曾经的淘宝云梯分布式计算平台的分析与研究

2014 春计算机- XXX

“淘宝网，淘我喜欢”，这句响亮的广告语，几乎对每一个网民来讲都不陌生，淘宝网（taobao.com）是中国深受欢迎的网购零售平台，目前拥有近 5 亿的注册用户数，每天有超过 6000 万的固定访客，同时每天的在线商品数已经超过了 8 亿件，平均每分钟售出 4.8 万件商品[1]。2013 年双 11 期间，交易额突破 1 亿只用了 55 秒，达到 10 亿用了 6 分 7 秒，50 亿用了 38 分钟，总交易额 350.19 亿。另外，2013 年的双 11 在 40 分钟内，淘宝服装类目就突破 10 亿元销售额。1 分钟有 9.8 万个包裹，与去年相比，包裹数同比增长 1.7 倍。处在销售额榜中有 7 位的全部为服装企业，领先的为优衣库、杰克琼斯、GXG、欧时力等品牌[2]。那么，在如此海量的数据中，淘宝是如何快速的存储和挖掘出这些有用的数据呢？答案就是淘宝的分布式计算平台-云梯。

淘宝云梯是一个 Hadoop 集群，为淘宝日常运维做关键的技术支撑，通过对云梯系统的数据进行分析，有效的发现系统问题，从而尽快的满足客户需求，也为决策提供了依据，图 1 是淘宝网的云梯整体架构图[3]。淘宝云梯是一个很复杂，很庞大的分布式系统，涉及到的技术非常多，也非常广，下面仅从数据的异构性，可伸缩性和透明性等方面进行分析。

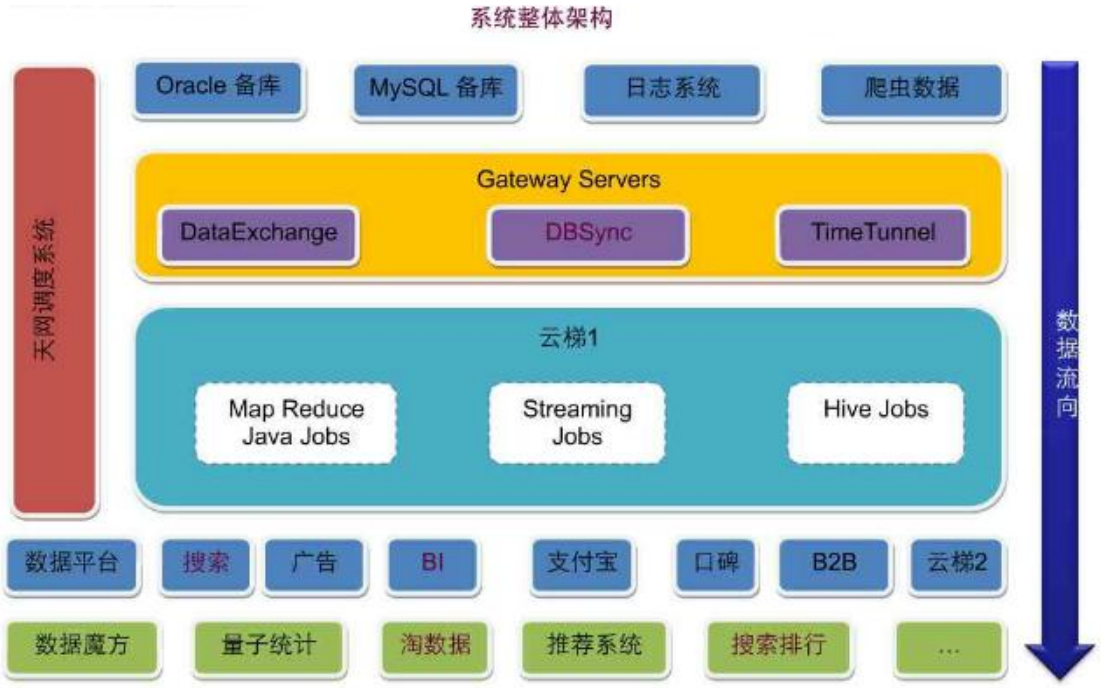

图 1 淘宝网云梯整体架构图

一 异构性

1.数据来源的异构性

从图 1 我们可以看出来，数据的来源也是多种多样的，有 Oracle 数据库，有 MySQL 数据库，还有一些其他的日志数据，爬虫数据等等。有关系型数据库如 Oracle，也有非关系型数据如日志等。如何统一处理来源不同，结构化和非结构化数据并存的数据，就显得尤为重要。

目前成熟的数据解决数据异构性的导入导出工具比较多，但是一般都只能用于数据导入或者导出，并且只能支持一个或者几个特定类型的数据库。这样带来的一个问题是，如果我们拥有很多不同类型的数据库/文件系统(Mysql/Oracle/Rac/Hive/Other…)，并且经常需要在它们之间导入导出数据，那么我们可能需 要开发/维护/学习使用一批这样的工具(jdbcdump/dbloader/multithread/getmerge+sqlloader /mysqldumper…)。而且以后每增加一种库类型，我们需要的工具数目将线性增长。这些工具有些使用文件中转数据，有些使用管道，不同程度的为数据中转带来额外开销，效率差别也非常大。另外，有些时候，我们也希望在一个很短的时间窗口内，将一份数据从一个数据库同时导出到多个不同类型的数据库，要综合使用这些没有集成的工具来做，难度很大，几乎没有办法实现。

为了解决这些问题，淘宝开发了异构数据源数据交换工具 DataX，DataX 是一个在异构的数据库/文件系统之间高速交换数据的工具，实现了在任意的数据处理系统(RDBMS/Hdfs/Local filesystem）之间的数据交换[4]。如图 2 所示，是 DataX 示意图。

图 2 DataX 示意图

DataX 采用框架+插件的架构方式，框架处理了缓冲，流控，并发，上下文加载等高速数据交换的大部分技术问题，插件仅需实现对数据处理系统的访问。DataX 框架内部通过双缓冲队列、线程池封装等技术，集中处理了高速数据交换遇到的问题，提供简单的接口与插

件交互，插件分为 Reader 和 Writer 两类，基于框架提供的插件接口，可以十分便捷的开发出需要的插件。比如想要从 oracle 导出数据到 mysql，那么需要做的就是开发出 OracleReader 和 MysqlWriter 插件，装配到框架上即可。

DataX 有如下特点：

- 在异构的数据库/文件系统之间高速交换数据
- 采用 Framework + plugin 架构构建，Framework 处理了缓冲，流控，并发，上下文加载等高速数据交换的大部分技术问题，提供了简单的接口与插件交互，插件仅需实现对数据处理系统的访问
- 运行模式：stand-alone
- 数据传输过程在单进程内完成，全内存操作，不读写磁盘，也没有 IPC
- 开放式的框架，开发者可以在极短的时间开发一个新插件以快速支持新的数据库/文件系统。

2.上层应用的异构性

在淘宝的整个系统中，作为云梯的应用层，有很多种应用，每种应用的开发语言和系统结构都不一样，由于上层业务不同，每个业务部门都有自己的应用，云梯必须能够很容易的支持各种不同的应用，而实际上，云梯也确实很好地处理了各种不同的应用都能够很好的使用云梯这个架构平台，而且集成得非常好。为什么能够做到这样的无缝集成，是因为云梯公开了很多基本 API，而且还发布了很多相关的专门 SDK，各种应用，无论是淘宝内部的应用，还是第三方的应用，都可以很快地，很顺利的集成到云梯这个平台上来，这样，也使得云梯这个架构能够很容易的流传和发展，壮大。

数据异构性的处理是近年大数据处理方面的一个热门话题，淘宝的 DataX 可以说很好地解决了数据异构性的问题。而且 DataX 采用 Framework+plugin 的方式，并且已经开发了一些数据库的插件，很重要的一点，它还是开放的，这就使得 DataX 很容易发展，完善和成熟起来，并且得到了很多公司和开发者的大力支持，并被广泛的使用和研究。

简单说一下 DataX 的应用场景，如果只是想一次性的把一些数据从一种数据库导入到另一种数据库，使用前面提到的一些工具就可以了，不必使用 DataX 来解决，但是如果是在一个长期运行的系统上，多种数据格式需要经常频繁的交换的话，DataX 是较好的选择，不但开发维护成本比较低，而且效率也很好，通常都比现有的工具在性能和内存使用上都会有显著提高[4]，如图 3 所示。

	原工具	新工具	速度提升	内存利用提升
Mysql-Hdfs	Jdbcdump	DataX	116%	114%
Oracle-Hdfs	Jdbcdump	DataX	87%	67%
Hdfs-Oracle	Getmerge+sqlloader	DataX	103%	42%

图 3 DataX 性能比较

二 系统伸缩性

1. NameNode 伸缩性

Hadoop 集群的伸缩性本身是很好的，当现有集群的机器的计算不能满足要求时，很容易的添加一些新的机器到集群中就可以很容易地扩展集群的计算能力，从而满足新的需求。

2013 年 4 月，阿里云梯集群所在的数据中心（IDC 机房）的机位已满，无法继续扩充集群。根据当时阿里集团数据量的增长趋势，在可以预见的很短时间内，集群规模将因为机房机位不足而无法继续扩充。由于当时云梯的 Hadoop 版本还不支持单集群跨机房分布的功能，所以阿里集团的大数据业务将因为集群规模的限制而停止发展。云梯的跨机房项目就在这种背景下开始的。目标非常明确：构建一个支持跨机房的 Hadoop 集群[5]。

要构建一个跨机房的 Hadoop 集群，有非常多的技术难点。

众所周知，Hadoop HDFS 中的 NameNode 单点是阻碍 Hadoop 集群能够无限扩充的一个最大问题点。云梯在跨机房之前一直是单 NameNode 的结构，不管如何优化，其服务能力有其上限。虽然经过云梯开发团队的多轮优化，已能超过 5000 台规模（日平均 RPC 访问量达到 25 亿次），但考虑将规模扩大一倍的话，显然 无法实现。所以云梯的 Hadoop 版本要能支持多 NameNode 就非常必要。

解决方案的详细步骤

明确了需求和难点，接下来就需要有明确的实施步骤，经过开发团队、测试团队、运维团队和业务团队的多方沟通和头脑风暴，云梯跨机房项目确定了如下的技术实施步骤。

第一步，将云梯集群升级为支持 Federation 版本（基于云梯自身的版本进行开发），将现有 NameNode 作为一个 NameSpace，为“NameNode1”，该“NameNode1”的 NameSpace 下拥有云梯的全量数据，规模为 5000 台。

第二步，在同机房中搭建另一个 NameNode，为“NameNode2”。该 NameNode 下的 NameSpace 为空，刚开始不管理任何数据。同时在所有的 DataNode 上创建针对 NameNode2 的 BlockPool，用来向 NameNode2 汇报。

第三步，将 NameNode1 中的部分数据（如 50%）迁移到 NameNode2（这里的迁移包括 NameSpace 中的元数据迁移和底下 DataNode 磁盘中的 block）。这一步完成之后，云梯结构如图 4 所示。这一步是一个非常大的难点。

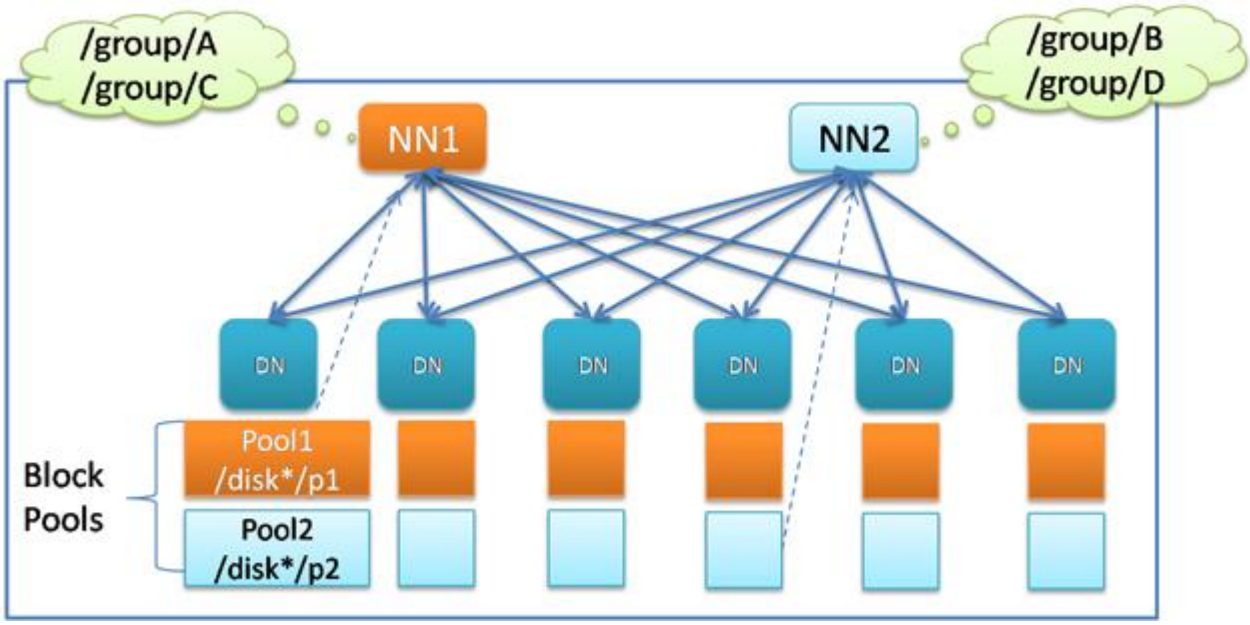

图 4 扩展 NameNode 后的示意图

可以看出，完成了这一步，基本上就解决了 NameNode 的伸缩性问题，到这一步结束，单点 NameNode 就变成了多个，原先由一个 NameNode 来承担的对文件系统所有元数据的访问被分摊到了多个 NameNode 上，NameNode 的性能、内存和扩展性问题都不复存在。

2.机房伸缩性

经历完上述三步，实际上云梯已经完成了多 NameNode 的切分，但数据仍然在一个机房里面，分别由两个 NameNode 来分别管理。此时将另一个机房（机房 B）已经准备就绪的 Slave 机器开始同时向两个 NameNode 进行汇报。也就是说，将另一个机房的 Slave 机器进行上线服务。

第四步，将 NameNode2 从原先的机房（机房 A）转移到另一个机房（机房 B）。这样，两个 NameNode 从物理机房上就已经分离，只不过 NameNode2 上管理的数据所对应的 block 块仍然分布在机房 A，需要对这部分数据（图 1 中的/group/B 和/group/D 的 blocks） 进行迁移。这里的迁移方式很特殊，云梯团队开发了一个新的 Master 节点，叫做 CrossNode，来实现数据的跨机房分布和跨机房拷贝的策略。图 5 是 CrossNode 的架构图。

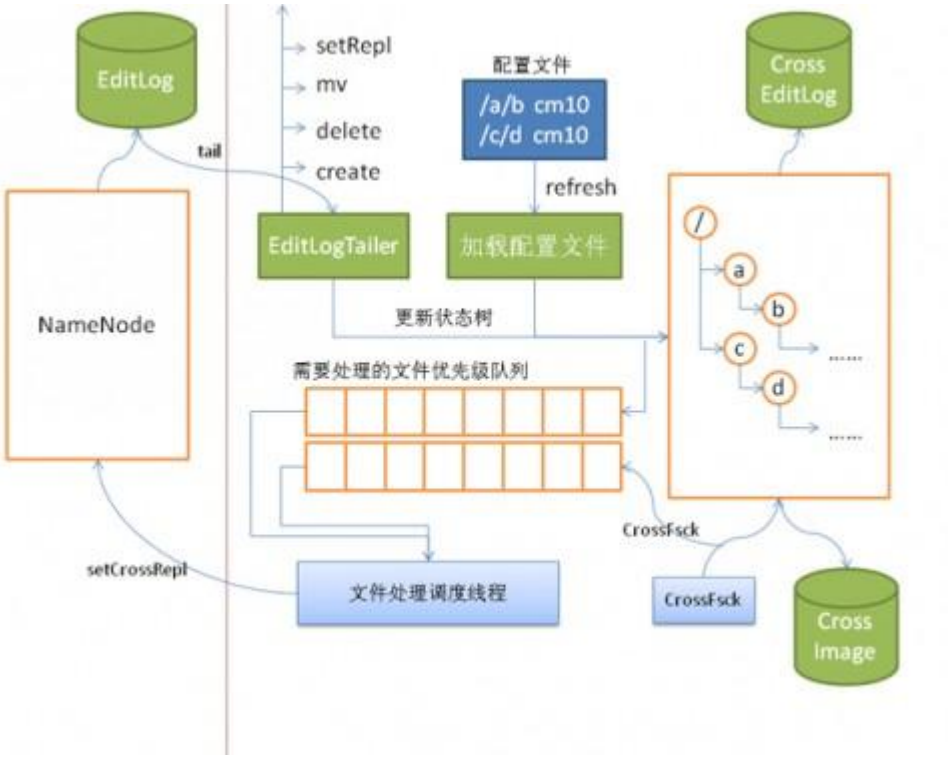

图 5 云梯 CrossNode 架构图

CrossNode 是一个全新独立的 Master 节点，它所管理的是：读取一个配置文件，这个配置文件中记录了哪些文件和目录下的数据需要跨机房放置。例如，一个文件原本有 3 个副本，都在 A 机房，把这个文件的路径写入到 CrossNode 的配置文件中，让 CrossNode 知道这个文件需要跨机房放置，并且在 A 机房是 3 个副本，B 机房需要 2 个副本，主角 CrossNode 只负责从 A 机房拷贝 2 个副本到 B 机房。同时由于在跨机房那个目录中会有新的文件被创建和写入，所以这些目 录下的文件在写入完成后，也需要后期对其进行跨机房副本放置的处理，这些也都是 CrossNode 来完成的。

因此，将 NameNode2 需 要管理的那部分数据从 A 机房迁移到 B 机房的方式为：将这些文件和目录配置到 CrossNode 的配置文件中，这样 CrossNode 就会发现有那么多的数 据需要在 A 机房和 B 机房同时放置，如 3:3，于是会对这些数据的 block 进行跨机房拷贝，直到所有的 block 全部在 B 机房拥有 3 个副本，然后将 CrossNode 中的配置进行修改，例如修改成 0:3，这样表示 A 机房需要 0 个副本，B 机房需要 3 个。于是 CrossNode 重新工作，将 A 机房原本的 3 个副本进行删除，保留 B 机房的 3 个副本，支持完成 NameNode2 管理的所有数据从 A 机房到 B 机房的迁移。大家可以发现，这个 CrossNode 的方 案，正好解决了难点 2，因为有了 CrossNode 以后，数据的迁移变成了一个计算的前序操作，每天云梯系统会根据前一天的计算分布和其他业务情况来决定 哪些文件需要跨机房分布，哪些需要进行迁移，哪些需要去除跨机房分布。这样在计算真正运行时，绝大部分计算 job 需要的数据都会存在于计算调度所在的本机 房内，并不需要跨机房读取和跨机房写入，这样机房间的带宽并不会成为影响计算作业效率的瓶颈。即使只有少量需要跨机房读写的访问存在的情况下，机房间的带 宽也完全能够处理得过来。

三 透明性

1.让底层变更对上层业务透明。

经过第四步以后，云梯已经达到多 NameNode 切分，以及数据的 NameSpace 切分以及相应的 block 多机房 分布。接下来的一个问题是：如何让底层的这些变更对上层业务完全透明？我们采取的策略是 ViewFS。简单地说，ViewFS 是云梯开发团队全新开发的一 个实现 Client 端对多 NameNode 的透明感知组件，让客户端能够自动找到正确的 NameNode 来进行数据的读写。实现了 ViewFS 的的云梯 Hadoop Client 结构如图 6 所示。

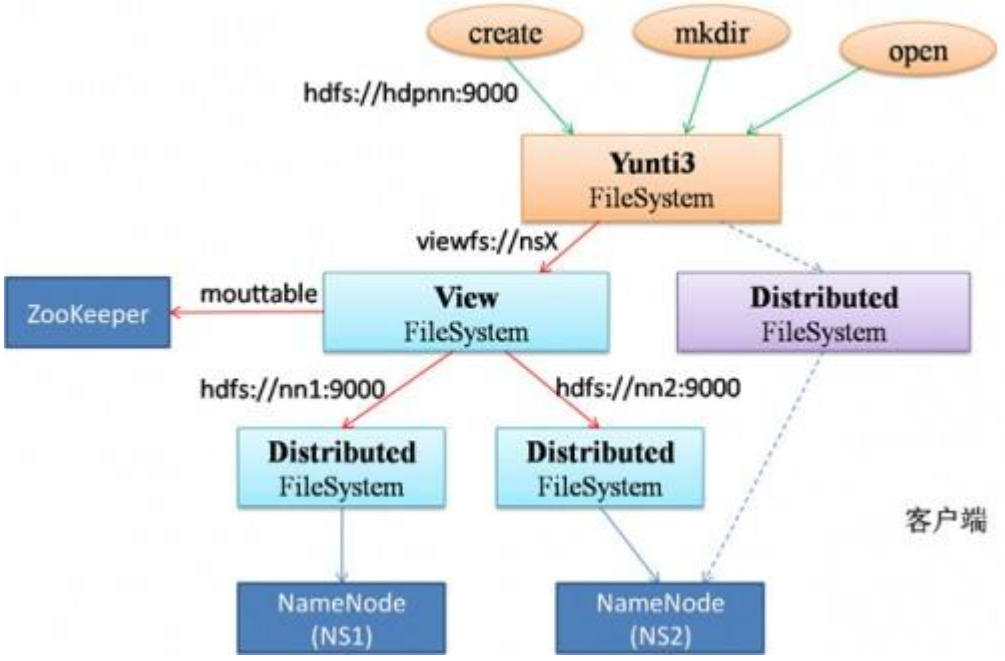

图 6 云梯 Client 架构图

经过客户端的改造和升级，以前老的访问方式由智能的 Client 接管。由 Yunti3FileSystem 根据访问路径自动选择要访问的 NameNode，实现了数据的分切对客户端透明。

2.访问透明性和位置透明性

通过这样一改造，用户和应用程序的访问也都是透明的，用户不需要关心具体的硬件在哪个机房，也不需要关心在哪一台具体的计算机上，当然，应用程序也不需要关心哪一个 NameNode 在哪一个机房，访问完全透明。

3.性能透明性

由于机房可以扩充到大于 3 个，自然服务器集群也就可以随之扩展，可以由原来 5000 台的集群规模扩充到上万台的集群规模，性能可以得到大幅提升，而在性能提升的时候，是

完全透明的，整个系统的性能提高，完全做到按需分配。

4.伸缩透明性

由于业务的提升，整个计算量也会大幅增加，系统很容易进行扩展来满足新的需求，而且扩展对整个系统来讲，系统的整体架构和算法等都不会改变。

经过上述多个步骤，云梯集群就实现了多 NameNode 切分、数据的跨机房分布和管理、计算的跨机房调度等，虽然物理上云梯集群跨越两个数据中心，但对上层业务来说，完全感知不到底层的变动。不仅如此，由于实现了多 Master 的切分，让多个 NameNode 来分担以前一个 NameNode 来管理的所有数据，所以也很大程度上释放了以前单 NameNode 节点在扩展性和性能上的瓶颈，让以前因 为单 Master 节点带来的种种问题全部都迎刃而解。同时也为将来集群的进一步扩充留下很大的余地。基本上，在基于跨机房的那个版本和框架下，云梯 Hadoop 集群几乎是没有物理上限的一个集群了。

至此，经过上述一系列步骤， 便实现了云梯 Hadoop 集群的跨机房服务。目前，云梯集群规模已接近万台，跨越两个 IDC 机房，数据的分布和计算的调度每天都在根据实际情况进行实时调 整。跨机房云梯的更重要意义在于：在未来的时间里，阿里集团运行在 Hadoop 平台上的大数据业务不需要再为数据规模和性能而担惊受怕，在云梯现有的架构 下，已经基本看不到集群规模的上限，性能也可以根据实际情况和访问情况来进行动态的调整。

通过分析研究淘宝的云梯分布式架构，了解到了目前大数据的一些发展方向和前景，可以说，淘宝的许多业务都是基于大数据，基于云梯这个平台展开的，如果没有云梯这个云计算的平台，阿里巴巴的很多业务不可能做到那么好，比如它的小额贷款业务，它的坏账率要比一般的银行低得多，而且云梯这个架构也展现了中国的计算机技术人才并不比国外的差，也展现了中国云计算在实践当中的具体应用，值得其他计算机相关公司，科研机构借鉴和学习。

淘宝云梯使用了 Hadoop 开源平台，但是又不仅仅局限于 Hadoop 本身，在 Hadoop 基础上也做了大量的改进，不但提升了淘宝本身的应用技术，也推动了 Hadoop 的发展，而且淘宝自己也对自己的一些框架进行了开源，也对整个开源世界注入的新鲜血液，也大大地推动了中国的开源价值观。

参考资料：

[1]http://www.taobao.com/about/?spm=1.6659421.774530365.15.YZHxOZ
[2]http://blog.1688.com/article/i34839292.html
[3]http://wenku.it168.com/d_000415480.shtml
[4]http://www.open-open.com/lib/view/open1325771223625.html
[5]http://www.w3c.com.cn/%e6%b7%98%e5%ae%9d%e4%ba%91%e6%a2%af%e7%9a%84%e5%a4%9anamenode%e5%92%8c%e8%b7%a8%e6%9c%ba%e6%88%bf%e4%b9%8b%e8%b7%af
[6]http://adc.alibabatech.org/carnival/history/schedule/2013/detail/main/286