**基于Hadoop大数据的用户行为记录处理系统**

**1.0**

**技术说明书**

华北电力大学-版权所有

**目录**

[一、系统介绍 1](#_Toc5815213)

[1.1 系统设计目标介绍 2](#_Toc5815215)

1[.2 系统应用领域介绍 1](#_Toc5815215)

[二、环境介绍 1](#_Toc5815214)

[2.1 开发环境 1](#_Toc5815215)

[2.1.1 开发平台 1](#_Toc5815216)

[2.1.2 开发语言 1](#_Toc5815217)

[2.2 硬件环境 2](#_Toc5815218)

[2.3 系统部署环境 2](#_Toc5815218)

[三、程序设计 4](#_Toc5815221)

[3.1 程序总体框架设计 4](#_Toc5815222)

[3.2 程序详细模块设计 4](#_Toc5815222)

[3.2.1 程序接口模块设计 4](#_Toc5815216)

[3.2.2 接口数据结构设计 5](#_Toc5815216)

[3.2.3 数据处理模块设计 5](#_Toc5815216)

[四、系统应用结果与计算过程监控 7](#_Toc5815224)

[4.1 数据处理结果](#_Toc5815226) 7

[4.2 Hadoop监控平台](#_Toc5815226) 8

# 一、系统介绍

**1.1 系统设计目标介绍**

此系统为解决目前网课师生交互性不强、教学实时行不佳以及学生观看视频时因重难点之处困惑而操作视频时产生的大量行为痕迹记录数据无法得以反馈利用的缺陷而设计。

**1.2 系统应用领域介绍**

该系统选用当前主流的大数据技术Hadoop平台作为用户行为痕迹处理和记录的平台对海量数据进行处理、存储，其具有处理数据的高效性与运行的稳定性。基于Hadoop技术，将从网络教学网站抓取的用户观看视频的行为痕迹处理、记录，并进行分析，这样能客观并且直观地发现教学中的难度较大与有待改进的环节，将学习视频中存在的问题全面且客观地反馈给网络教学网站和教师，便于教师改进优化教学方案，提升教学质量。其次，它能提醒学生在重点、难点处集中注意力认真研习，更了解自己知识的薄弱环节，便于洞察知识中的漏洞，清楚学习情况，做出科学合理的学习计划。

# 二、环境介绍

## 2.1 开发环境

### 2.1.1 开发平台

基于Linux系统系列中的CentOS搭建完全分布式Hadoop环境，其内核Shell为bash 4.2.46，为运用大数据处理方式处理用户观看视频行为痕迹记录提供开发平台。

### 2.1.2 开发语言

运用Javascript函数与ajax方法将前端获取的数据以JSON格式发送至服务器HDFS中。之后运用Java语言进行研究与开发，编写Job函数与MapReduce函数分布式处理用户的倍速播放的片段与次数以及前进、回退播放的片段起始、结束位置与次数等行为痕迹数据。

## 2.2 硬件环境

(1) CPU：Intel Xeon E5-2620 v2 @ 24x 2.6GHz

(2) 硬盘:56G以上

(3) 内存：15833MiB

## 2.3 系统部署环境

## 该系统的Hadoop节点部署为1个master节点、6个slave节点，其中master与slave1、slave2、slave3以及slave4、slave5、slave6分别部署于三个不同的服务器上，并且slave2,slave3,slave5,slave6均为基于Linux系统虚拟化搭建的DataNode节点。

master节点为NameNode节点，其将会给每一个slave节点布置数据块处理的指令，slave节点为DataNode节点，其为读、写请求提供服务以及按照NameNode的指令执行数据块的创建、删除与复制，并且当完成NameNode节点布置的数据处理任务之后，将存储数据处理结果并把结果反馈给NameNode节点。

NameNode节点与DataNode节点通过心跳机制进行通信，并以此监测各个节点是否正常运行。

NameNode与SecondaryNameNode之间的通信可合并NameNode的edit logs到fsimage文件中以防止当NameNode故障时导致fsimage文件的丢失。其通讯原理图为：

将fsimage拷贝回NameNode上

将edit logs改动更新至fsimage中

fsimage

fsimage

定时查询NameNode上的edit logs

SecondaryNameNode

NameNode

图 1 NameNode与SecondaryNameNode通讯原理图

注：fsimage 为NameNode启动时对整个文件系统的快照

edit logs为NameNode启动后对文件系统的改动序列

系统节点分布架构图如下所示：

Slave5 Slave6

Vmware

DataNode DataNode

Slave2 Slave3

Vmware

DataNode DataNode

Master

NameNode

SecondaryNameNode

DataNode

Linux

Slave1

DataNode

Linux

Slave4

DataNode

Linux

Master与Slave通信以Slave2为例

：任务调度

：结果反馈

：心跳机制

：NameNode与SecondaryNameNode通讯

图 2系统节点分布架构图

# 程序设计

## 3.1 程序总体框架设计

前端将用户观看视频时所产生的行为痕迹记录数据例如用户的倍速播放的片段与次数以及前进、回退播放的片段起始、结束位置与次数等数据转化为JSON格式并通过Ajax方法将数据颗粒化写入HDFS文件系统中，之后在Hadoop框架上经过Map、Shuffle和Reduce 三个阶段对用户行为的数据进行计算、积累，获得每个视频以秒为单位的用户累计观看次数，并做成图表，并将最新数据处理结果写入文件系统HDFS中，并反馈给前端，以曲线的形式显示在播放器进度条上，给当前用户作为参考。

程序总体设计架构图如下所示：

JSON格式数据

反馈给前端

HDFS

缓冲区

Reduce阶段

Map阶段

Shuffle阶段

## 图 3程序总体框架设计图

## 3.2 程序详细模块模块设计

## **3.2.1 程序接口模块设计**

使用JavaScript实现对用户视频观看行为痕迹的记录，并且将记录的数据转换为JSON格式，使用ajax方法发送请求将JSON文件写入HDFS中。

## **3.2.2 接口数据结构设计**

程序以每秒为一个时间段划分视频，每秒均有一组（start，end，speed）标记记录，分别代表每个时间段的开始时间，结束时间与倍速播放的速值，用户在观看视频时将鼠标抬起将开启一个新时间段并赋值给start；按下将结束此时间段并赋值给end；speed的初值为1.0，观看视频时将每各时间段current\_speed的值赋值给speed。

因此每段数据结构为：

视频URL/用户ID/(start,end,speed)(start,end,speed)..(start,end,speed)

以此记录用户观看视频时的行为痕迹数据。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名称 | 数据类型  （精度范围） | 允许为空Y/N | 唯一Y/N | 默认值 |
| 视频URL | string | N | Y |  |
| 用户ID | string | N | Y |  |
| start | float | N | N | 0.0 |
| end | float | N | N | 0.0 |
| speed | float | N | N | 1.0 |

图 4数据结构设计图

注：URL与ID无默认值，其值视具体视频信息与用户信息而定。

**3.2.3 数据处理模块设计**

数据处理程序分为Map、Shuffle和Reduce 三个阶段。

Map阶段：各个节点将前端提交的每一条用户行为数据，按照多段起始时间及对应倍速，对数组进行加权计算，从而得到多个观看次数数组，数组的key值为视频URL，value值为某个用户观看该视频的过程中，对每秒视频的根据观看倍速进行加权的数值。

Shuffle阶段：MapReduce的关键环节，将key值即视频的URL相同的信息分到一类，并交给同一个Reducer处理，从而对信息进行整理和分类。

Reduce阶段：将key值相同的信息中的value值累加在一起，更新每个视频所有用户观看行为的总记录，并将结果写入Hadoop文件系统HDFS中。

系统的数据处理程序算法设计图如下所示:

**Map**

**阶段**

URL1/ID1/start1/end1/speed1/start2/end2/speed2/......../start10/end10/speed10

URL1/ID2/start1/end1/speed1/start2/end2/speed2/......../start13/end13/speed13

URL2/ID3/start1/end1/speed1/start2/end2/speed2/......../start15/end15/speed15

........(此时数据格式为多个字符串)

URL1|ID1|1|1|1|1|2|2|2|2|2|2|1.5|1.5|1.5|1.5|..........|3|3|3|3|3|3|1|1|1|1|1|1|1|1|1|2|2|2|2|2|2|

URL1|ID2|2|2|2|2|2|2|2|3|3|3|3|3|3|3|4|4|4|1|1|1|1|...............|5|5|5|5|5|5|5|5|5|3|3|3|3|3|3|3|

URL2|ID3|1|1|1|1|1|1|1.5|1.5|1.5|2|2|2|2|2|2|2|2|1|1|1|1|........|0.5|0.5|0.5|2|2|2|2|2|2|2|4|4|4|

........(此时数据格式已转化为多个数组)

**Shuffle**

**阶段**

**URL1**|ID1|1|1|1|1|2|2|2|2|2|2|1.5|1.5|1.5|1.5|..........|3|3|3|3|3|3|1|1|1|1|1|1|1|1|1|2|2|2|2|2|2|

**URL1**|ID2|2|2|2|2|2|2|2|3|3|3|3|3|3|3|4|4|4|1|1|1|1|...............|5|5|5|5|5|5|5|5|5|3|3|3|3|3|3|3|

**URL2**|ID3|1|1|1|1|1|1|1.5|1.5|1.5|2|2|2|2|2|2|2|2|1|1|1|1|........|0.5|0.5|0.5|2|2|2|2|2|2|2|4|4|4|

........(Hadoop机制将key相同的信息归为一类)

**Reduce**

**阶段**

**URL1**|ID1|1|1|1|1|2|2|2|2|2|2|1.5|1.5|1.5|1.5|..........|3|3|3|3|3|3|1|1|1|1|1|1|1|1|1|2|2|2|2|2|2|

**+**

**URL1**|ID2|2|2|2|2|2|2|2|3|3|3|3|3|3|3|4|4|4|1|1|1|1|...............|5|5|5|5|5|5|5|5|5|3|3|3|3|3|3|3|

**=**

**URL1|3|3|3|3|3|4|4|4|4|5|5|4.5|4.5|4.5|5.5|5.5|4|4|4|4|4|...............|8|8|8|8|8|8|6|6|6|6|5|5|5|5|5|5|**

**URL2**|ID3|1|1|1|1|1|1|1.5|1.5|1.5|2|2|2|2|2|2|2|2|1|1|1|1|........|0.5|0.5|0.5|2|2|2|2|2|2|2|4|4|4|

+........(将key值相同的value累加在一起)

图 5 数据处理算法设计图

**四、系统应用结果与计算过程监控**

## 4.1 数据处理结果

## 将前端接口提交的24M大小的JSON格式的用户行为痕迹记录数据转换为视频每秒的播放次数并颗粒化存储到HDFS中，结合用户数据进行统计分析，获得不同用户、不同视频的学习情况总结。

处理结果如下图所示：

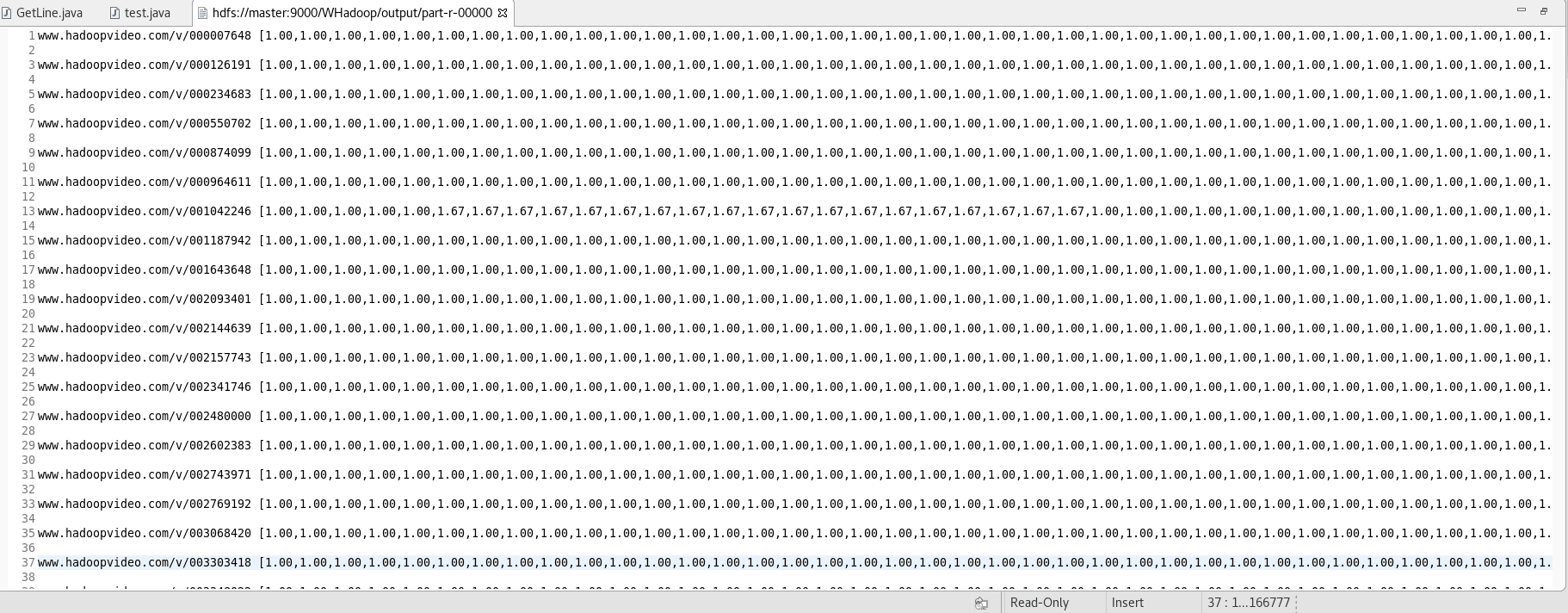


图 6 数据处理结果

在Hadoop框架中应用分布式计算处理速度如下图所示：

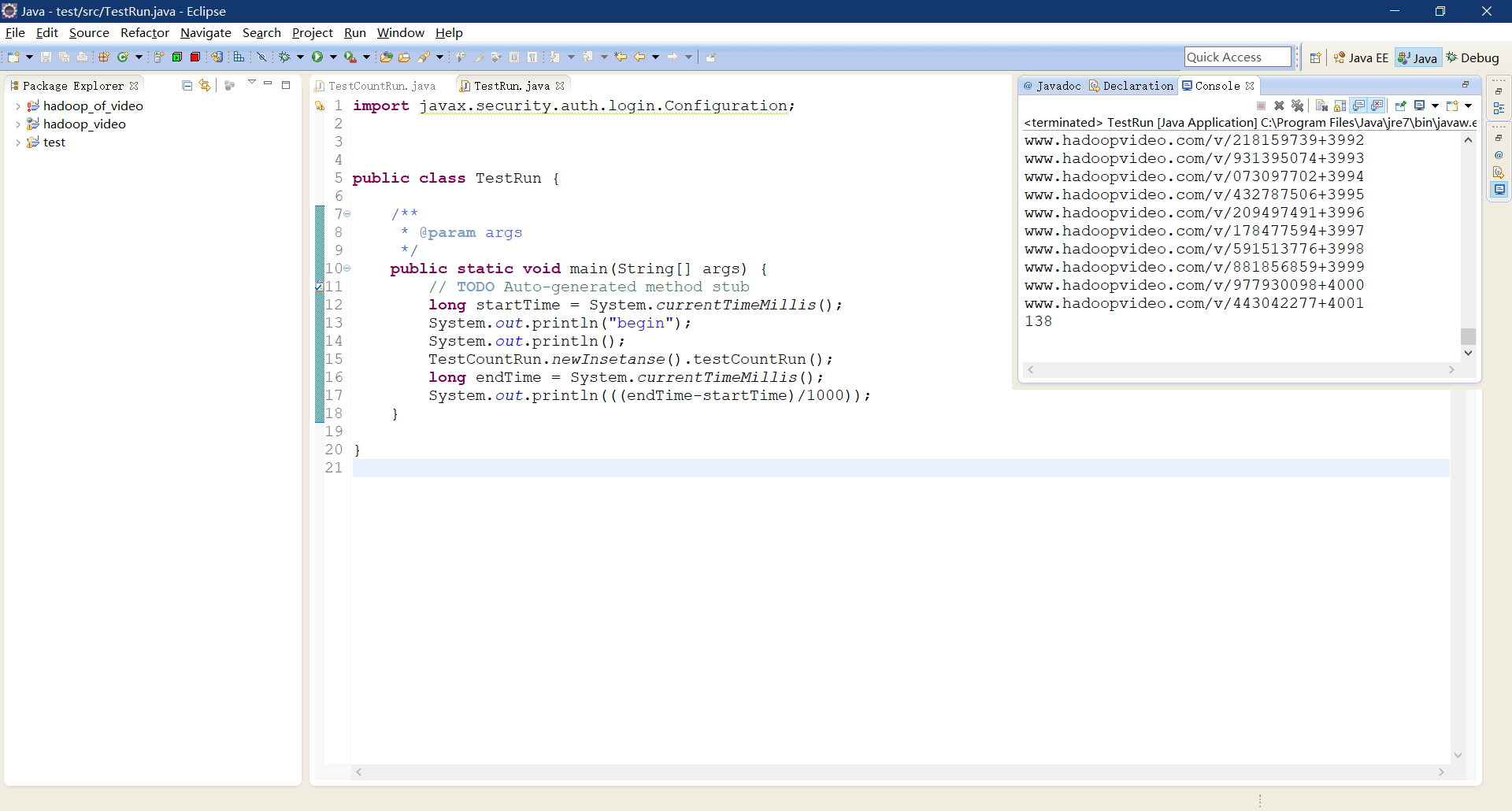


图 7 数据处理速度结果

## 4.2 Hadoop监控平台

Hadoop监控是系统管理的重要内容，可通过监控检测集群在何时未提供所期望的服务以及处理数据时所处理的数据块数、CPU与内存的开销、调用的节点数等信息，以此排查故障或优化配置。

于master节点上启动完全分布式Hadoop，同时可通过jps命令监测各节点启动进程并分别运用命令行方式与Web方法检测DataNode运行状况，以此验证完全分布式Hadoop集群的成功启动，并且可以检测集群所提供的服务是否符合期望。

1. Master节点中NameNode、ResourceManager、SeconderyNameNode、JobHistoryServer进程：



图 8 NameNode、ResourceManager、SeconderyNameNode、JobHistoryServer进程监控图

1. slave节点中DataNode与NodeManager进程：



图 9 DataNode、NodeManager进程监控图

3）命令行方式与Web方式检测DataNode启动状况：

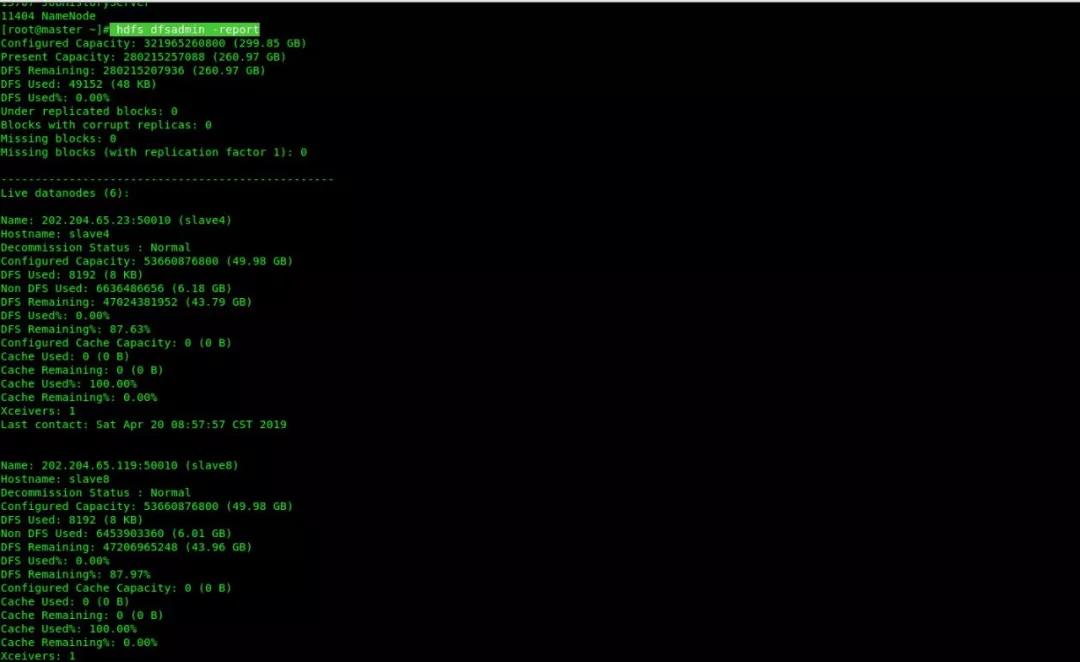


图 10 命令行方式的DataNode启动监控图

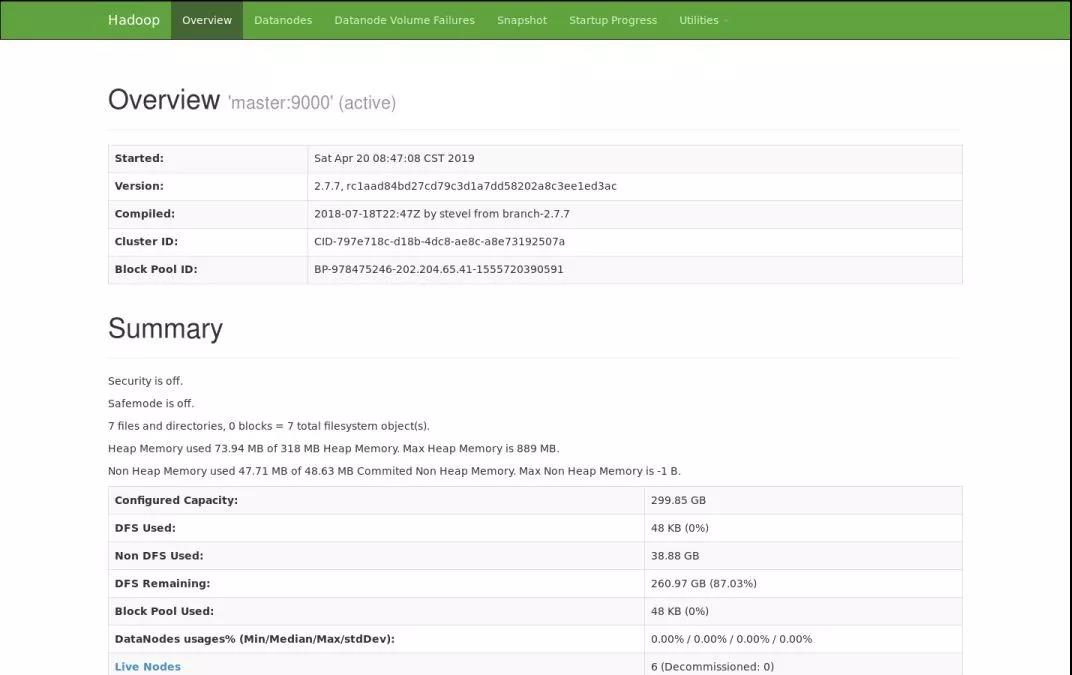


图 11 Web方式的DataNode启动监控图