**Apache Hama**

|  |  |
| --- | --- |
| **审核人** |  |
| **重要性** | 中 |
| **紧迫性** | 中 |
| **拟制人** | 张包峰 |
| **提交日期** |  |

**作者：张包峰**

(版权所有,翻版必究)

**修改记录**

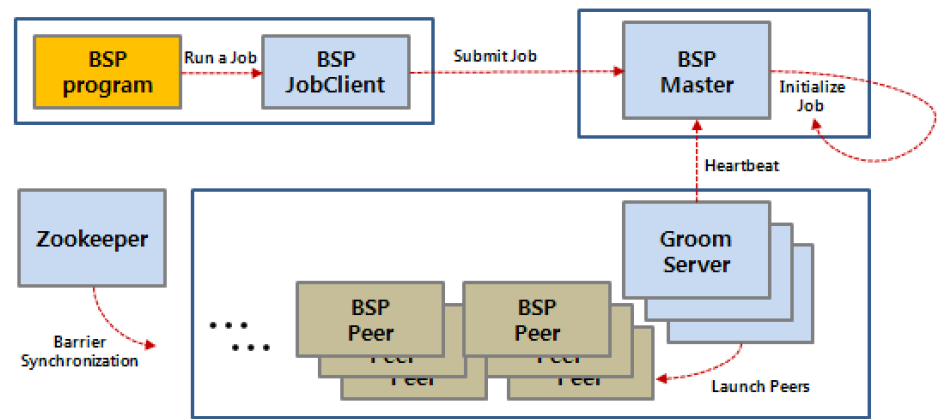
|  |  |  |
| --- | --- | --- |
| **更新时间** | **变更内容** | **变更理由** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

[**作者：张包峰** 1](#_Toc400810233)

[1.1 Overview 4](#_Toc400810234)

## Overview

Hama是基于BSP(BulkSynchronous Parallel)计算技术的并行计算框架，用于大量的科学计算（比如矩阵、图论、网络等）。BSP计算技术最大的优势是加快迭代，在解决最小路径等问题中可以快速得到可行解（http://wiki.apache.org/hama/Benchmarks）。同时，Hama提供简单的编程，比如flexible模型、传统的消息传递模型，而且兼容很多分布式文件系统，比如HDFS、Hbase等。用户可以使用现有的Hadoop集群进行Hama BSP.



### 组成

Hama主要有三部分构成：BSPMaster、GroomServers 和Zookeeper。与Hadoop结构很相似，但没有通信和同步机制的部分。

Hama的集群由一个BSPMaster和多个互不关联的GroomServer作计算结点组成，HDFS和Zookeeper都可以是独立的集群。启动从BSPMaster开始，如果是master会启动BSPMaster、GroomServer两个进程，如果只是计算结点则只会启动GroomServer，启动/关闭脚本都是Master机器远程在GroomServer机器上执行。

BSPMaster 即集群的主，负责了集群各GroomServer结点的管理与作业的调度，相当于Hadoop的JobTracker或HDFS的NameNode。其基本作用如下：

1. 维持Groom服务器状态。

2. 维护supersteps和集群中的计数器。

3. 维护Job的进度信息。

4. 调度作业和任务分配给Groom服务器

5. 分配执行的类和配置，整个Groom服务器。

6. 为用户提供集群控制接口（Web和基于控制台）

GroomServer是一个process，通过BSPMaster启动BSP任务。每一个Groom都有BSPMaster通信，可以通过BSPMaster获取任务，报告状态。GroomServer在HDFS或者其他文件系统上运行，通常，GroomServer与与数据结点在一个物理结点上运行，以保证获得最佳性能。

Zookeeper用来管理BSPPeer的同步，用于实现BarrierSynchronisation机制。在ZK上，进入BSPPeer主要有进入Barrier和离开Barrier操作，所有进入Barrier的Peer会在zk上创建一个EPHEMERAL的node（/bsp/JobID/Superstep NO./TaskID），最后一个进入Barrier的Peer同时还会创建一个readynode(/bsp/JobID/Superstep NO./ready)，Peer进入阻塞状态等待zk上所有task的node都删除后退出Barrier

### BSP Model

BSP(BulkSynchronous Parallel，整体同步并行计算模型)是英国计算机科学家Viliant在上世纪80年代提出的一种并行计算模型。Google发布的一往篇论文(《Pregel: A System for Large-Scale Graph Processing》)使得这一概念被更多人所认识，据说在Google 80%的程序运行在MapReduce上，20%的程序运行在Pregel上。和MapReduce一样，Google并没有开源Pregel，Apache按Pregel的思想提供了类似框架Hama。

Hama BSP 是基于大容量同步并行模型，利用分布式节点计算大量步骤。通常，BSP程序包含一序列的superstep。每一个superstep包含三个步骤：

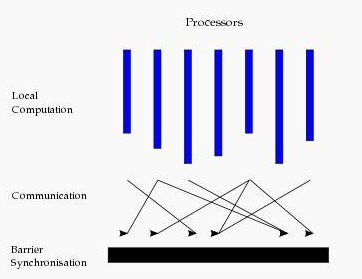
1. Local computation

2. Process communication

3. Barrier synchronization

Hama提供用户自定义的函数bsf()，通过bsf函数，用户可以编写自己的BSP程序，并且BSP程序可以控制整个程序的并行部分，意味着bsf函数不仅仅是程序普通的一部分。

BSP是一种跟MapReduce平行的一种并行计算方法, 如果说MapReduce是把底层的数据传输和分配完全对用户屏蔽了的话, 那BSP就是一种要对底层的数据传输和分配进行手动编程规定的模式了. 这点上跟MPI(一种古老的并行模式)很像.



每个计算节点进行并行计算, 在communication的阶段进行收发, 将运行结果记录在barrier上, 等到所有计算节点运行到barrier,所有的计算节点在继续运行。通过这些原理可以理解为三个步骤：send, sync, receive.

1 )本地计算阶段， 每个处理器只对存储本地内存中的数据进行本地计算。

2 )全局通信阶段， 对任何非本地数据进行操作。

3 )栅栏同步阶段， 等待所有通信行为的结束。

BSP模型相对于其他两种模型而言， 具有如下两个方面的优点：

•MPI 和 PVM两种并行计算模型，依赖于接收和发送的操作对。这样通信方式容易导致上层应用程序产生死锁，而BSP并行计算库是一个程序划分为超步(superstep)，使得死锁不再发生

•BSP模型由于其本身的特点，使得对于程序的正确性和时间的复杂性预测成为可能

### Pregel

Google的Pregel是采用GFS或BigTable进行持久存储，Google的Pregel是一个Master-slave主从结构，有一个节点扮演master角色，其它节点通过name service定位该顶点并在第一次时进行注册，master负责对计算任务进行切分到各节点(也可以自己指定，考虑load balance等因素)，根据顶ID哈希分配顶点到机器(一个机器可以有多个节点，通过name service进行逻辑区分)，每个节点间异步传输消息，通过checkpoint机制实行容错(更高级的容错通过confined recovery实现)，并且每个节点向master汇报心跳(ping)维持状态。