Tachyon--以内存为核心的开源分布式存储系统

0

平台 (http://www.csdn.net/tag/平台/news) 数据 (http://www.csdn.net/tag/数据/news)
开源 (http://www.csdn.net/tag/开源/news) 系统 (http://www.csdn.net/tag/系统/news)
内存 (http://www.csdn.net/tag/内存/news)

Tachyon (http://tachyon-project.org/)是一个以内存为核心的开源分布式存储系统，也是目前发展最迅速的开源大数据项目之一。Tachyon为不同的大数据计算框架（如Apache Spark，Hadoop MapReduce, Apache Flink等）提供可靠的内存级的数据共享服务。此外，Tachyon还能够整合众多现有的存储系统（如Amazon S3, Apache HDFS, RedHat GlusterFS, OpenStack Swift等），为用户提供统一的、易用的、高效的数据访问平台。本文首先向读者介绍Tachyon项目的诞生背景和目前发展的情况；然后详解Tachyon系统的基本架构以及目前一些重要的功能；最后，分享一个Tachyon在百度大数据生产环境下的几个应用案例。

1.Tachyon简介

随着技术的发展，内存的吞吐量在不断地提高，单位容量的内存价格在不断降低，这为“内存计算”提供可能。在大数据计算平台领域，采用分布式内存计算模式的Spark验证了这一点。Spark相比于MapReduce大大提升了大数据的计算性能，受到了业界和社区的广泛关注。然而，还是有很多问题在计算框架层难以解决，如：不同的Spark应用或不同计算框架（Spark，MapReduce，Presto）间仍需通过基于磁盘的存储系统（如HDFS，Amazon S3等）交换数据；当Spark计算任务崩溃，JVM缓存的数据会丢失； JVM中大量缓存的数据增加了Java垃圾回收的压力。

Tachyon最初出现是为了有效地解决了上述问题，它计划构建一个独立的存储层来快速共享不同计算框架的数据，实现方式上将数据置于堆外(off-heap)内存以避免大量垃圾回收开销。例如，对应Spark应用而言，可以带来以下作用：

1. 不同Spark应用，甚至不同计算平台上的应用需要数据共享时，通过Tachyon进行内存读写，避免缓慢的磁盘操作。
2. 使用Tachyon进行数据缓存，当Spark任务崩溃，数据仍缓存在Tachyon内存中，任务重启后能够直接从Tachyon中读取数据。
3. 多个Spark应用理论上甚至可以共享同一份Tachyon缓存的数据，避免内存资源的浪

费，减轻Java垃圾回收的压力。

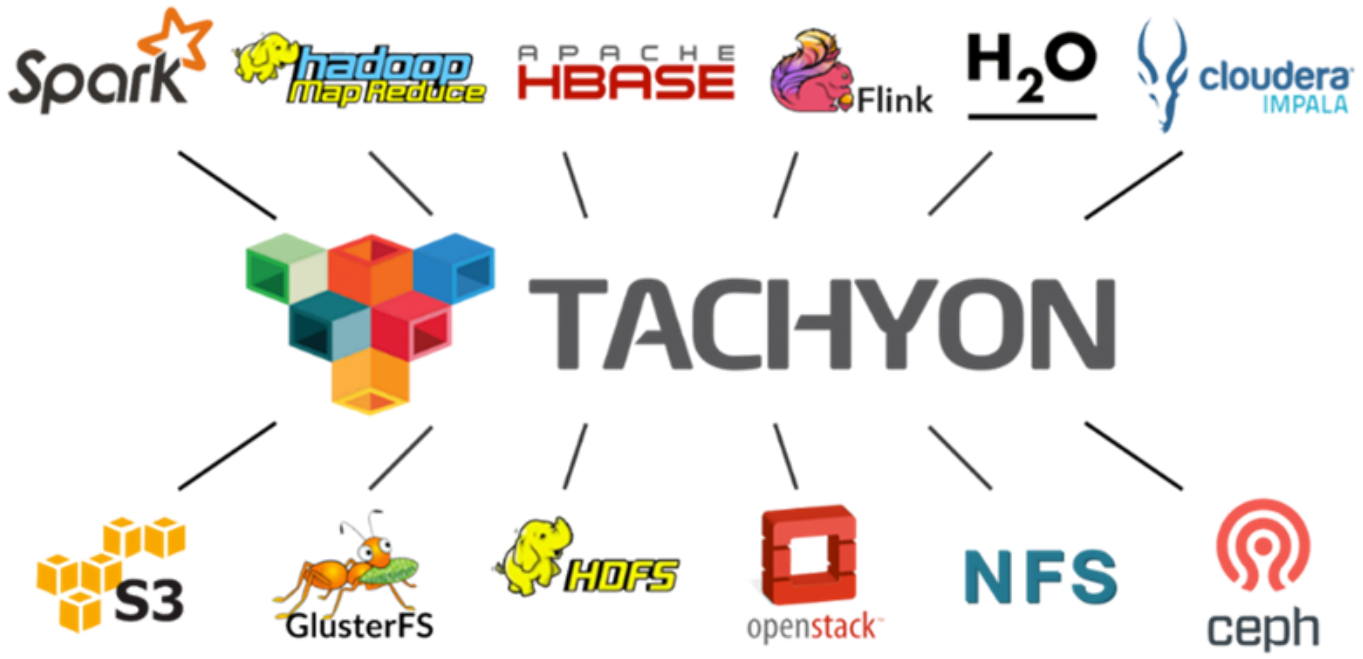


图1. Tachyon在生态系统的位置

图1给出了Tachyon部署时所处的位置。Tachyon被部署在计算平台之下和现有的存储系统之上，能够在不同计算框架间共享数据。同时，现有的海量数据不需要进行迁移，上层的计算作业仍能通过Tachyon访问到底层存储平台上的数据。Tachyon作为一个以内存为中心的中间存储层，不仅能极大地提升上层计算平台的性能，还能充分利用不同特性的底层存储系统，更可以有效地整合两者的优势。

由李浩源b的研究项目（该实社区不断壮大，已经机构的200多人参与于此同时，Tachyon的核心创建者和开发人员创立了Tachyon Nexus (http://www.tachyonnexus.com/)公司，其中不乏UC Berkeley、CMU等博士以及Google, Palantir, Yahoo!等前员工。 2015年3月美国华尔街日报 (http://blogs.wsj.com/venturecapital/2015/03/17/andreessen-horowitz-invests-7-5m-in-big-data-startup-tachyon/)报道了Tachyon Nexus获得硅谷著名风投Andreessen Horowitz 的750万美元A轮投资。

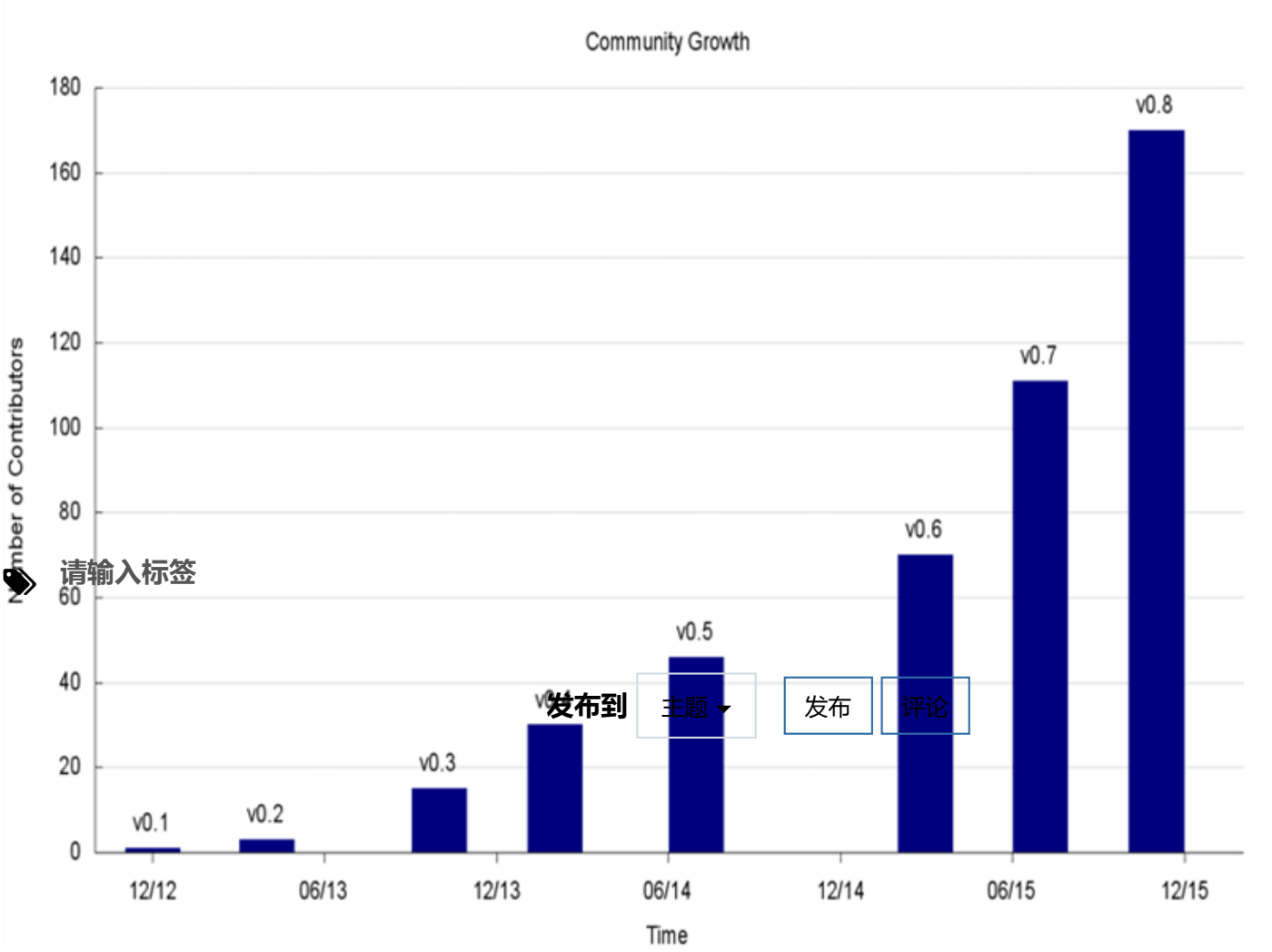


图2. Tachyon项目贡献者的增长情况

在学术界， 国内的南京大学PASA大数据实验室 (http://pasa-bigdata.nju.edu.cn/)一直积极关注并参与到Tachyon项目的开发中，共向Tachyon社区贡献了100多个PR，近300次commit，包括为Tachyon实现性能测试框架tachyon-perf，增加LFU、LRFU等多个替换策略，改进WebUI页面，以及其他一些性能优化的工作。此外，我们还撰写了Tachyon相关的中文博客 (http://blog.csdn.net/u014252240/article/category/2755319)，以便中文读者和用户能够更深入地了解和使用Tachyon。

在工业界，百度也把Tachyon运用到其大数据系统中， Tachyon在过去一年中稳定的支持着百度的可交互式查询业务，令百度的交互式查询提速30倍。在验证了Tachyon的高性能以及可靠性后，百度在内部使用Tachyon的0.9版成功部署了1000个worker的世界最大Tachyon集群，总共提供50TB的内存存储。此集群在百度内部已经稳定运行了一个月，也验证的Tachyon的可扩展性。于此同时，百度的另外一个Tachyon部署中用Tachyon层次化数据管理了2PB数据。

2.Tachyon系统架构

这一章中我们简介Tachyon系统的基本架构，包括Tachyon的基本组件及其功能。

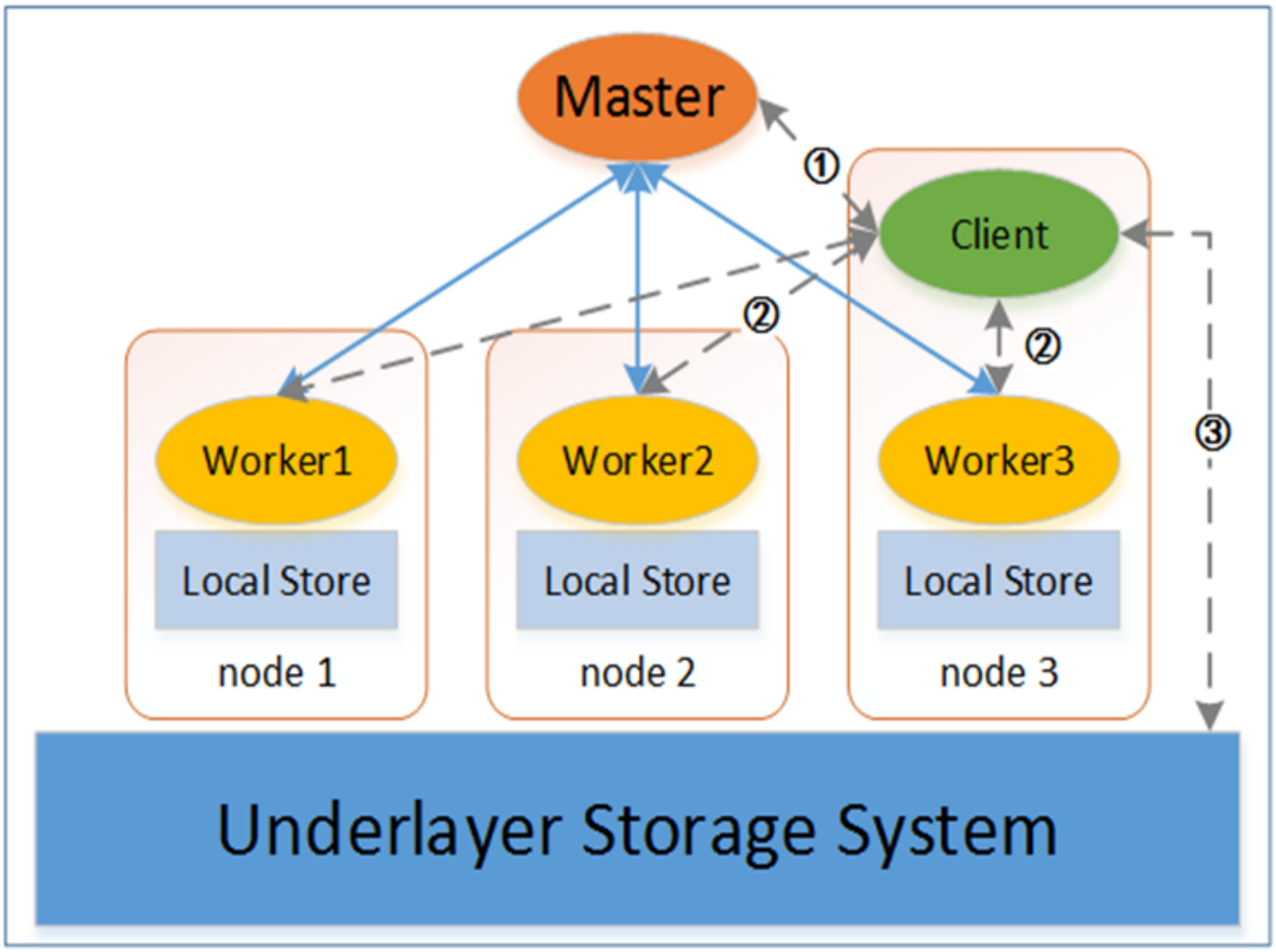


图3. Tachyon的系统架构

图2是Tachyon系统的基本架构，主要包括4个基本组件：Master、Worker和Client，以及可插拔的底层存储系统（Underlayer Storage System）。每个组件的具体功能职责如下：

- Tachyon Master主要负责管理两类重要信息。第一，Tachyon Master中记录了所有数据文件的元数据信息，包括整个Tachyon命名空间（namespace）的组织结构，所有文件和数据块的基本信息等。第二，Tachyon Master监管着整个Tachyon系统的状态，包括整个系统的存储容量使用情况，所有Tachyon Worker的运行状态等。

- Tachyon Worker负责管理本地节点上的存储资源，包括内存、SSD和HDD等。Tachyon中的所有数据文件被划分为一系列数据块，Tachyon Worker以块为粒度进行存储和管理，如：为新的数据块分配空间、将热数据块从SSD或HDD移至内存、实时或定期备份数据块到底层存储系统。同时，Tachyon Worker定时向Tachyon Master发送心跳（heartbeat）以告知自身的状态信息。

- Tachyon Client是上层应用访问Tachyon数据的入口。访问过程可以包括如下几步：①Client向Master询问数据文件的基本信息，包括文件位置，数据块大小等；②Client尝试从本地Worker中读取对应数据块，若本地不存在Worker或者数据块不在本地Worker中，则尝试从远程Worker中读取；③若数据还未被缓存到Tachyon中，则Client会从底层存储系统中读取对应数据。此外，Tachyon Client会向所有建立连接的Tachyon Master和Tachyon Worker定时发送心跳以表示仍处于连接租期中，中断连接后Tachyon Master和Tachyon Worker会回收对应Client的临时空间。

- 底层存储系统既可以被Tachyon用来备份数据，也可以作为Tachyon缓存数据的来源，上层应用在使用Tachyon Client时也能直接访问底层存储系统上的数据。底层存储系统保证了Tachyon Worker在发生故障而崩溃后不会导致数据丢失，同时也使得上层应用在迁移到Tachyon的同时不需要进行底层数据的迁移。目前Tachyon支持的底层存储系统有HDFS，GlusterFS，Amazon S3，OpenStack Swift以及本地文件系统，且能够比较容易地嵌入更多的现有存储系统。

在实际部署时， Tachyon Master通常部署在单个主节点上（Tachyon也支持多个节点上部署Tachyon Master，并通过使用ZooKeeper来防止单点故障）；将Tachyon Worker部署在多个从节点；Tachyon Client和应用相关，可以位于任何一个节点上。

3.Tachyon的特色功能

本节我们简介Tachyon面向上层应用的特色功能。

3.1 支持多种部署方式

作为大数据系统中的存储层，Tachyon为用户提供了不同的启动模式、对资源管理框架的支持、以及目标运行环境，能够部署多种大数据平台环境中：

- 启动模式：以正常模式启动单个Tachyon Master；以高级容错模式启动多个Tachyon Master，并使用ZooKeeper进行管理；
- 资源管理框架：以Standalone方式直接运行在操作系统之上；运行在Apache Mesos之上；运行在Apache Hadoop Yarn之上；
- 目标运行环境：部署在本地集群环境中；部署在Virtual Box虚拟机中；部署在容器（如Docker）中；部署在Amazon EC2云平台上（Tachyon社区正在开发支持Tachyon部署在阿里云OSS上(http://tachyon-project.org/documentation/master/Configuring-Tachyon-with-OSS.html)）

用户可以自由选择不同的启动模式、资源管理框架和目标运行环境，Tachyon为多种组合都提供了相应的启动脚本，能够很方便地将Tachyon部署在用户的环境中。

3.2 层次化存储

Tachyon的层次化存储充分利用了每个Tachyon Worker上的本地存储资源，将Tachyon中的数据块按不同热度存放在了不同的存储层中。目前Tachyon所使用的本地存储资源包括MEM（Memory，内存）、SSD（Solid State Drives，固态硬盘）和HDD（Hard Disk Drives，磁盘）。在Tachyon Worker中，每一类存储资源被视作一层（Storage Tier），每一层又可以由多个目录（Storage Directory）组成，并且用户可以设置每个存储目录的容量。

在读写Tachyon数据时，分配器（Allocator）负责为新的数据块选择目标存储目录，替换器（Evictor）负责将冷数据从内存剔至SSD和HDD，同时将热数据从SSD和HDD提升至内存中。目前分配器所使用的分配策略包括Greedy、MaxFree和RoundRobin。替换器所使用的替换策略包括Greedy、LRU/PartialLRU、LRFU。额外地，Tachyon还为用户提供了Pin功能，支持用户将所需要的数据始终存放在内存中。关于如何配置Tachyon层次化存储，可以进一步参考Tachyon官方文档 (http://tachyon-project.org/documentation/Tiered-Storage-on-Tachyon.html)。

3.3 灵活的读写机制

为了充分利用多层次的存储资源和底层存储系统，Tachyon为用户提供了不同的读写类型（ReadType/WriteType）API，用于灵活控制读写数据时的行为方式，不同的读写类型及其含义如表1所示。

表1. 读写类型（ReadType/WriteType）的取值及其含义

类型	值	含义
读类型 ReadType	NO_CACHE	读数据时不进行额外操作
	CACHE	如果数据不在本地存储中，则将读取的数据块缓存在本地存储中
	CACHE_PROMOTE	如果数据不在本地内存中，则将读取的数据块缓存在本地内存中
写类型 WriteType	MUST_CACHE	将数据仅写入本地存储中
	CACHE_THROUGH	将数据同时写入本地存储与底层存储系统中
	THROUGH	将数据仅写入底层存储系统中

除了上述的读写类型外，Tachyon还提供了另一套控制方式：TachyonStorageType和UnderStorageType，用于分别控制在Tachyon存储和底层存储系统上的读写行为，具体取值及其含义如表2所示。实际上，这种控制方式是Tachyon-0.8之后新增的，控制粒度更细，功能也更多，因此推荐用户采用这种方式控制读写行为。

表2. TachyonStorageType/UnderStorageType的值及其含义

类型	值	含义
TachyonStorageType	STORE	读数据时，如果数据不在本地存储中，则将读取的数据块缓存在本地存储中； 写数据时，将数据写入本地存储中
	NO_STORE	读数据时，不进行额外操作； 写数据时，不将数据写入本地存储中
	PROMOTE	读数据时，如果数据不在本地内存中，则将读取的数据块缓存在本地内存中； 写数据时，将数据写入本地存储中
UnderStorageType	SYNC_PERSIST	写数据时，将数据同步写入底层存储系统中
	NO_PERSIST	写数据时，不将数据写入底层存储系统中
	ASYNC_PERSIST	写数据时，由 Tachyon Worker 后台将数据异步写入底层存储系统中

3.4 文件系统层的Lineage容错机制

在Tachyon中，Lineage表示了两个或多个文件之间的世系关系，即输出文件集B是由输入文件集A通过怎样的操作得到的。有了Lineage信息后，在文件数据意外丢失时，Tachyon就会启动重计算作业，根据现有的文件重新执行同样的操作，以恢复丢失的数据。图3给出了一个Lineage示例，文件集A通过一个Spark作业生成文件集B；文件集C通过另一个Spark作业生成文件集D；B和D作为同一个MapReduce作业的输入，输出为文件集E。那么，如果文件集E意外丢失，并且没有备份，那么Tachyon就会重新启动对应的MapReduce作业，再次生成E。

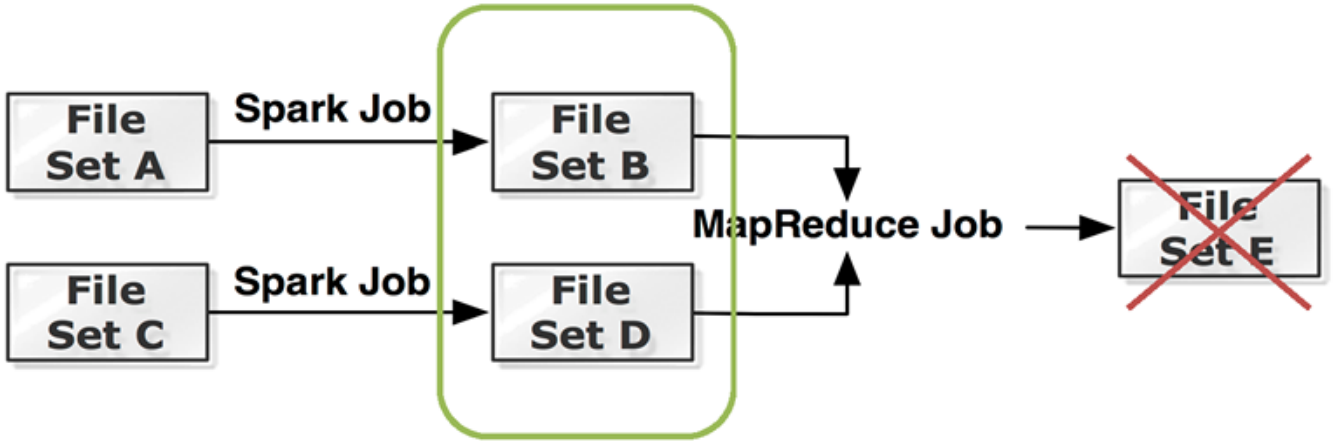


图4. Tachyon的Lineage机制

3.5 统一命名空间

对于Tachyon的用户而言，通过Tachyon提供的接口所访问到的是Tachyon文件系统的命名空间。当用户需要访问Tachyon以外的文件和数据时，Tachyon提供了Mount接口，能够将外部存储系统的文件或目录挂载到Tachyon的命名空间中。这样用户就能够在统一的Tachyon命名空间中，使用相同或者自定义的路径，访问其他存储系统上的文件和数据。

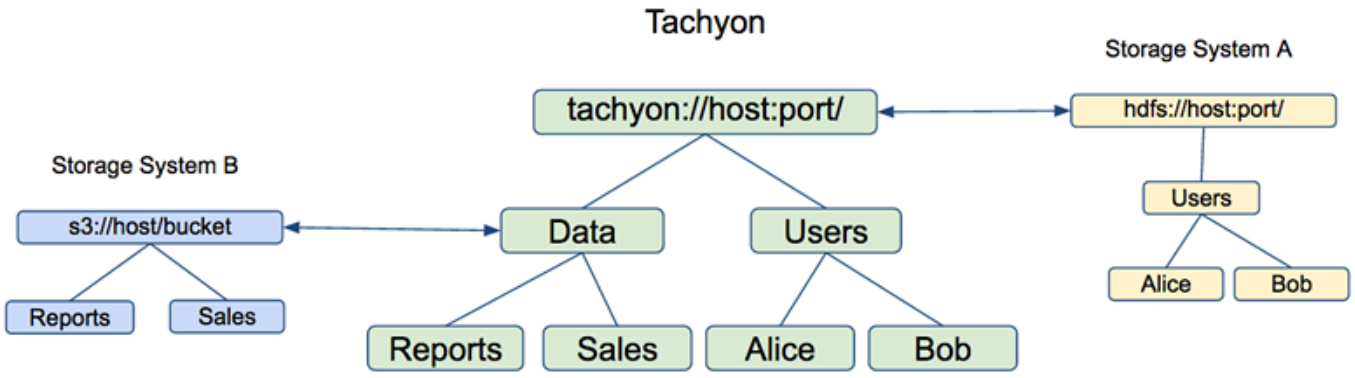


图5. Tachyon的统一命名空间

3.6 HDFS兼容接口

在Tachyon出现之前，诸如Hadoop MapReduce以及Apache Spark的应用大多使用HDFS、Amazon S3等存储文件。Tachyon为这些应用提供了一套HDFS兼容的接口（确切地说，是兼容了org.apache.hadoop.fs.FileSystem的接口），用户可以在不改动应用源码的情况下，通过以下3个步骤，将目标文件系统更改为Tachyon：

> 1.将对应版本Tachyon Client的jar包添加至运行环境的CLASSPATH中；
> 2.添加Hadoop配置项<“fs.tachyon.impl”，“tachyon.hadoop.TFS”>；
> 3.将原先的”hdfs://ip:port/file/X”路径更改为”tachyon://ip:port/file/X”。

通常，用户可以结合使用“HDFS兼容接口”和“统一命名空间”这两个特性，将原先的大数据应用直接运行在Tachyon之上，而不需要进行任何代码和数据的迁移。

3.7 丰富的命令行式工具

Tachyon自带了一个名为 “tfs”的命令行工具，能够让用户以命令行的方式与Tachyon交互，而不需要编写源码来查看、新建、删除Tachyon文件。例如：

```
$ ./bin/tachyon tfs ls /                                  //查看根目录下所有文件
$ ./bin/tachyon tfs mkdir /foo                            //新建文件夹
$ ./bin/tachyon tfs copyFromLocal /local/bar /foo/bar     //将文件从本地拷至 Tachyon
$ ./bin/tachyon tfs ls /foo                               //查看目标文件夹下所有文件
$ ./bin/tachyon tfs fileinfo /foo/bar                     //查看目标文件信息
```

“tfs” 工具提供的全部命令使用方式详见Tachyon官方文档 (http://tachyon-project.org/documentation/Command-Line-Interface.html)。

3.8 方便管理的WebUI

除了“tfs”工具外，Tachyon还在Tachyon Master和每个Tachyon Worker节点上启动了一个网页管理页面，用户可以通过浏览器打开对应的WebUI（默认为http://:19999和http://:30000）。WebUI上列举了整个Tachyon系统的基本信息、所有Tachyon Worker的运行状态、以及当前Tachyon系统的配置信息。同时，用户可以直接在WebUI上浏览整个Tachyon文件系统、预览文件内容、甚至下载具体的某个文件。

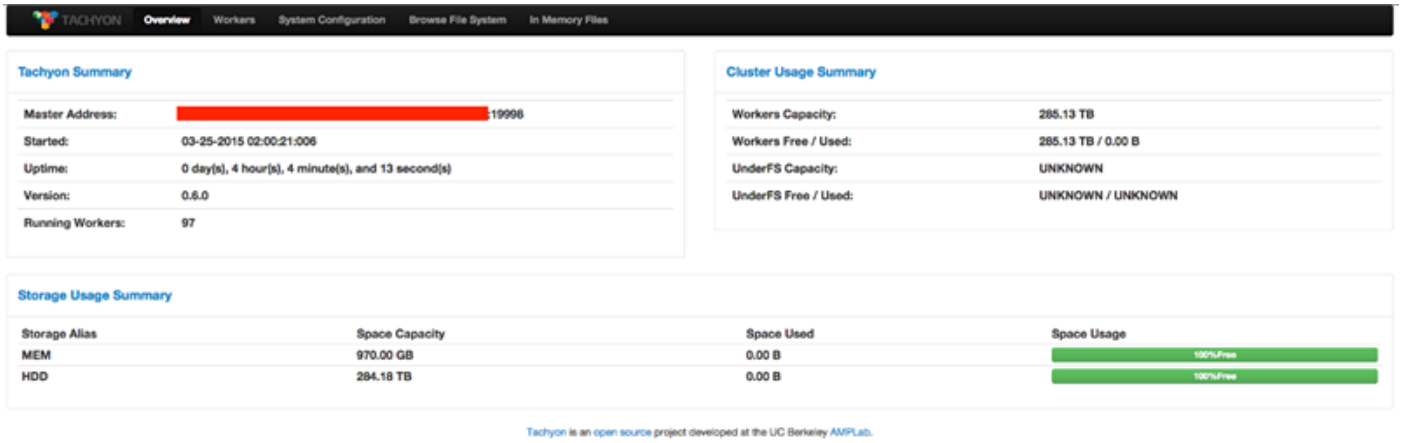


图6. Tachyon的WebUI

3.9 实时指标监控系统

图7. Tachyon监控的实时指标（WebUI模式、JSON格式）

对于高级用户和系统管理人员，Tachyon提供了一套实时指标监控系统，实时地记录和管理了Tachyon中一些重要的统计信息，包括存储容量使用情况、现有Tachyon文件数、对文件的操作次数、现有的数据块数、对数据块的操作次数、总共读写的字节数等。根据用户的配置，这些指标能够以多种方式进行输出：标准控制台输出、以CSV格式保存为文件、输出到JMX控制台、输出到Graphite服务器以及输出到Tachyon的WebUI。

3.10支持Linux FUSE

Tachyon-FUSE (http://tachyon-project.org/documentation/master/Mounting-Tachyon-FS-with-FUSE.html)是Tachyon最新开发版的新特性，由Tachyon Nexus和IBM共同主导开发。在Linux系统中，FUSE（Filesystem in Userspace，用户空间文件系统）模块使得用户能将其他文件系统挂载到本地文件系统的某一目录下，然后以统一的方式进行访问。Tachyon-FUSE的出现使得用户同样可以将Tachyon文件系统挂载到本地文件系统中。通过Tachyon-FUSE，用户/应用可以使用访问本地文件系统的方式来访问Tachyon。这更加方便了用户对Tachyon的管理和使用，以及现有基于FUSE接口的应用通过Tachyon进行内存加速或者数据共享。

4.Tachyon在百度大数据平台的应用案例

在百度，我们从2014年底开始关注Tachyon。当时我们使用Spark SQL进行大数据分析工作，由于Spark是个基于内存的计算平台，我们预计绝大部分的数据查询应该在几秒或者十几秒完成以达到交互查询的体验。然而，我们却发现实际查询几乎都需要上百秒才能完成，其原因在于我们的计算资源与数据仓库可能并不在同一个数据中心。 在这种情况下，我们每一次数据查询都可能需要从远端的数据中心读取数据，由于数据中心间的网络带宽以及延时的问题，导致每次查询都需要较长的时间（>100秒）才能完成。更糟糕的是，很多查询的重复性或相似性很高，同样的数据很可能会被查询多次，如果每次都从远端的数据中心读取，必然造成资源浪费。

为了解决这个问题，在一年前我们借助Tachyon管理远程及本地数据读取和调度，尽量避免跨数据中心读数据。 当Tachyon被部署到Spark所在的数据中心后，每次数据冷查询时，我们还是从远端数据仓库拉数据，但是当数据再次被查询时， Spark将直接从同一数据中心的Tachyon中读取数据， 从而提高查询性能。在我们的环境和应用中实验表明：如果是从非本机的Tachyon读取数据的话，耗时降到10到15秒，比原来的性能提高了10倍； 最好的情况下，如果从本机的Tachyon读数据，查询仅需5秒，比原来的性能提高了30倍， 效果很明

显。除了性能的提高，更难能可贵的是Tachyon运行稳定，在过去一年中很好的支持着百度的交互式查询业务， 而且社区在每一版迭代更新中都不断提供更多的功能以及不断提高系统的稳定性，让业界对Tachyon系统更有信心。

在过去一个月，百度在为大规模使用Tachyon做准备，验证Tachyon的可扩展性。我们使用Tachyon的最新版成功部署了1000个worker的Tachyon集群，在本文完成时这应该是世界最大的Tachyon集群。此集群总共提供超过50TB的内存存储，在百度内部已经稳定运行了一个月，现在有不同的百度业务在上面试运行以及压力测试。在百度的图搜变现业务上，我们与社区合作在Tachyon上搭建了一个高性能的Key/Value存储，提供线上图片服务。同时由于图片直接存在Tachyon里，我们的线下计算可以直接从Tachyon中读取图片。 这使得我们将线上以及线下系统整合成一个系统，既简化了开发流程，也节省了存储资源，达到了事半功倍的效果。本文篇幅有限，期待在后期给大家详细介绍百度是1000 worker的Tachyon 集群的实用案例，包括如何使用Tachyon整合线上线下的存储资源等。

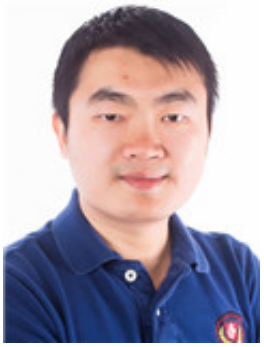
5.结语

作为一个以内存为中心、统一的分布式存储系统，Tachyon极大地增强了大数据生态中存储层的功能。虽然Tachyon项目相对还比较年轻，但已经很成熟稳定，并且已经在学术界以及工业界取得了成功。随着整个计算机产业的发展，内存变的越来越便宜，在计算集群中可使用的内存容量会不断增长，我们相信Tachyon也必将会在大数据平台中发挥越来越重要的作用。

现在Tachyon项目发展迅速，更多的功能也在逐步得到完善，应用前景也颇为广阔。Tachyon正不断地在支持更多的底层存储系统（特别地，社区中已经有人正在实施支持阿里云OSS存储系统以及百度开放云平台，这对国内的用户和开发者来说是个很好的机会）；同时Tachyon也在实现安全性相关的支持，以充分满足业界生成环境的需要；更进一步地，Tachyon目前更多地被视为文件系统，而作为一个统一存储系统，Tachyon也将支持更多的数据结构，以满足不同计算框架的需要。在本文完成时Tachyon已经准备发布下一版，有兴趣的读者们可以多关注Tachyon，到社区里进行技术讨论以及功能开发。

作者简介：

顾荣 (http://weibo.com/njugurong)，Tachyon项目核心开发者之一，南京大学PASA大数据实验室博士生。曾在Microsoft Research Aisa, Intel, Baidu以及Transwarp从事过大数据平台和算法相关的实习工作。目前主要研究兴趣为大数据计算和存储平台、分布式机器学习。



刘少山 (http://www.linkedin.com/in/shaoshanliu)，Tachyon项目核心开发者之一，百度公司美国研发中心高级架构师。加州大学欧文分校计算机博士。曾在LinkedIn, Microsoft, Microsoft Research, INRIA, Intel以及Broadcom工作。目前主要从事百度大数据，深度学习,以及异构计算平台架构与开发。

（责编/魏伟，关注Docker和OpenStack，投稿请联系微信“k15751091376”或者邮箱weiwei@csdn.net）

(http://geek.csdn.net/user/publishlist/karamos)
CSDN魏伟 (http://geek.csdn.net/user/publishlist/karamos)
发布于 云计算 (http://geek.csdn.net/forum/49) 2016-01-15 09:25

评论

已有14条评论 最新

业余草 (http://geek.csdn.net/user/publishlist/xmt1139057136) 2016-01-15 14:10
(http://geek.csdn.net/user/publishlist/xmt1139057136)
看起来很高大上
0 回复 投诉

AllenSJ (http://geek.csdn.net/user/publishlist/sujing1012) 2016-01-15 12:56
(http://geek.csdn.net/user/publishlist/sujing1012)
已分享学习
0 回复 投诉

yuzhiweilai (http://geek.csdn.net/user/publishlist/yuzhiweilai) 2016-01-15 11:27

很不错的分布式存储系统，我其实在基于“内存”的框架飞速发展的时候，值得关注。

0 回复 投诉

伊蔚瀚 (http://geek.csdn.net/user/publishlist/u013673608) 2016-01-15 11:16

对于经常处理大数据的人来说，必须接触众多相关但配置又各异的系统或平台足够让人头疼！而Tachyon的出现使得这一问题得到极大缓解。不管怎么说，在”便于使用”这一目标上，Tachyon迈进了一大步！期待Tachyon更好的发展

0 回复 投诉

saltylin711 (http://geek.csdn.net/user/publishlist/saltylin711) 2016-01-15 11:11

文章写得不错！对百度的提速案例很感兴趣，期待更多的相关分享！！

0 回复 投诉

huangshengbin426 (http://geek.csdn.net/user/publishlist/huangshengbin426) 2016-01-15 11:10

很好的材料，学习了。近几年一直在关注，Tachyon确实在不断地发展、加入新的模块和功能，旨在为大数据处理提供一个高性能、高容错、基于内存的开源分布式存储解决方案。很有潜力和前景，会一直关注的！

0 回复 投诉

summerdg (http://geek.csdn.net/user/publishlist/summerdg) 2016-01-15 11:08

Andreessen Horowitz 都投资了，看来潜力不小啊！关注中。。。

0 回复 投诉

看马和平使者 (http://geek.csdn.net/user/publishlist/teddybear1314) 2016-01-15 11:06

最近在学习Tachyon，请问Tachyon的数据放在内存里如何实现容错？，多副本的话对内存消耗很大，希望有大牛可以给我解答一下

0 回复 投诉

Branjeoyang (http://geek.csdn.net/user/publishlist/yangwen_jia) 2016-01-15 11:05

挺不错的。最近的项目正准备使用tachyon

0 回复 投诉

XL_BHWFY (http://geek.csdn.net/user/publishlist/u014735026) 2016-01-15 11:01

很好的开源项目，一直有关注，文章讲的挺好~

0 回复 投诉