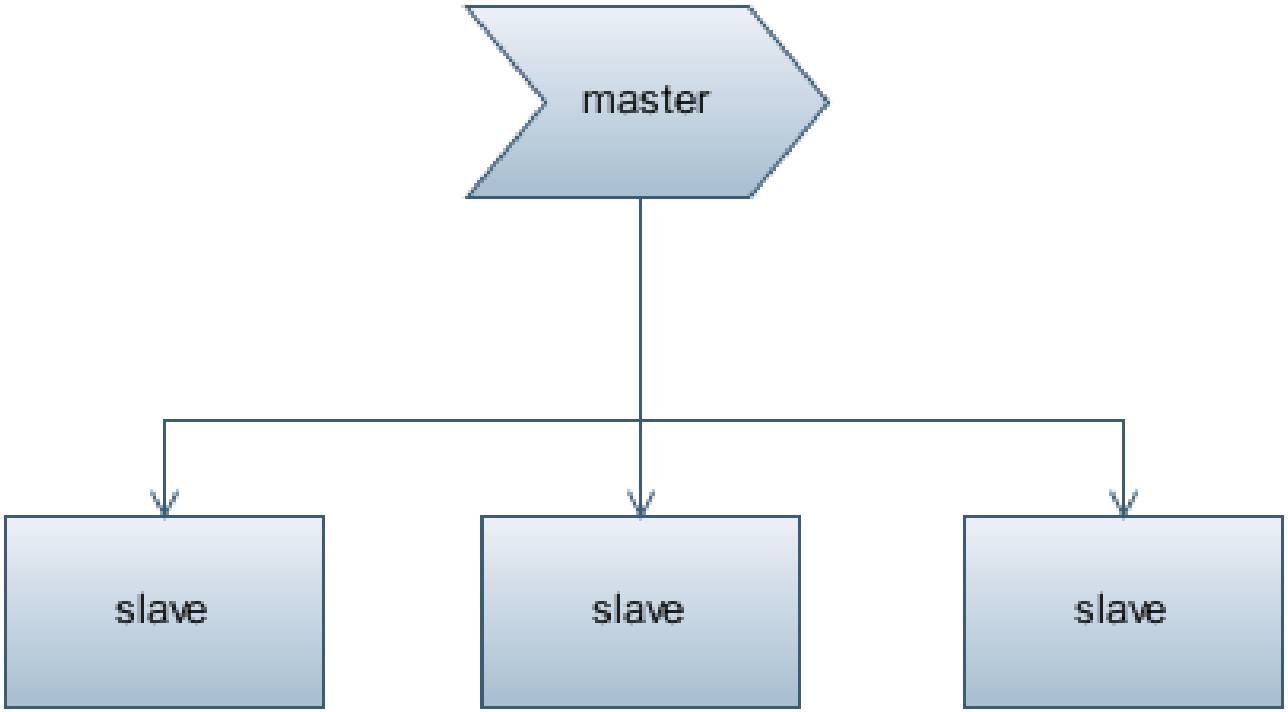
3、它可以并行高效进行大数据和逻辑复杂的运算，并对外提供调用服务，所有的业务逻辑由业务开发方提供，未来并行计算平台上可以高效支持秒杀器作弊分析、炒信软件分析、评价数据分析、帐务结算系统等业务逻辑运算

Fourinone介绍

当我们把复杂的hadoop当作一门学科学习时，似乎忘记了我们想解决问题的初衷：我们仅仅是想写个程序把几台甚至更多的机器一起用起来计算，把更多的cpu和内存利用上，来解决我们数量大和计算复杂的问题，当然这个过程中要考虑到分布式的协同和故障处理。如果仅仅是为了实现这个简单的初衷，为什么一切会那么复杂。

Fourinone（中文名字：四不像）是一个新的分布式并行计算框架，他集成了Hadoop,Zookeeper,MQ,分布式缓存四大主要的分布式计算功能， Fourinone的功能强大用途广泛，他实现了zookeeper的所有功能并进行了很多改进，它同时又提供完整的分布式缓存支持，包括中小型缓存以及大型集群缓存，他使用不同于map/reduce的全新设计模式解决问题，模仿现实中生产加工链式加并行处理的“包工头/农民工/手工仓库”方式设计分布式计算，他还可以当做简单的mq使用。Fourinone整体仅仅80k，就一个jar包没有任何依赖，很方便嵌入式开发使用

Fourinone分布式计算

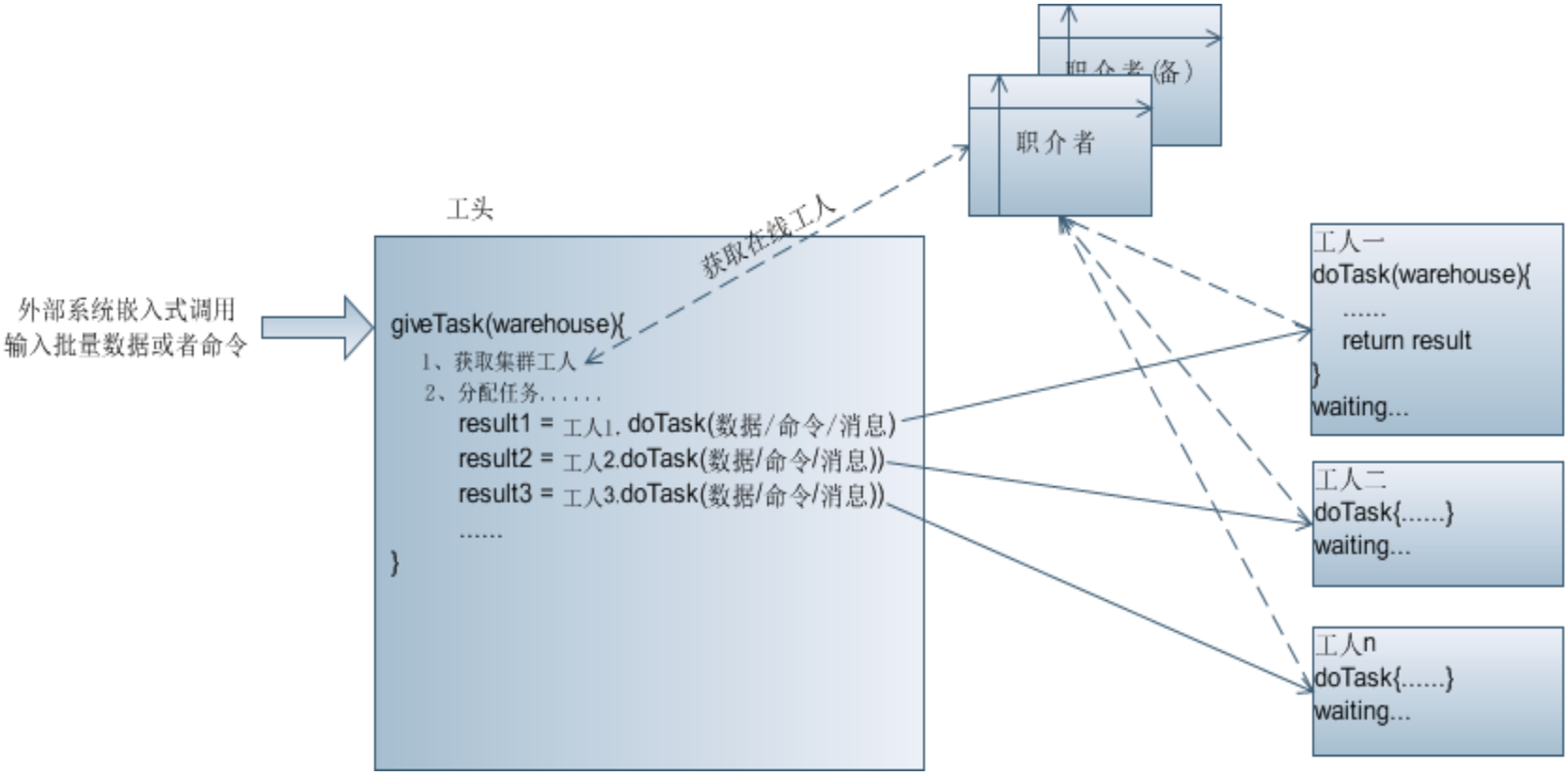


最简单的master-slave计算结构

master是一个服务程序, slave跟master耦合太紧

master除分配任务外需要负责协同一致性等处理

Fourinone分布式计算



fourinone的简化分布式并行计算结构

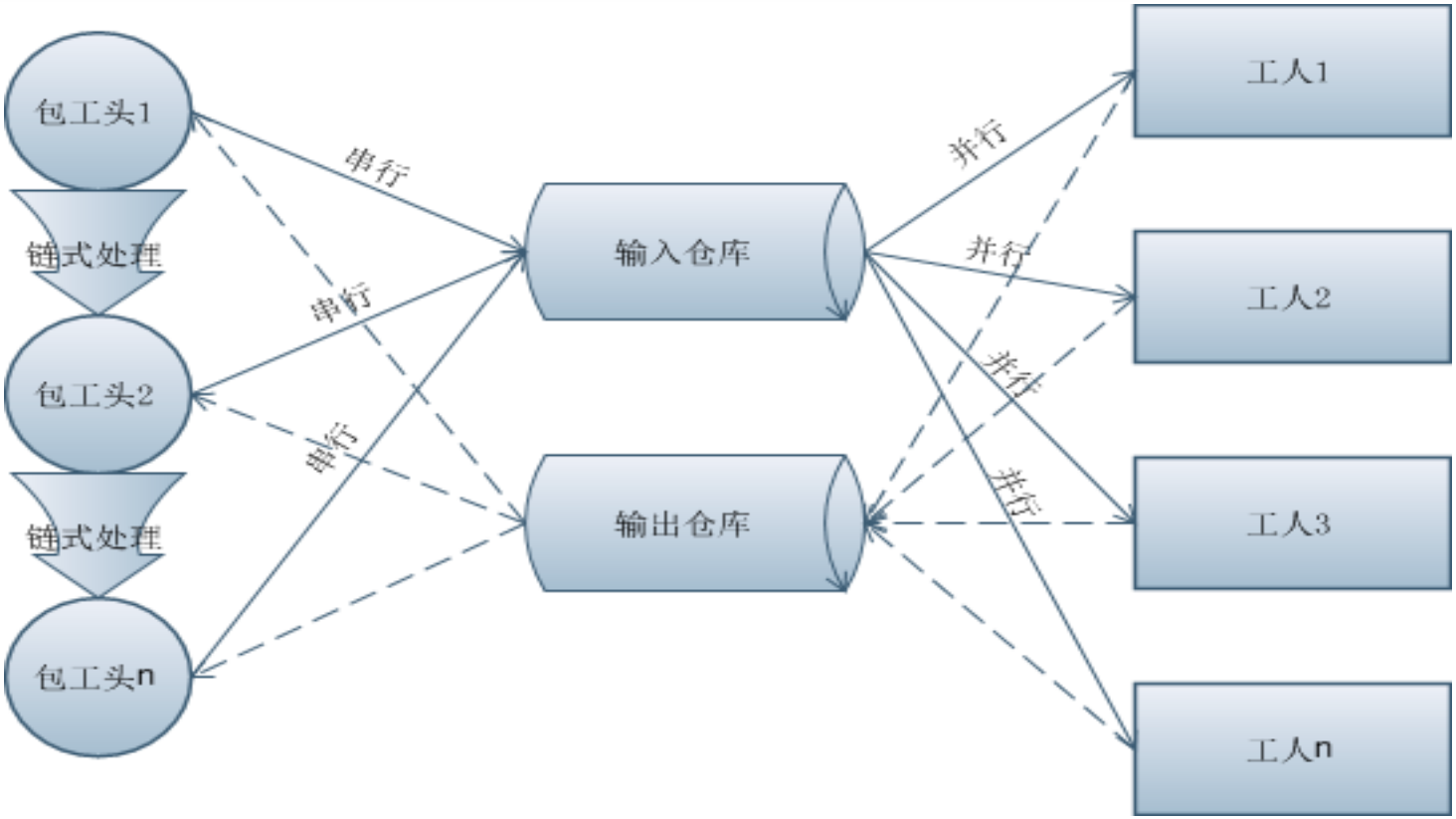
包工头去服务化，嵌入式，负责分配任务，开发者实现分配任务接口

农民工负责执行任务，开发者实现任务执行接口

职介者负责协同一致性等处理（登记，介绍，保持联系）

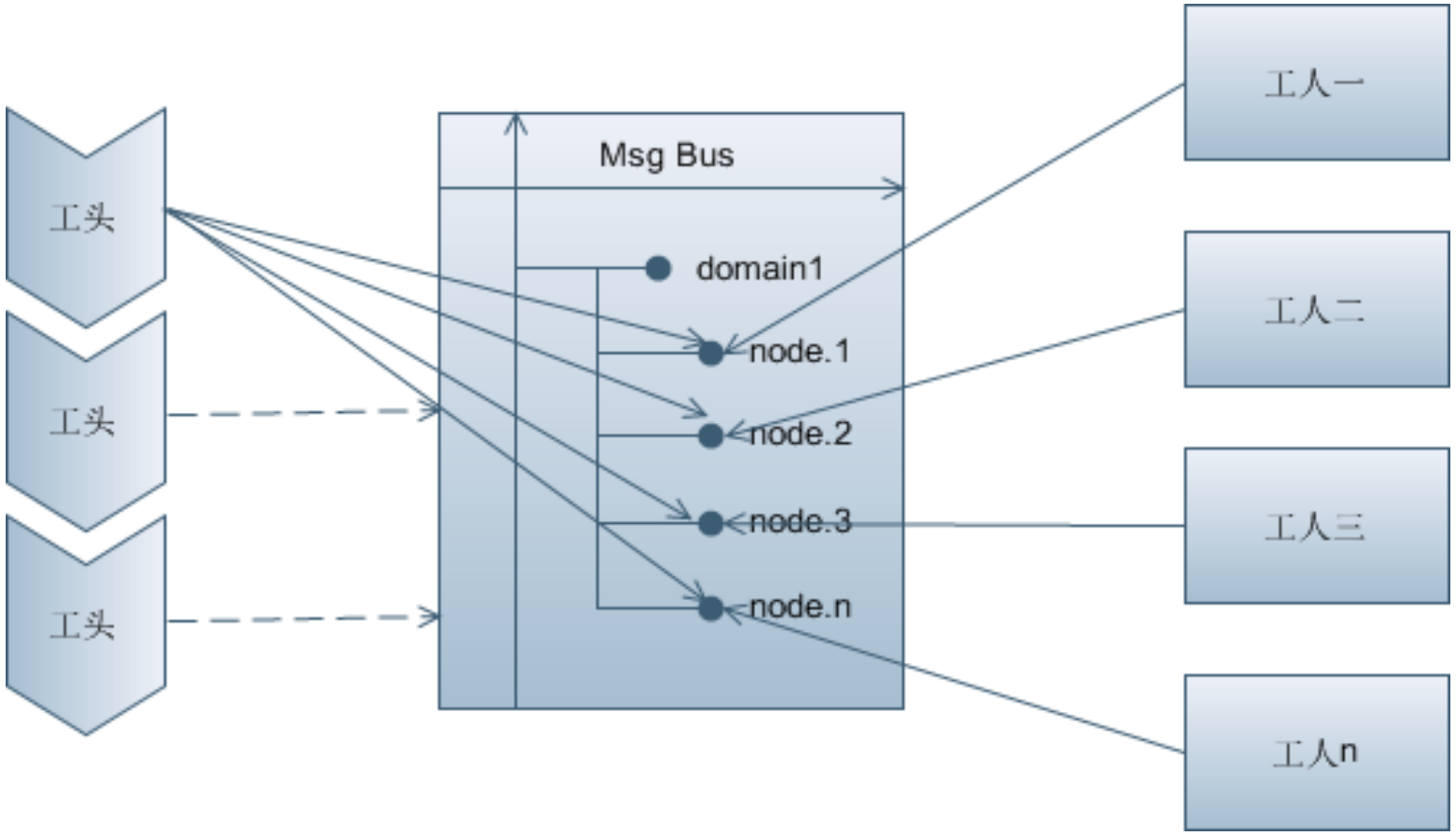
思考：是否能满足storm这样的实时流计算模型？

Fourinone分布式计算



总的来说，是将大数据的复杂分布式计算，设计为一个链式的多“包工头”环节去处理，每个环节包括利用多台“农民工”机器进行并行计算，无论是拆分计算任务还是合并结果，都可以设计为一个单独的“包工头”环节。这样做的好处是，开发者有更大能力去深入控制并行计算的过程，去保持使用并行计算实现业务逻辑的完整性，而且对各种不同类型的并行计算场景也能灵活处理，不会因为某些特殊场景被map/reduce的框架限制住思维，并且链式的每个环节也方便进行监控过程。

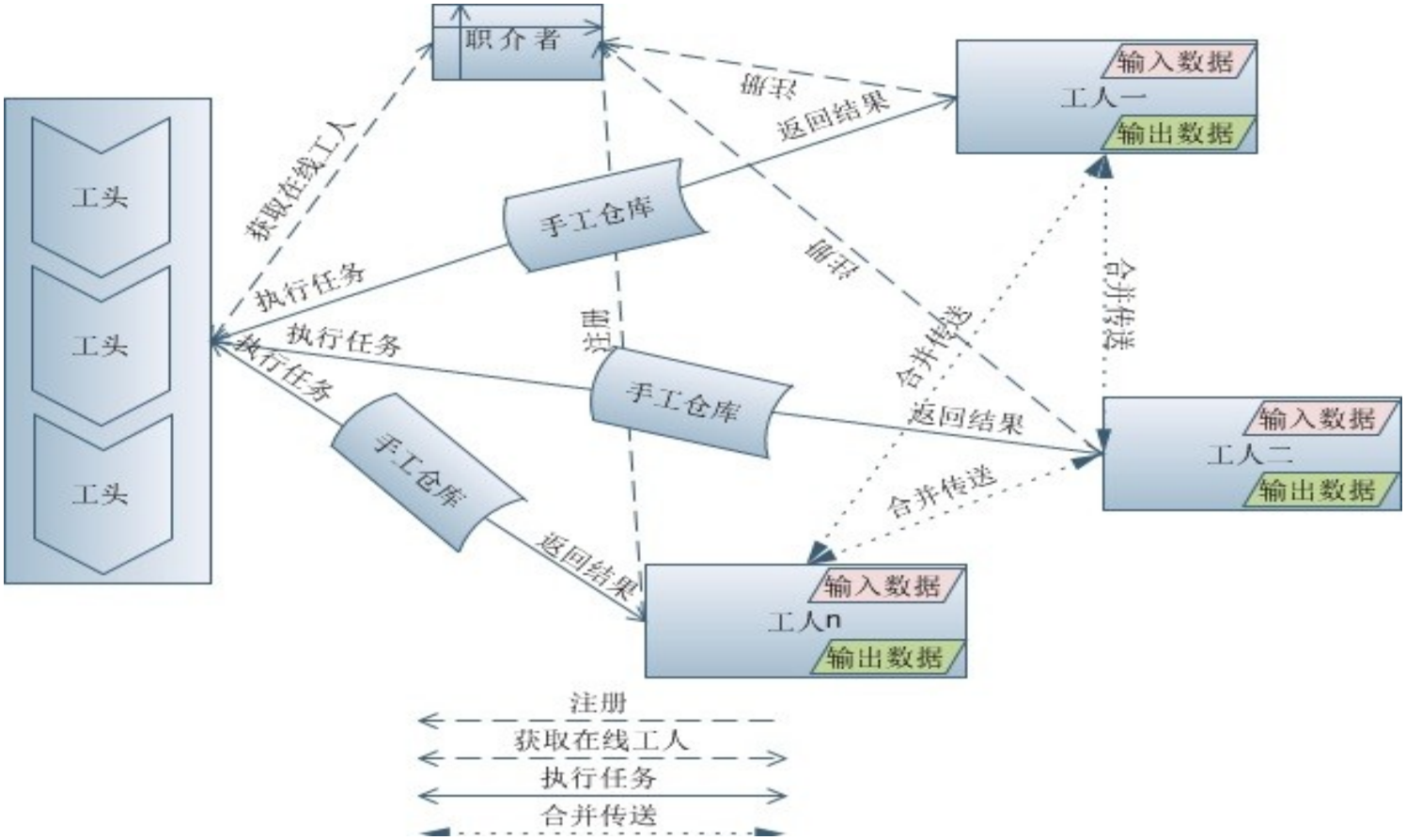
Fourinone分布式计算



- 模式一：基于消息中枢的计算模式

优势？缺点？

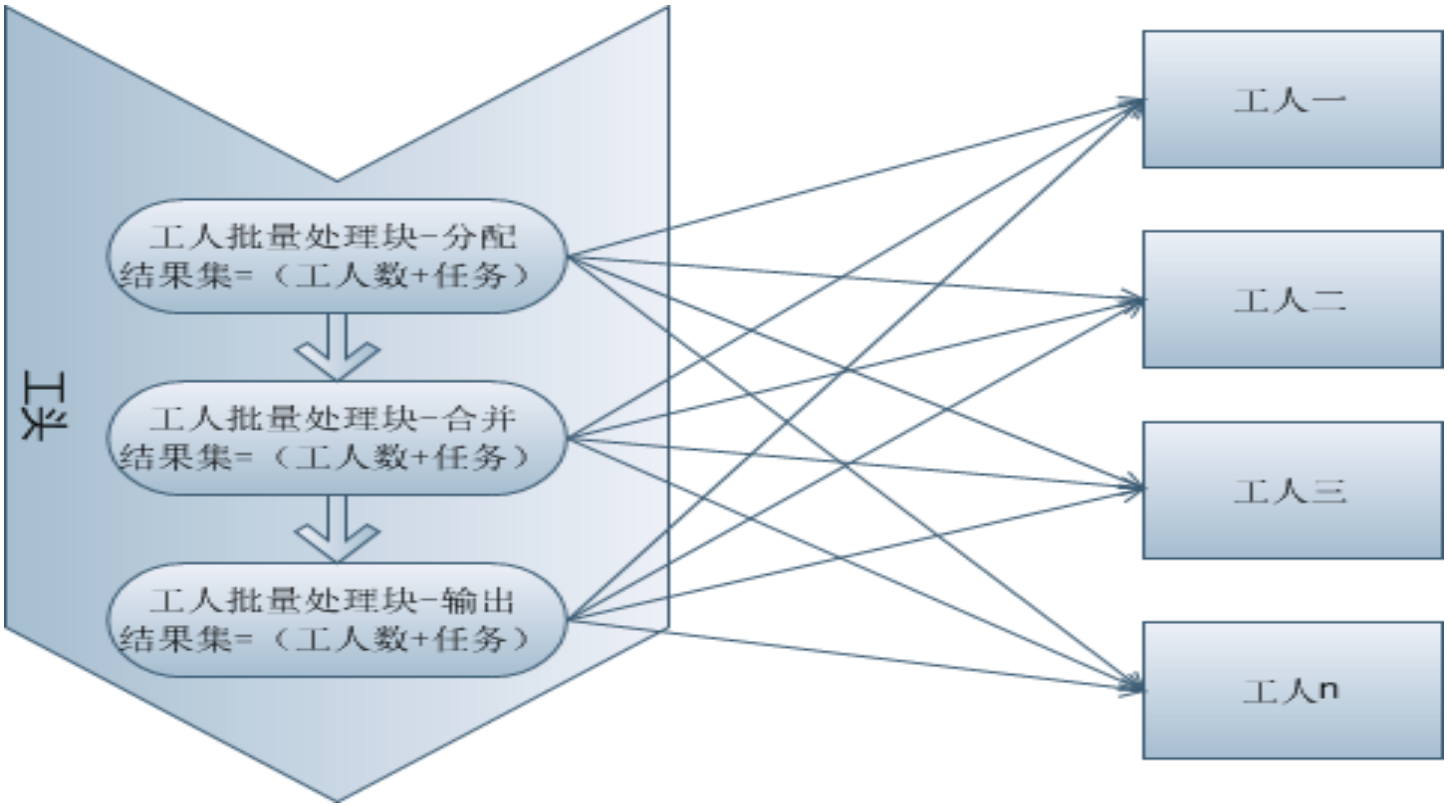
Fourinone分布式计算



- 模式二：基于工人服务的网状交互计算模式

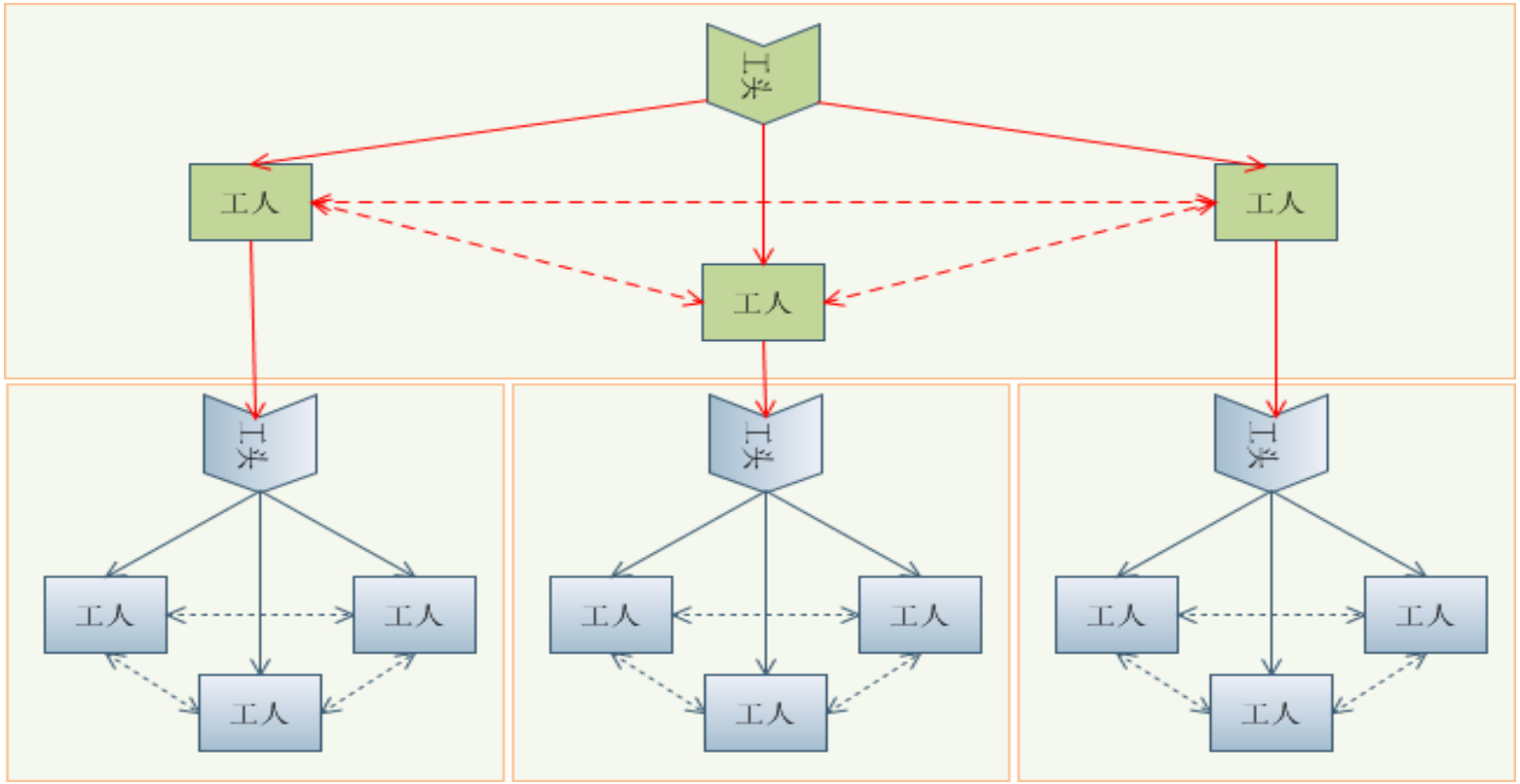
  优势？缺点？

Fourinone分布式计算



- 单个工头支持多阶段工人批量任务处理
- 思维发散：多工头的任务并行拆分

Fourinone分布式计算



- 多工头并行的计算集群搭建（兼顾遗留计算系统）
- 模仿现实中加工生产原材料承包分配

Fourinone和hadoop的对比

	fourinone-1.11.09	hadoop-0.21.0
体积	82K	71M
依赖关系	就一个jar,没有依赖	约12项jar包依赖
配置	就一个配置文件	较多配置文件和复杂属性
集群搭建	简单，每台机器放一个jar和配置文件	复杂，需要linux操作基础和ssh等复杂配置，还需要较多配置文件配置
计算模式	提供两种计算模式：包工头和工人直接交互方式，包工头和工人通过消息中枢方式交互，后者不需要工人节点可直接访问	计算更多倾向于文件数据的并行读取，而非计算过程的设计。JobTracke 跟 TaskTracker 直接交互， 查询 NameNode 后，TaskTracker直接从Datanode获取数据。
并行模式	N*N，支持单机并行，也支持多机并行，多机多实例并行	1*N，不支持单机并行，只支持多机单实例并行
内存方式	支持内存方式设计和开发应用，并内置完整的分布式缓存功能	以hdfs文件方式进行数据处理，内存方式计算支持很弱
文件方式	自带文件适配器处理io	Hdfs处理文件io
计算数据要求	任意数据格式和任意数据来源，包括来自数据库，分布式文件，分布式缓存等	Hdfs内的文件数据，多倾向于带换行符的数据
调度角色	包工头，可以有多个，支持链式处理，也支持大包工头对小包工头的调度	JobTracke，通常与NameNode一起
任务执行角色	农民工，框架支持设计多种类型的工人用于拆分或者合并任务	TaskTracker，通常与Datanode一起
中间结果数据保存	手工仓库，或者其他任意数据库存储设备	Hdfs中间结果文件
拆分策略	自由设计，框架提供链式处理对于大的业务场景进行环节拆分数据的存储和计算拆分根据业务场景自定义	以64m为拆分进行存储，以行为拆分进行计算 实现map接口，按行处理数据进行计算
合并策略	自由设计，框架提供农民工节点之间的合并接口，可以互相交互设计合并策略，也可以通过包工头进行合并	TaskTracker不透明，较少提供程序控制，合并策略设计复杂 实现reduce接口进行中间数据合并逻辑实现
内存耗用	无需要制定JVM内存，按默认即可，根据计算要求考虑是否增加JVM内存	需要制定JVM内存，每个进程默认1G，常常namenode，jobtracker等启动3个进程，耗用3G内存
监控	框架提供多环节链式处理设计支持监控过程，通过可编程的监控方式，给于业务开发方最大灵活的监控需求实现，为追求高性能不输出大量系统监控log	输出较多的系统监控log，如map和reduce百分比等，但是会牺牲性能，业务监控需要自己实现
打包部署	脚本工具	上传jar包到jobtracker机器
平台支撑	支持跨平台，windows支持良好	多倾向于支持linux，Windows支持不佳，需要模拟linux环境，并且建议只用于开发学习
其他	协同一致性、分布式缓存、通讯队列等跟分布式计算关系密切的功能支持	不支持
总结：	Hadoop并不是为了追求一个并行计算的框架而设计，提供快捷和灵活的计算方式去服务各种计算场景， 它更多的是一个分布式文件系统，提供文件数据的存储和查询，它的map/reduce更倾向于提供并行计算方式进行文件数据查询。而fourinone相反。	

Fourinone和hadoop的对比

Fourinone 和 hadoop 运行 wordcount 的对比测试（平均 4 核 4g 配置，输入数据为文件）：

	fourinone-1.11.09(n*4)	fourinone-1.11.09(n*1)	hadoop-0.21.0(n*1)
3 台机器*256M	4s	12s	72s
3 台机器*512M	7s	30s	140s
3 台机器*1G	14s	50s	279s
19 台机器*1G	21s	60s	289s
10 台机器*2G	29s		
5 台机器*4G	60s		

N*4 说明:Fourinone 可以充分利用单机并行能力,4 核计算机可以 4 个并行实例计算,hadoop 目前只能 N*1；另外，可以由上图看出，如果要完成 20g 的数据，实际上 fourinone 只需要使用 5 台机器用 60 秒完成，比使用 19 台机器完成 19g 的 hadoop 节省了 14 台机器，并提前了 200 多秒

Fourinone分布式协同

分布式协同方面，fourinone实现了Zookeeper所有的功能，并且做了很多改进：

1、简化Zookeeper的树型结构，用domain/node两层结构取代

2、简化Watch回调多线程等待编程模型，用更直观的容易保证业务逻辑完整性的内容变化事件以及状态轮循取代

3、Zookeeper只能存储信息不大于1M的内存内容，fourinone提供了内存管理控制，针对jvm的默认内存和调优内存等情况都能进行内存占用报警异常，避免内存溢出。

4、简化了Zookeeper的ACL权限功能，用更为程序员熟悉rw风格取代

5、简化了Zookeeper的临时节点和序列节点等类型，取代为在创建节点时是否指定保持心跳，心跳断掉时节点会自动删除。

6、FourInOne是高可用的，没有单点问题，可以有任意多个复本，它的复制不是定时而是基于内容变更复制，有更高的性能

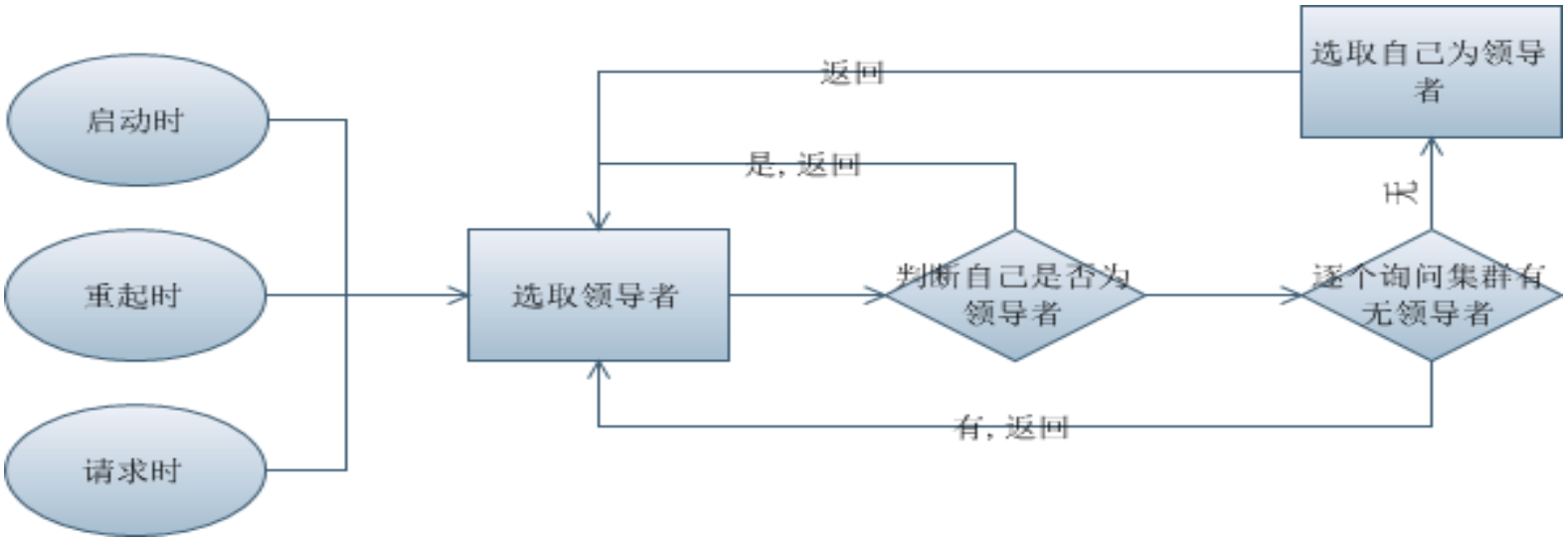
7、FourInOne实现了领导者选举算法（但不是Paxos），在领导者服务器宕机情况下，会自动不延时的将请求切换到备份服务器上，选举出新的领导者进行服务，这个过程中，心跳节点仍然能保持健壮的稳定性，迅速跟新的领导者保持心跳连接。

基于FourInOne可以轻松实现分布式配置信息，集群管理，故障节点检测，分布式锁，以及淘宝configserver等等协同功能。

Fourinone分布式协同

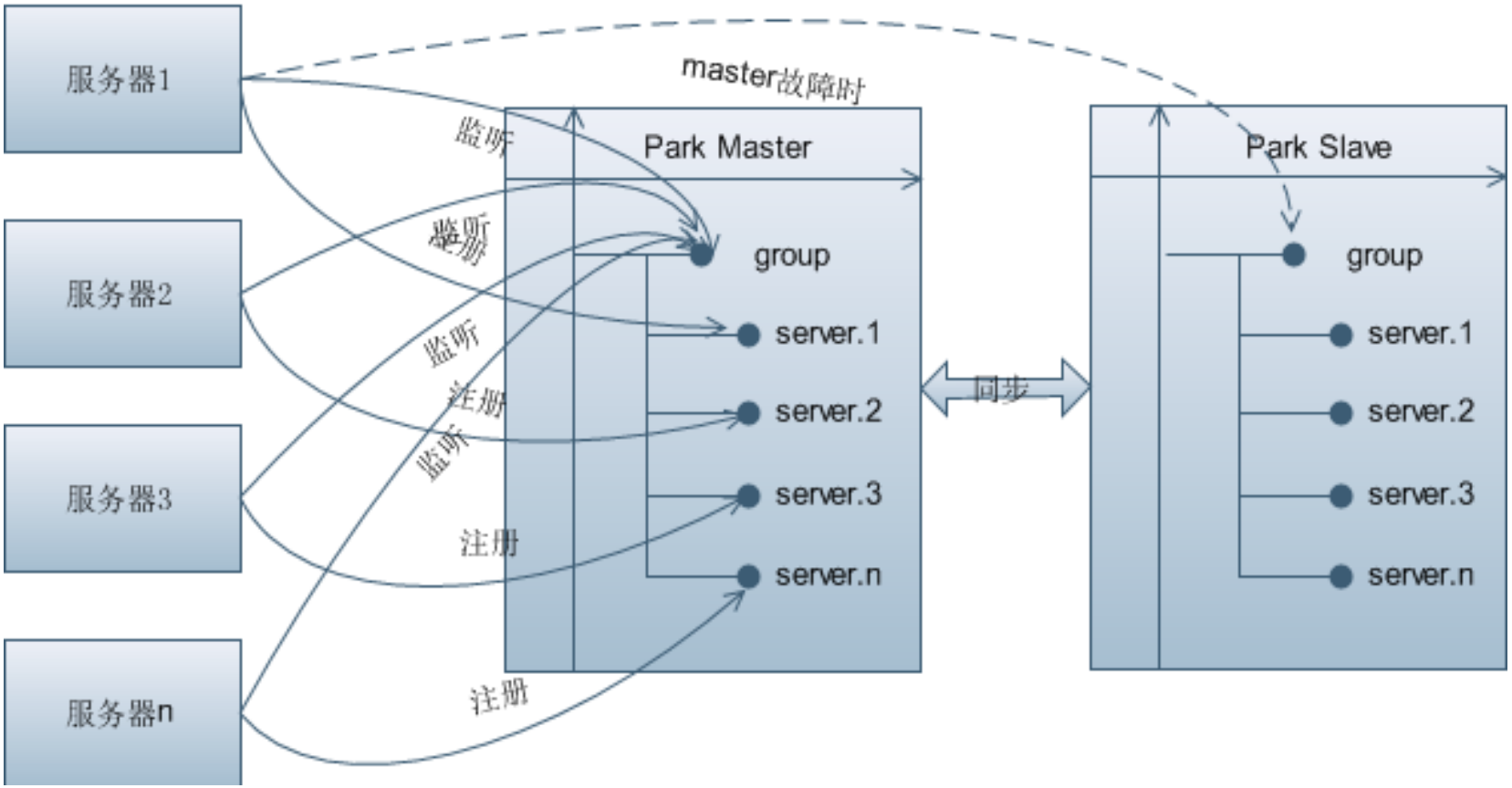
fourinone对比zookeeper的优势：zookeeper没有获取最新版本信息的方法支持，它只能粗暴的在每次写入更新等方法时注册一个watch，当这些方法被调用后就回调，它不考虑信息内容是否变化，对于没有使信息内容发生改变的更新，zookeeper仍然会回调，并且zookeeper的回调比较呆板，它只能用一次，如果信息持续变化，必须又重新注册watch, 而fourinone的事件处理则可以自由控制是否持续响应信息变化。

Fourinone分布式协同



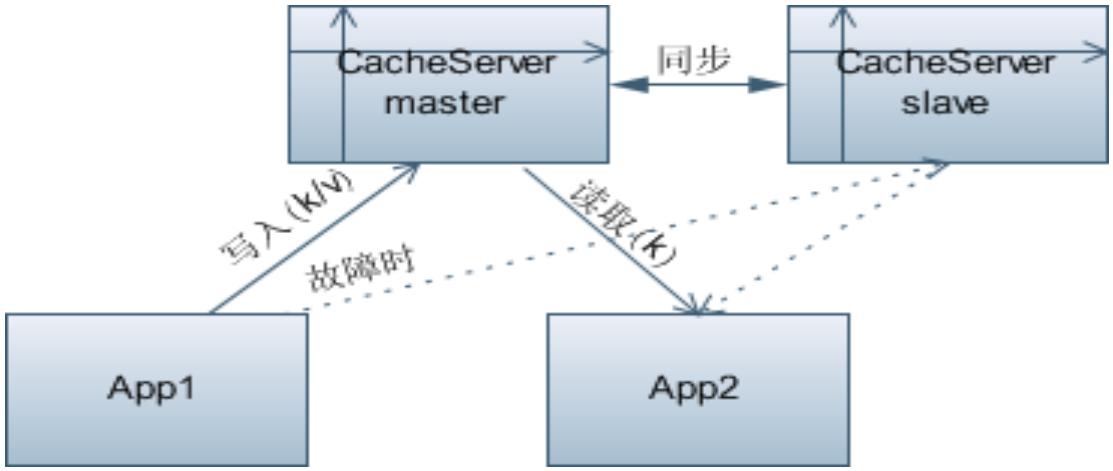
领导者选举：ZooKeeper的领导者选举实现虽然比原始的Paxos要简化，但是它仍然存在领导者（Leader)、跟随者（Follower）、观察者（observer）、学习者(Learner）等众多角色和跟随状态（Following)、寻找状态（Looking）、观察状态（Observing）、领导状态 (Leading）等复杂状态。fourinone的集群领导者算法，只存在领导者和候选者两种角色，同一时刻只有一个领导者处于领导状态，其余处于候选状态，对领导者选举算法进一步简化，能够更快捷的实现。

Fourinone分布式协同



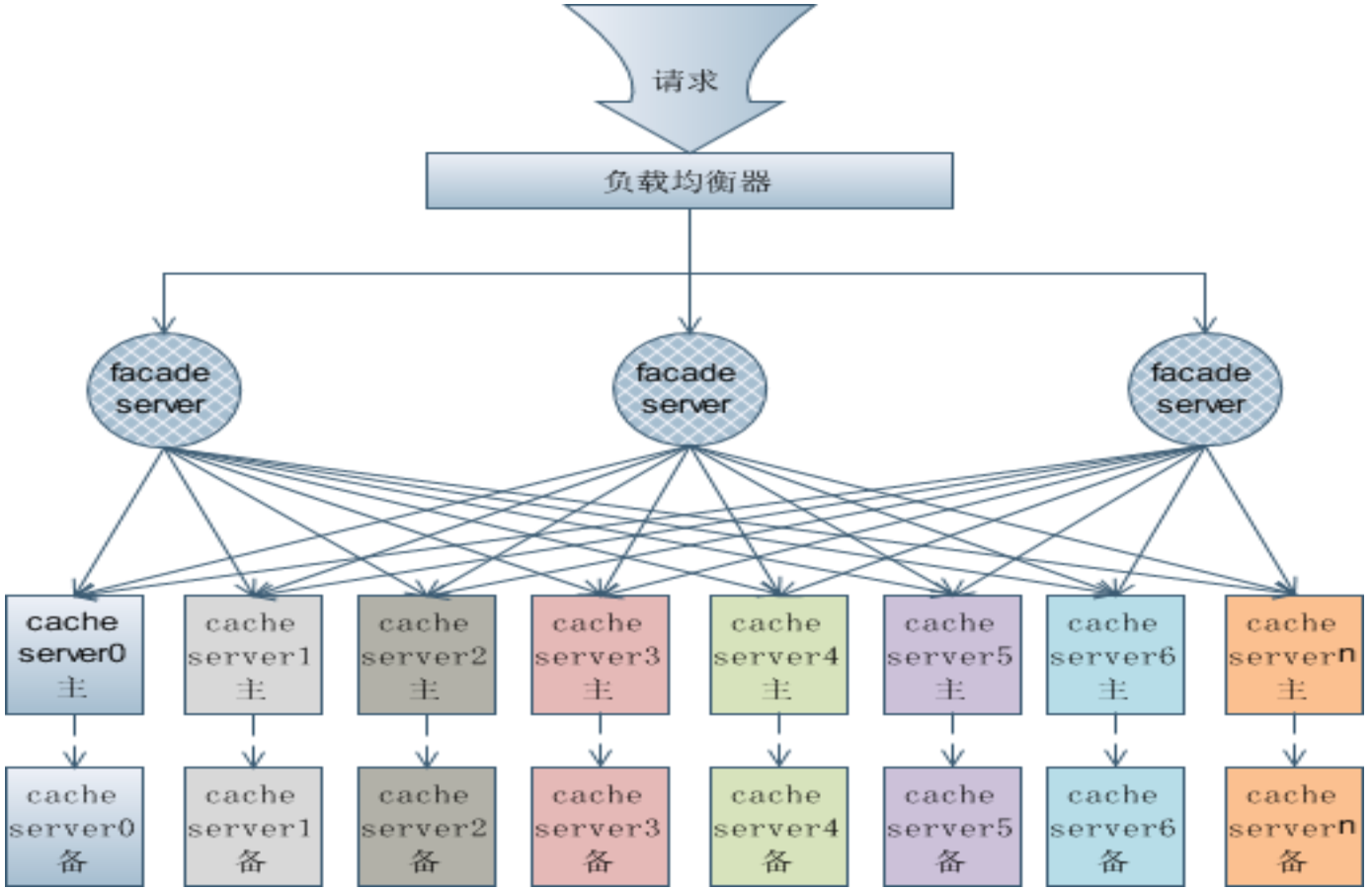
我们需要一个集群管理者管理集群里的服务器，同一个集群中任何一台服务器宕机,其他服务器都能感知. 如果是集群管理者宕机，集群中所有的服务器不能受任何影响，能实时切换到备份管理者上被提供服务。

Fourinone分布式缓存



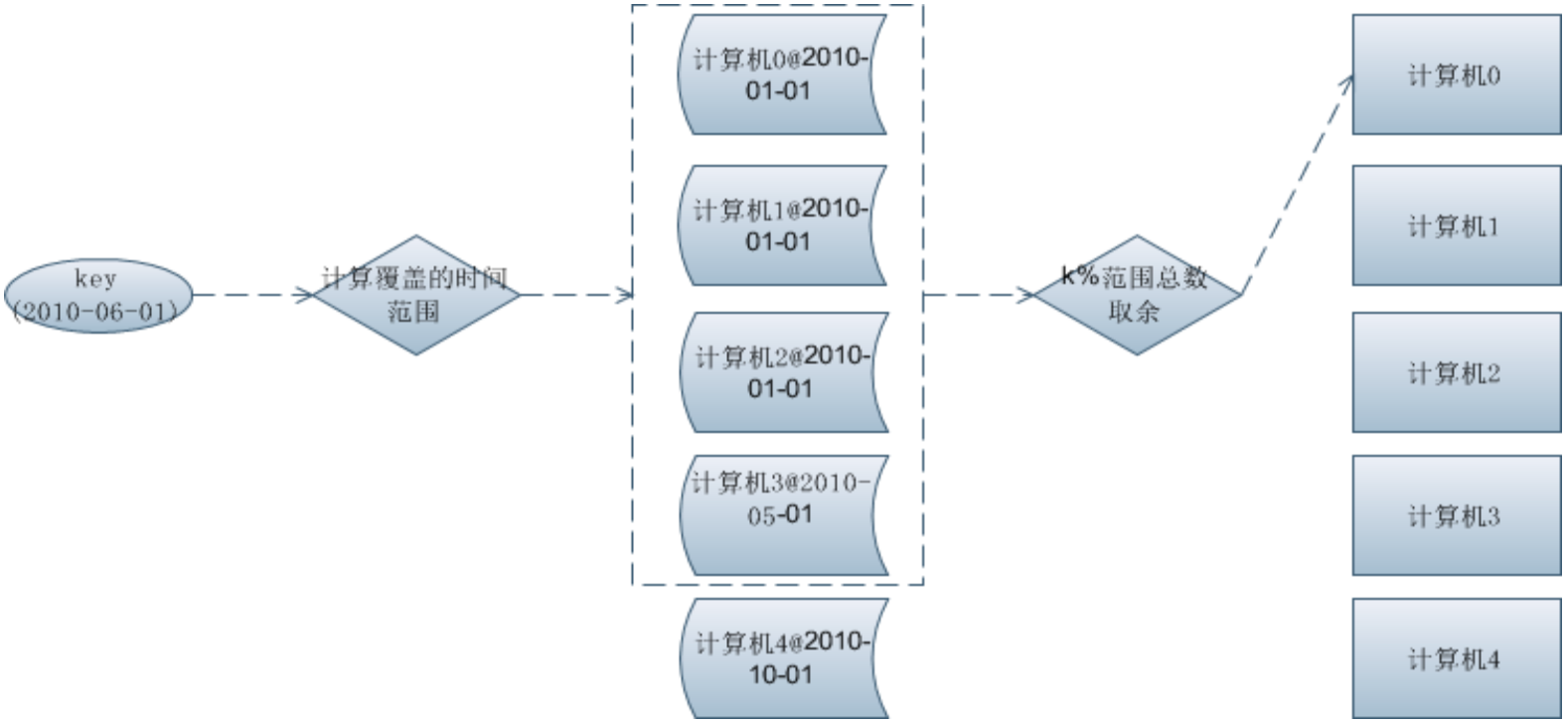
如果对一个中小型的互联网或者企业应用，仅仅利用domain/node进行k/v的存储即可，因为domain/node都是内存操作而且读写锁分离，同时拥有复制备份，完全满足缓存的高性能与可靠性。对于大型互联网应用，高峰访问量上百万的并发读写吞吐量，会超出单台服务器的承受力

Fourinone分布式缓存



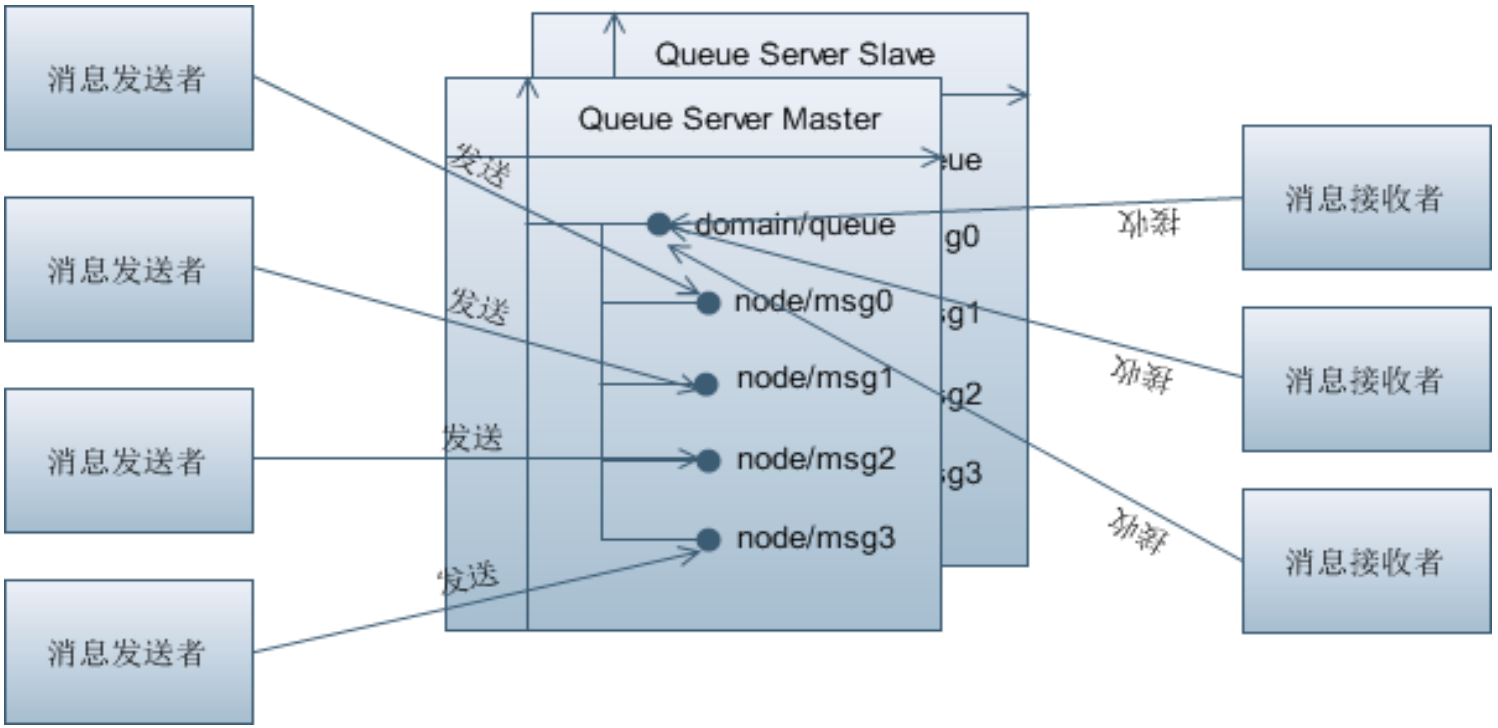
Fourinone提供了facade的解决方案去解决大集群的分布式缓存，利用硬件负载均衡路由到一组facade服务器上，facade可以自动为缓存内容生成key，并根据key准确找到散落在背后的缓存集群的具体哪台服务器，当缓存服务器的容量到达限制时，可以自由扩容，不需要成倍扩容，因为facade的算法会登记服务器扩容时间版本，并将key智能的跟这个时间匹配，这样在扩容后还能准确找到之前分配到的服务器。基于Fourinone可以轻松实现web应用的session功能，只需要将生成的key写入客户端cookie即可。

Key取模设计



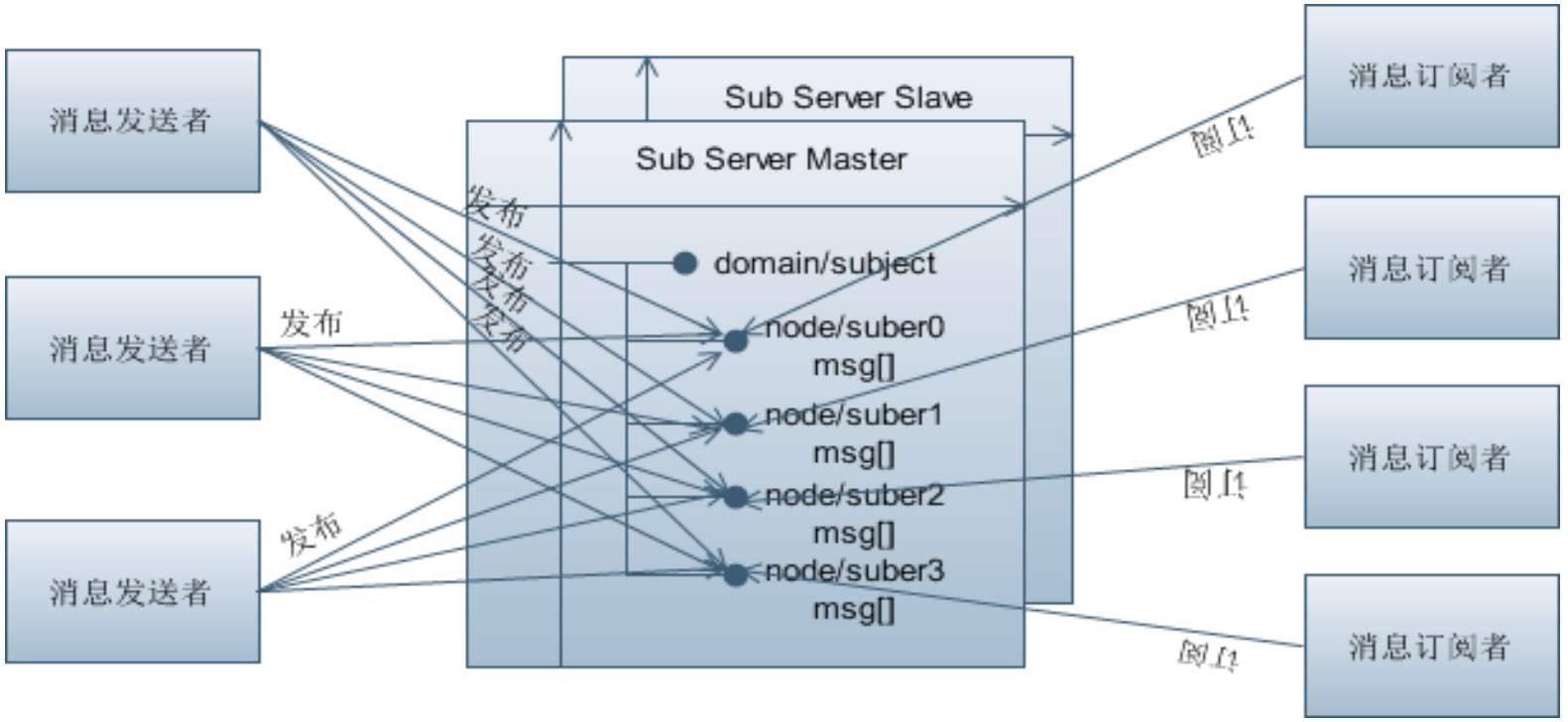
传统key取模这种方式有很大的缺陷，当集群数量扩充时，取模变的不准确，如果要维持准确，通常成倍方式去扩容，会造成成本增加和浪费。本发明通过生成含有日期信息的key，并对集群扩容增加日期配置，通过key和集群配置的日期匹配计算出覆盖范围的机器数，再取模的方式准确得到负载的计算机，对于集群的任意数量的扩容都不会受到影响。

MQ发送接收模式



Fourinone也可以当成简单的mq来使用，发送接收模式实现: 将domain视为mq队列，每个node为一个队列消息，监控domain的变化事件来获取队列消息。

MQ主题订阅模式



将domain视为订阅主题，将每个订阅者注册到domain的node上，发布者将消息逐一更新每个node，订阅者监控每个属于自己的node的变化事件获取订阅消息，收到后清空内容等待下一个消息，多个消息用一个arraylist存放

FourInOne不实现JMS的规范，不提供JMS的消息确认和消息过滤等特殊功能，不过开发者可以基于FourInOne自己去扩充这些功能，包括mq集群。如果需要事务处理可以将多个消息封装在一个集合内进行发送，上面的队列接收者收到消息后删除实际上是一种消息确认方式，也可以将业务逻辑处理完后再进行删除。如果需要持久保存消息可以再封装一层消息发送者，发送前后根据需要进行数据库或者文件持久保存。利用一个独立的domain/node建立队列或者主题的key隐射，再仿照上面分布式缓存的智能根据key定位服务器的做法实现集群管理。

提纲

- 背景:我们需要解决的问题
- 分布式计算*并行计算*云计算
- Hadoop*Zookeeper*Hbase概述
- Fourinone介绍
- Fourinone应用场景:上亿数据排序
- Fourinone 2.0 新功能介绍

API

工头:

WareHouse giveTask(WareHouse inhouse)

实现分配工人要做的任务

WorkerLocal[] getWaitingWorkers(String workerType)

获取集群中等待的工人

WareHouse[] doTaskBatch(WorkerLocal[] wks,
WareHouse wh)

所有工人批量完成给定任务处理

doProject(WareHouse inhouse)

工头开始项目启动

toNext

多个包工头链式处理

工人:

WareHouse doTask(WareHouse inhouse);

实现工头分配的任务

waitWorking(String workerType)

等待工作状态,指定工人类型

Workman[] getWorkerAll();

获取所有的工人

Workman[] getWorkerElse();

获取除自己外的其他工人

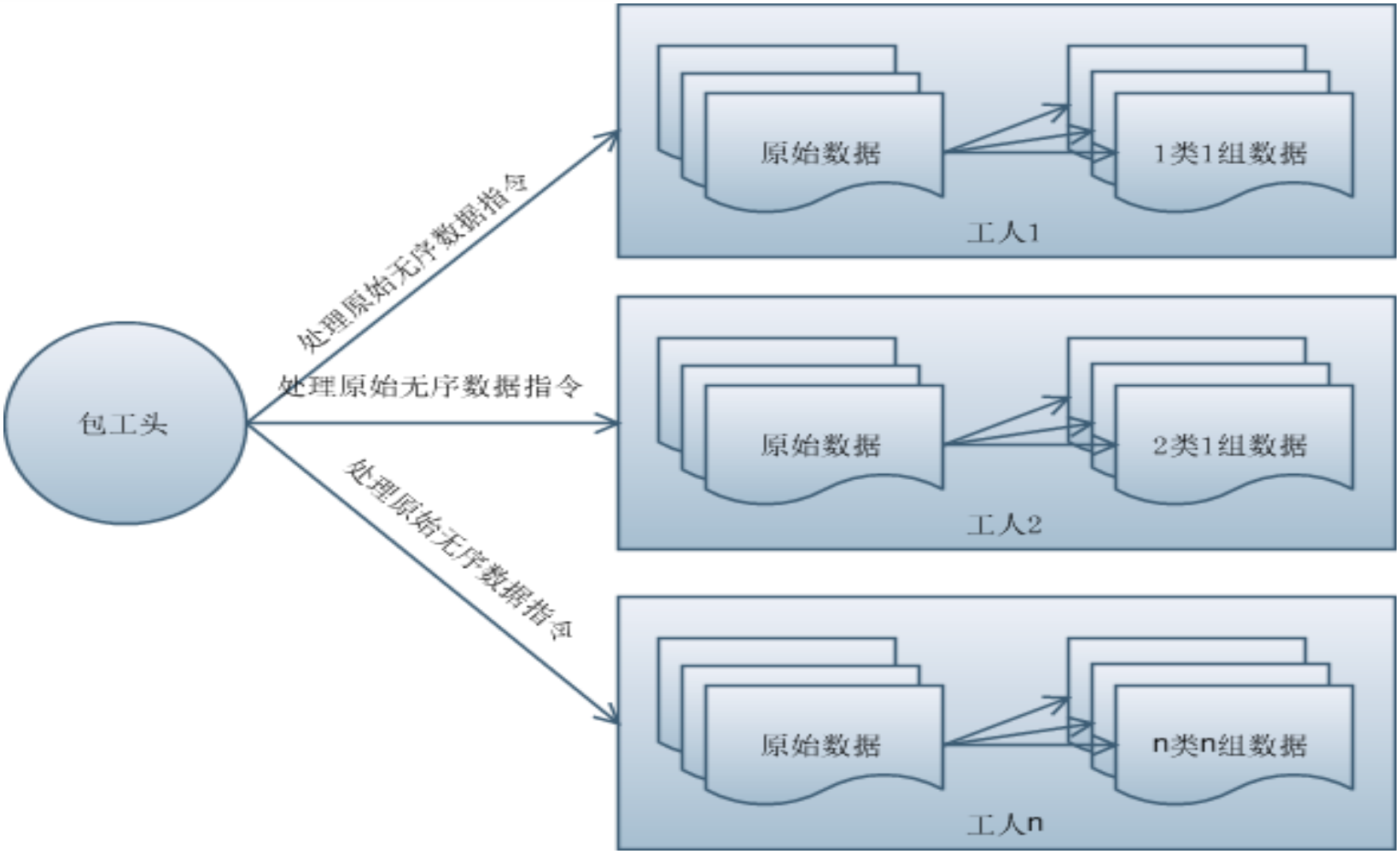
int getSelfIndex();

获取自己在工作中的位置

boolean receive(WareHouse inhouse)

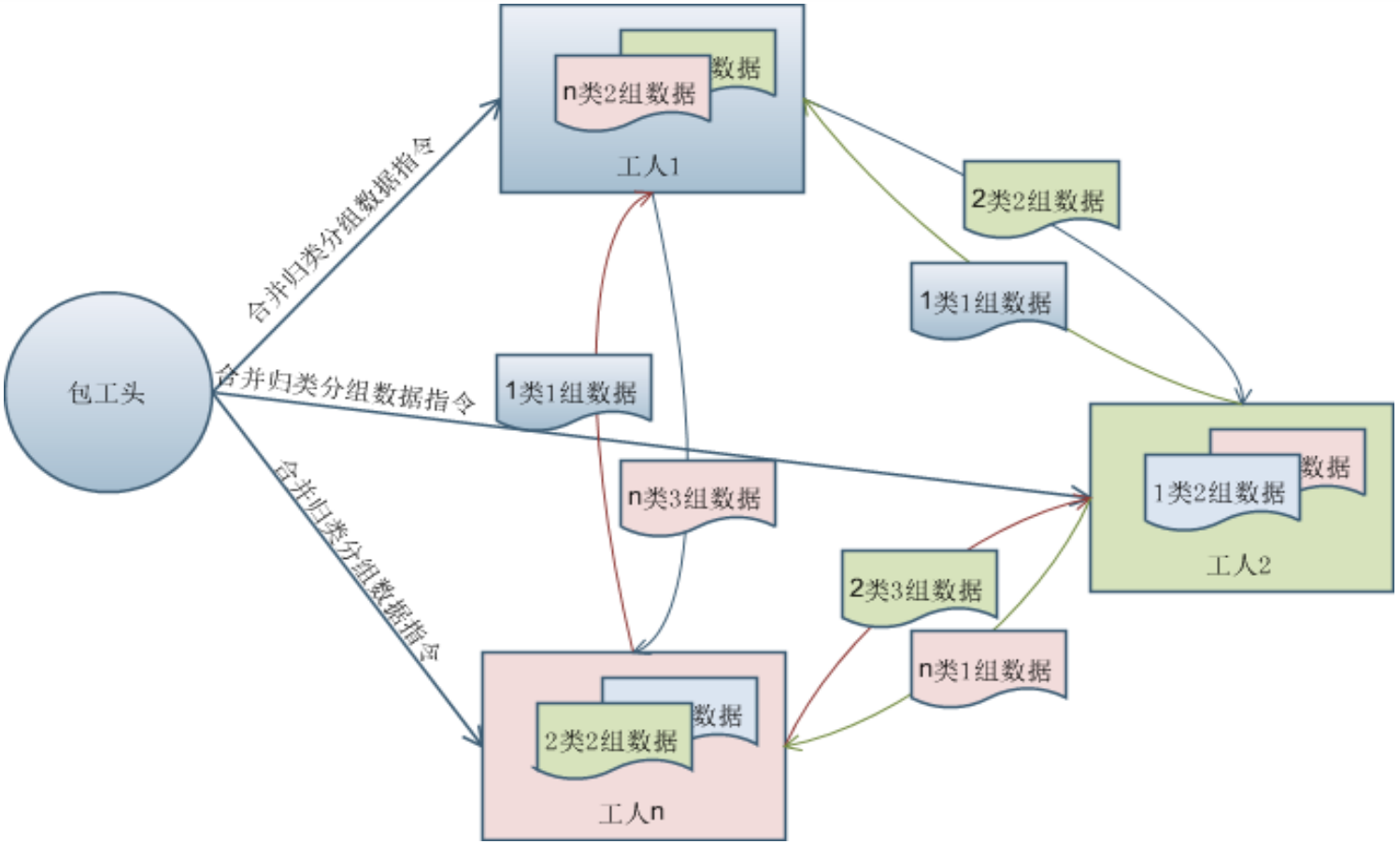
接收来自其他工人的传递

第一个环节:分类



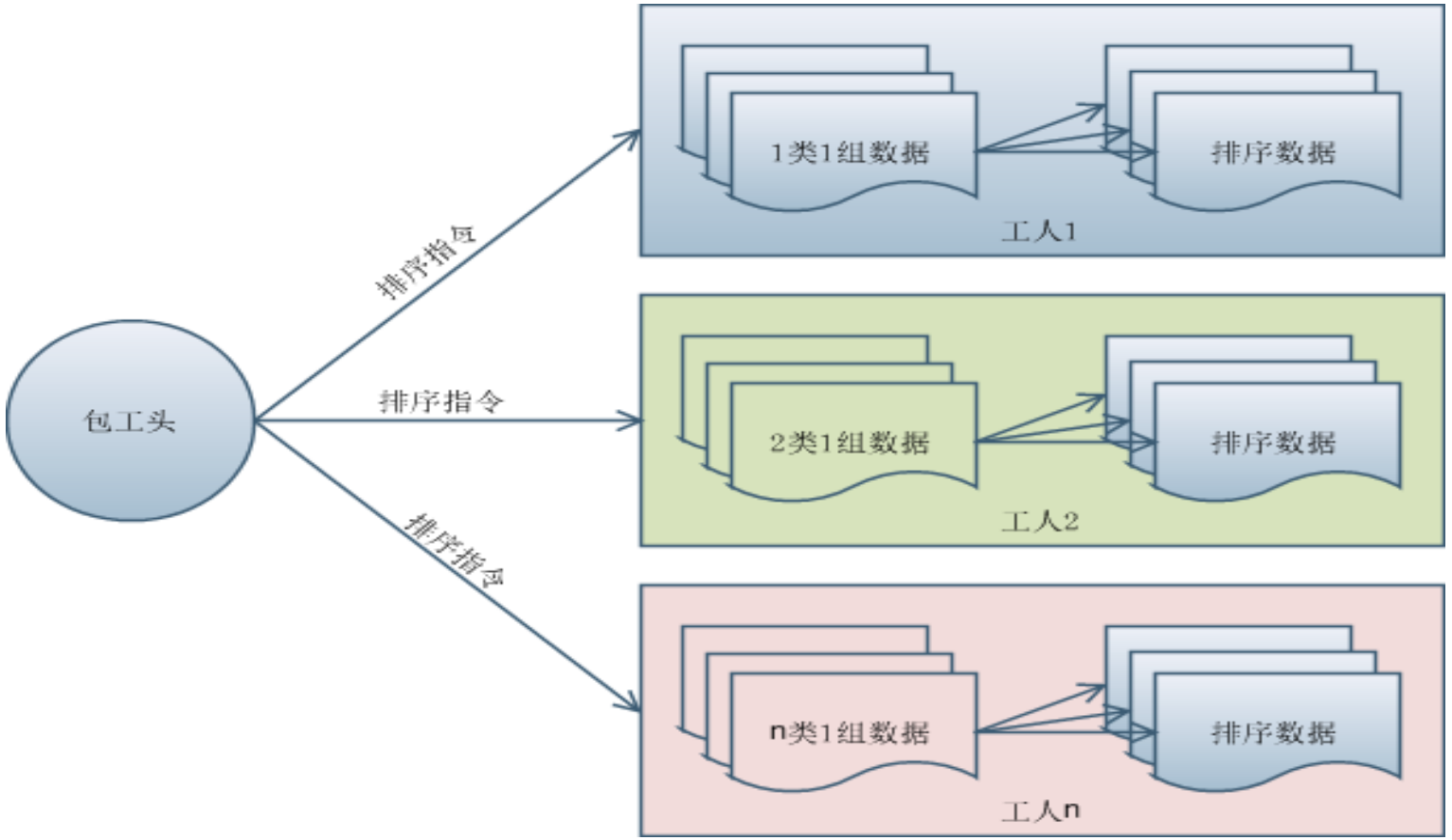
将分散到多台计算机的海量无序数据，按照工人数量和预计处理的每份数据文件大小两个唯度分类，计算出每个工人所属的数据范围

第二个环节:合并



工人彼此之间进行数据合并，合并规则：将属于其他工人的范围数据发给对方，接受对方发给属于自己范围的数据。结果：每个工人机器形成粗的范围的有序数据，但是范围内的数据仍然无序

第三个环节:排序



工人对属于自己范围的数据进行排序，最后得到一个整体原始数据的排序结果，但是它是根据范围分散到不同任务计算机上存放的。完成后返回通知包工头完成排序

Fourinone2.0新功能

一、将集群看做一个操作系统，像本地一样操作远程文件

1、以统一的fttp文件路径访问方式操作集群所有文件

windows：fttp://v020138.sqa.cm4/d:/data/a.log

linux：fttp://v020138.sqa.cm4/home/user/a.log

2、提供对整个集群文件的操作支持，包括元数据访问，添加删除，按块拆分，并行读写，排它读写（按文件部分内容锁定），集群复制，集群所有文件目录浏览器等支持。

3、对文件解析的方便支持（包括按行，按分割符，按最后标识读取）

4、对整形数据的高性能读写支持（ArrayInt比ArrayList存的更多更快）

5、文件操作的两阶段提交和补偿事务支持

二、自动化class和jar包部署

class和jar包只需放在工头机器上， 各工人机器会自动获取并执行，兼容操作系统，不需要进行安全密钥复杂配置

三、网络波动状况下的策略处理，设置抢救期，抢救期内网络稳定下来不判定结点死亡

淘宝网
Taobao.com

提问/交流

开源中国博客地址

http://my.oschina.net/u/177760/

fourinone群：

qq群:1313859

旺旺群：849833763

fourinone@yeah.net

淘宝网
Taobao.com

Thank You