基于实时计算的Web分析系统的研究与开发

# 研究背景及意义

随着互联网的迅速发展，运营商面临的竞争日益激烈，挖掘用户行为产生的数据，可帮助运营商监控网站运行状态，更好的改善网站结构，做更好的业务和营销决策。现在很多公司每天都会产生数以TB级的大数据，如何对这些数据进行挖掘，分析成了很重要的课题。大数据时代，传统的web分析系统架构难以适应海量数据挖掘的要求。实时监控网站运行情况的需求也日益剧增。例如电子商务网站，需要处理并且挖掘用户行为产生的数据，产生推荐，从而带来更多的流量和收益，而每天处理海量的用户数据，需要一个低延时高可靠的实时流式分布式计算系统。再如社交网站，大家每天都会去社交网站是为了看看现在发生了什么，周围人在做什么，流式计算可以把用户关注的热点聚合，实时反馈给用户，从而达到一个圈子的聚合效果。

本课题研究工作是基于对公司网站，实现用户行为数据的采集，数据存储，以及实时分析。

# 与本课题相关的研究现状

## Web分析发展过程

美国网站分析协会[1]WAA(Web Analytics Association)（现已改名为DAA:Digital Analytics Association）对网站分析的官方定义“Web Analytics is the measurement, collection, analysis and reporting of Internet data for the purposes of understanding and optimizing Web usage.”

网站分析工具的鼻祖是[WebTrends](http://webtrends.com/)在1993年美国波特兰成立，基于日志分析用户行为[2]。在1996年第一个网站计数器[Web-Counter](http://www.hitwebcounter.com/)诞生，之后在各大网站都开始出现访客计数器。 2005年Google收购Urchin之后推出[Google Analytics](http://www.google.com/analytics/" \t "_blank)，2008年10月Yahoo!收购IndexTools后推出[Yahoo! Web Analytics](http://web.analytics.yahoo.com/" \t "_blank)[3]，2011年Google宣布推出针对大客户的付费版Google Analytics[4] Premium。同时国内的百度公司也推出了百度统计，并在2011年将百度统计开放给任何人使用，全部免费。

## Web分析流程

最早提出Web挖掘的是Oren Etzioni [5]，在他1996年的论文中声称，Web挖掘是利用数据挖掘技术从互联网的文档和服务中来自动发现可用信息。就挖掘的内容而言，Web 挖掘可分为 Web 内容挖掘、Web 结构挖掘和 Web 使用记录挖掘[6]。Web使用记录挖掘，也就是本论文研究的Web分析，流程可分为源数据收集、数据存储、数据挖掘和模式分析3个阶段[7]。图1所示为Web分析的整个过程。

--An Approach for Frequent Access Pattern Identification in Web Usage Mining

--- Formation of Algorithms Module and Dynamic Interface in Web Mining and Storage Process to Retrieve Metadata

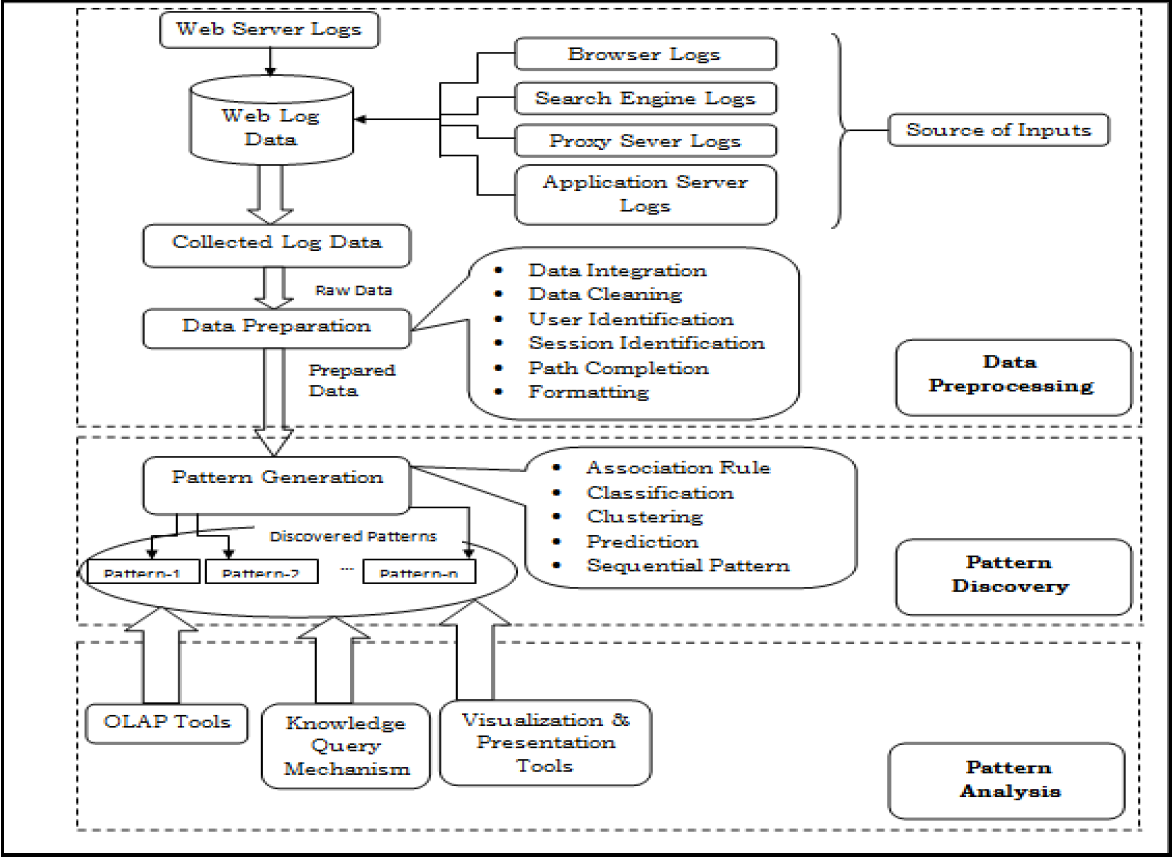


图2‑1 Web分析流程

### 数据收集

当访客和web站点进行交互时,主要用来收集数据的方法有:web日志、Javascrit标记法[7]。而有些公司的高级电子商务软件内置了数据收集机制,如收集重要的业务事件和背景数据的事件日志。有时候需要数据收集的方法不止一种,可能会多种方法同时使用。

1、Web 日志技术

web日志从web出现的时候起就一直是数据收集的最原始数据源,它最初是被开发出用来捕获Web服务器产生的错误[8]。随着分析的需求从基于技术转向基于营销,它的功能慢慢地被“强化”为捕获更多的数据。当访客客户端同浏览器建立连接后，浏览器通过 socket向网站服务器发送请求，服务器接收到请求，然后根据请求找到客户端需要的文件，然后在日志中记录下这个访客的请求信息，最后服务器将浏览器请求的文件发送给浏览器。图2-2所示为基本Web日志格式。

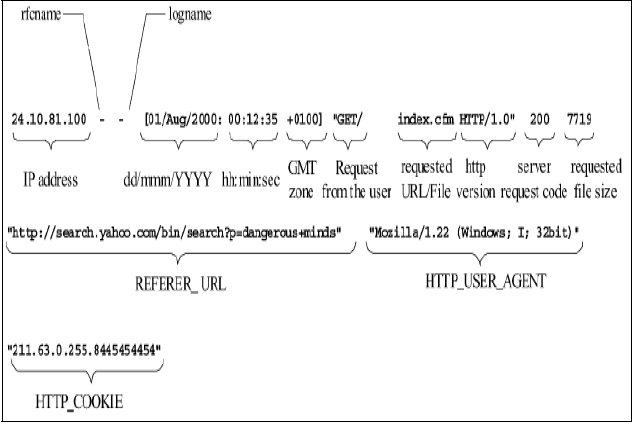


图 2‑2 Web日志基本格式

Web Usage Mining: A Review on Process, Methods

and Techniques -- 主要讲过程，综述，没有细节。

典型的网站日志文件，通常包含如下信息[9]：

远程主机的 IP 地址或名字；

访客登录名；

访客登录全名；

请求所发生的日期时间；

请求方法；

请求的文件地址；

请求所遵守的协议；

请求的状态；

被请求文档的大小。

服务器日志分析法的优点在于他分析的数据与终端无关，即无论访问网站的是手机，电脑还是其他终端，他都能够真是地记录访客信息；服务器日志分析的数据不依赖于第三方的支持，只要有访客访问服务器，日志就会被记录；服务器日志分析的数据源是网站的日志，所以他的数据源是最容易获得的；采用服务器日志分析法分析网站的结果数据是可恢复的，只要有原始日志，即使分析结果丢失，也可以重新恢复；由于服务器日志是记录服务器的响应行为，所以她还能记录搜索引擎机器人在服务器上访问行为；还有一点就是用服务器日志法分析网站不怕客户端防火墙，由于数据源是从服务器上直接获得的，与访问终端无关，所以客户端防火墙的设置不影响数据收集。然而，日志分析法也存在缺点，只有访客访问服务器才会产生日志，但是随着代理服务器缓存和浏览器缓存的诞生，本来访客可能访问服务器的情况由于缓存的原因而未发生，导致日志记录的访问量少于实际访问量。日志分析法不能精确地辨别独立访客，由于日志分析法主要是根据访客 IP 地址来区分访客的，但是这样做导致同一个局域网下的所有访客被人为是一个访客。显然这种分析方法相对来说准确度很低。

2、JavaScript标记技术

JavaScript标记[10]目前是业界最普遍和最受欢迎的技术,大部分的供应商和Web分析的解决方案都依靠JavaScript标记来收集数据。JavaScript标记法网站分析是在网页源文件中插入一小段可执行的 JavaScript程序代码当网页被下载到用户客户端的浏览器上时，这段 JavaScript代码就会被执行。随后，它就会如实的把访客在页面上的访问行为不间断的发送给这个网站分析所对应的服务器。

JavaScript标记法网站分析是最流行的一种分析技术[11]。同日志分析法相比，页面标记法不怕代理服务器缓存和浏览器缓存的影响，由于页面标记的源代码是放在网页源文件中的，即使网页被代理服务器的缓存或客户端浏览器的缓存保存下来，页面标记的源代码也会跟着一起被保存，并且当浏览器载入页面的时一起被执行；页面标记法主要依赖于Cookie 中的记录来辨识访客的身份信息，同仅依赖于访客 IP 地址辨识访客身份的日志分析法相比，页面标记法具有相对准确的访客记录；页面标记法网站分析可以只对网站的某个具体的页面进行分析，灵活性更大；它具有更好的实时性，基本上在访客访问网站的同时就完成了页面统计。

### 数据存储

数据库存储技术已经走过三十几年的发展，各种数据库系统的设计为满足不同尺度的数据规模以及复杂变化的应用需求。随着近年来电子商务网站、超大规模的门户网站及高并发性SNS纯WEB2.0动态网站的兴起，越来越多的系统对数据库提出了高并发读写、高效率存储和访问、高扩展和高可用性等需求，而传统的关系型数据库无法满足这些需求。在 WEB2.0的网站中，关系数据库大部分都出现了瓶颈。在磁盘IO、数据库可扩展上都花费了开发人员相当多的精力来优化，比如做分表分库、主从复制、异构复制等等[12]。而非关系型数据库[13]的一个共同特点都是去掉关系数据库的关系型特性。数据之间无关系，这样就非常容易扩展。也无形之间，在架构的层面上带来了可扩展的能力。接下来针对其三种常见类型，键值存储、列存储、以及文档存储。

1、键值存储

即Key-Value存储，简称KV存储。它的数据按照键值对的形式进行组织，索引和存储。每个key都是唯一的，客户端根据key来获取value值。键值数据库的出现，主要是受到了Amazon's Dynamo[14]的影响。在Dynamo中，数据被分区存到多个服务器中并且有多个备份。可扩展性和持久性依赖于两个关键机制：分区与复制、数据版本控制。

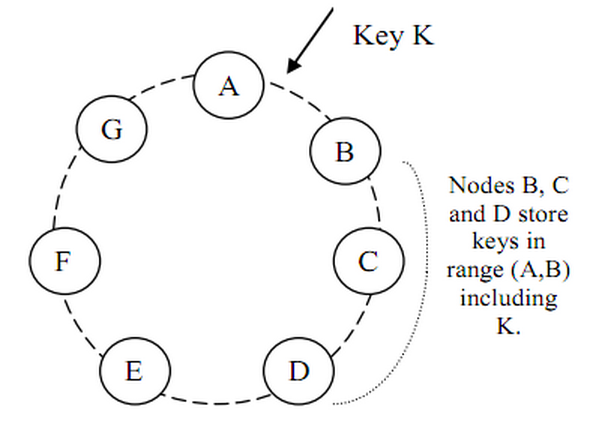


图 2‑3 **Dynamo环上的分区和复制**

* 分区与复制：

Dynamo对所有保存的数据Key做MD5哈希生成128位的特征码，将所有128位的数值首尾相接就形成了图2-3的环（Dynamo Ring），Dynamo Ring被节点（Node）切割形成了若干数据段（Key Range）。根据特征码将保存于Dynamo的数据对象分布到不同的Key-Range上，集群中的服务器（Host）各持有一个Node，负责存储当前Node与上一个Node划分出的Key-Range上的数据对象[15]。伴随Dynamo存储的数据不断膨胀，必然需要添加更多的Node来把Dynamo Ring切分成更小范围以扩张系统的性能。添加的Node会从其它相邻的节点上“偷取”一部分Key Range。这样在添加Node时受影响的就只有与之直接毗邻的那个Node了。比如我们在A、B之间添加Node A’，则只有Node 会受到影响将自己的一部分数据转移给A’。    同样的在一个Node发生故障或者移除Node的时候，受影响的也只有它直接毗邻的那个Node。

------From：http://my.oschina.net/pangyangyang/blog/151093

* 数据版本控制：

在Dynamo中，由于并行的更新和错误的发生，可以导致某个对象的版本有分枝出现。此时系统无法协调这个对象的版本，必须保同的版本，等待客户端来决定最终的版本[16]。Dynamo通过这种方式来保证尽量不丢失任何的更新。Dynamo中采用向量时钟来解决这个问题。每个对象都有一个向量列表，列表中每个元素是一个（node, counter）对，表示该对象包含了node上第counter次更新的结果。由于正常情况下，更新都是在preference list中的前N个节点上发生，所以向量列表的长度不会太长。万一出现一些异常情况，可能会导致向量列表的增长，Dynamo采用了一种简单的方式来处理这种问题，当向量列表长度超过某个定值的时候（例如10），把最早的向量（node, counter）对删除，为此，每个向量还需要带有一个时间戮，表示该更新发生的时间。删除向量可能会导致某些更新丢失，但一般来说，Dynamo在生产环境中很少遇到这样的问题。

其他的键值数据库包括：Voldemort [17],Redis [18], Tokyo Cabinet [19] , Tokyo Tyrant[20],Memcached [21] , MemcacheDB [22], Riak [23],和 Scalaris [24]. Voldemort, Riak, tokyo Cabinet and Memecached 可以存储在内存中也可以存储在磁盘上。其他的存储在内存中，并提供磁盘恢复，或者依赖复制来恢复数据。

2、列式存储

列式[数据库](http://baike.baidu.com/view/1088.htm" \t "_blank)[25]是以列相关[存储](http://baike.baidu.com/view/87682.htm" \t "_blank)架构进行数据存储的[数据库](http://baike.baidu.com/view/1088.htm)，为了实现可扩展性，行与列被分割存放到个多个节点中。

BigTable[26]是一个分布式存储系统，它可以支持扩展到很大尺寸的数据：PB级别的数据，包含几千个商业服务器。Google的许多项目都存储在BigTable中，包括WEB索引、Google Earth 和Google Finance。BigTable是一个稀疏的、分布的、永久的多维排序图。采用行键盘（row key）、列键（column key）和时间戳（timestamp）对图进行索引。图中的每个值都是未经解释的字节数组。

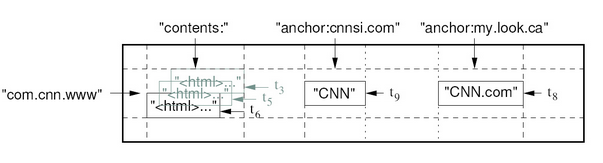


图 2‑4 BigTable存储例子

其他的列式存储数据库包括Facebook的Cassandra[27]，HBase[28]，以及Hyper-table[29]等。列式存储都借鉴自Google的Bigtable，每一行数据的各项被存储在不同的列中，而每一列中每一个数据都包含时间戳属性，这样列中的同一个数据项的多个版本都能保存下来。

--- from 书

3、文档存储

相对于键值数据库，文档数据库[30]支持更复杂的数据结构。文档数据库可以看作是键值数据库的升级版，允许在存储的值中再嵌套键值。文档数据库主要包括MongoDB,CouchDB,SimpleDB,他们都是以JSON来存储数据，最主要的区别是数据复制以及一致性机制。

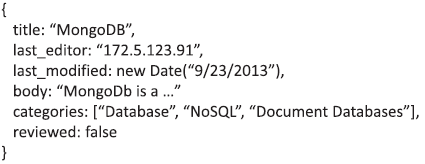


图 2‑5 MongoDB JSON数据格式

* 数据复制与切分

MongoDB的复制至少需要两个实例。其中一个是主节点，负责处理客户端请求，其余的都是从节点，负责复制主节点上的数据。主节点记录在其上的所有操作oplog，从节点定期轮询主节点获取这些操作，然后对自己的数据副本执行这些操作，从而保证从节点的数据与主节点一致。SimpleDB复制数据到不同数据中心的不同机器上以保证数据安全并提升性能。CouchDB当前没有分片机制，使用乐观的复制实现可扩展性，每个CouchDB的数据库可以同步到另一个实例;因此，可以建立任何类型的复制拓扑。

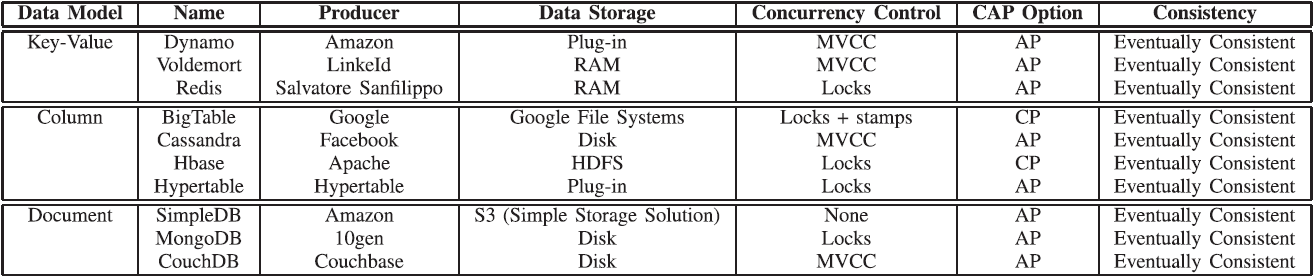
* 一致性

MongoDB和SimpleDB没有并发版本控制和事务管理机制，但他们提供最终一致性。CouchDB的一致性根据配置而不同，对于主-主，CouchDB提供最终一致性，对于主-从，CouchDB支持强一致性。

4、各种类型数据库对比

数据模型有着各自的优缺点，它们适用于不同的领域。不管是选择关系模型，还是非关系模型，都要根据实际应用的场景做出选择。也许你会发现单一的数据模型不能满足你的解决方案，许多大型应用可能需要集成多种数据模型。表2-1对三种数据存储类型做了归纳对比：

表 2‑1 三种非关系型数据对比



### 数据处理

大数据中蕴含的宝贵价值成为人们存储和处理大数据的驱动力。Mayer-Schönberger 在《大数据时代》一书中指出了大数据时代处理数据理念的三大转变，即要全体不要抽样，要效率不要绝对精确，要相关不要因果[31].因此，海量数据的处理对于当前存在的技术来说是一种极大的挑战。目前，人们对大数据的处理形式主要是对静态数据的批量处理[32]，对在线数据的流式处理[33]，以及对图数据的综合处理。

1、静态数据的批量处理

利用批量数据挖掘合适的模式,得出具体的含义,制定明智的决策,最终做出有效的应对措施实现业务目标是大数据批处理的首要任务.大数据的批量处理系统适用于先存储后计算,实时性要求不高,同时数据的准确性和全面性更为重要的场景

* MapReduce

谷歌在2004 年公开的MapReduce[34] 分布式并行计算技术，是新型分布式计算技术的代表。一个MapReduce 系统由廉价的通用服务器构成，通过添加服务器节点可线性扩展系统的总处理能力（Scale Out），在成本和可扩展性上都有巨大的优势。谷歌的MapReduce 是其内部网页索引、广告等核心系统的基础。之后出现的开源实现Apache HadoopMapReduce 是谷歌 MapReduce 的开源实现，目前已经成为目前应用最广泛的大数据计算软件平台。MapReduce 架构能够满足“先存储后处理”的离线批量计算（batch processing）需求，但也存在局限性，最大的问题是时延过大，难以适用于机器学习迭代、流处理等实时计算任务，也不适合针对大规模图数据等特定数据结构的快速运算。

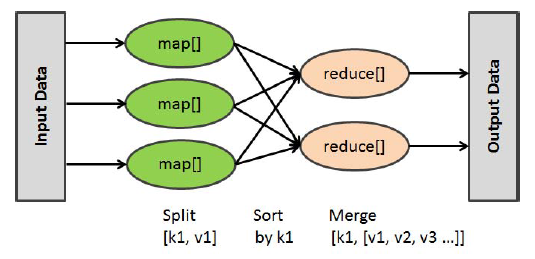


图 2‑6 MapReduce工作原理

* Dryad

2010年12月21日微软推出了dryad[35]的公测版，Dryad也通过分布式计算机网络计算海量数据，成为谷歌MapReduce分布式数据计算平台的竞争对手。与 MapReduce不同的是，Dryad是针对运行Windows HPC Server的计算机集群设计的，而且Dryad LINQ使用.NET的LINQ查询语言模型。Dryad使.Net程序员可以利用数据中心的服务器集群对数据进行并行处理，这样使得程序员在操作数千台机器时，无需关心并行处理的细节。Dryad系统的总体的构建用来支持有向无环图(Directed Acycline Graph，DAG)类型数据流的并行程序。一个Dryad 任务被表示为一个有向无环图，当发生重要的计算事件时，这些图甚至在执行期间也能发生变化。Dryad Job结构如下图1所示Dryad构建在Cluster Service和分布式文件系统之上，它的工作原理是这样的：当用户使用Dryad平台时，任务管理器(Job Manager，JM)获取图之后，便会在程序的输入通道准备完毕，并且有可用机器的时候对它进行调度。JM从命名服务器(Name Server，NS)那里获得一个可用的机器，并通过一个维护进程(daemon，D)来调度这个程序。文件，共享内存，或TCP管道都可以作为程序(节点)之间通信用的通道。在运行过程中可以动态改变图的形状，并得到很好的容错性。在调试的时候，整个图也可以运行在单个系统上。

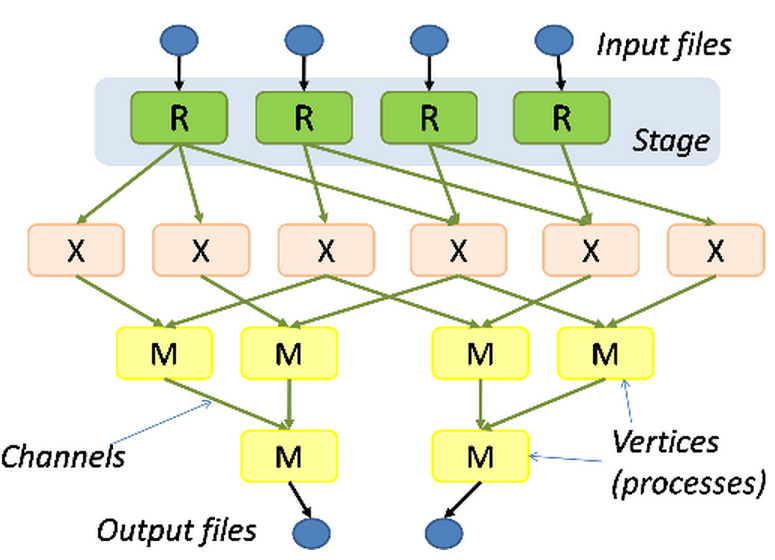


图 2‑7 Dryad job结构图

2、流式处理

流式计算中,无法确定数据的到来时刻和到来顺序,也无法将全部数据存储起来.因此,不再进行流式数据的存储,而是当流动的数据到来后在内存中直接进行数据的实时计算.如 Twitter 的Storm、Yahoo 的 S4就是典型的流式数据计算架构,数据在任务拓扑中被计算,并输出有价值的信息.

* **Storm**

Storm[36]是Twitter支持开发的一款分布式的、开源的、实时的、主从式大数据流式计算系统，其核心部分使用了高效流式计算的函数式语言Clojure 编写，极大地提高了系统性能.但为了方便用户使用，支持用户使用任意编程语言进行项目的开发. 任务拓扑(topology)是 Storm 的逻辑单元，一个实时应用的计算任务将被打包为任务拓扑后发布，任务拓扑一旦提交后将会一直运行着，除非显式地去中止.一个任务拓扑是由一系列 Spout 和 Bolt 构成的有向无环图，通过数据流(stream)实现 Spout 和 Bolt 之间的关联，如图 10所示.其中，Spout负责从外部数据源不间断地读取数据，并以 Tuple 元组的形式发送给相应的 Bolt;Bolt 负责对接收到的数据流进行计算，实现过滤、聚合、查询等具体功能，可以级联，也可以向外发送数据流。

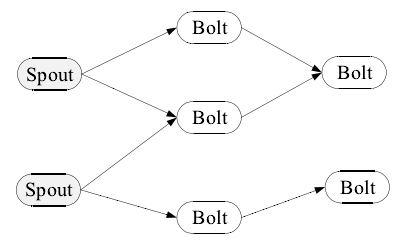


图 2‑8 Storm处理图

* **S4**

S4[37]是 Yahoo 支持开发的一款分布式的、可扩展的、可插拔的、对称的大数据流式计算系统，编程语言为 Java处理单元 PE(processing element)如图 2-9 所示，是 S4 中的基本计算单元，由4个组件构成，即:(a)函数.实现了与该处理单元 PE 相对应的功能和配置;(b)事件类型.规定了该处理单元 PE 所接收的事件类型;(c)主键.规定了该处理单元 PE 所关心的事件主键;(d)键值.规定了该处理单元 PE 所匹配的键值处理单元PE只关心与其事件类型相匹配的事件，并仅仅处理与其主键、键值相一致的事件，即，只有事件类型、主键、键值全部匹配后，处理单元 PE 才会处理该类事件.当一个新事件没有可以匹配的处理单元 PE 时，系统将会为该事件新创建一个处理单元 PE.因此，需要高效、动态地创建、管理和删除处理单元 PE;同时，处理单元 PE 的类型设计及其拓扑结构也需要更合理地规划。

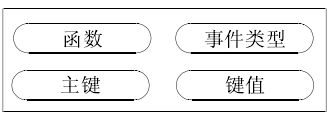


图 2‑9 S4基本计算单元

3、图数据的综合处理

图由于自身的结构特征，可以很好地表示事物之间的关系，在近几年已成为各学科研究的热点。图中点和边的强关联性，需要图数据处理系统对图数据进行一系列的操作，包括图数据的存储、图查询、最短路径查询、关键字查询、图模式挖掘以及图数据的分类、聚类等。随着图中节点和边数的增多(达到几千万甚至上亿数)，图数据处理的复杂性给图数据处理系统提出了严峻的挑战图能很好地表示各实体之间的关系，因此，在各个领域得到了广泛的应用，如社交网络(如Facebook、人人网)、微博(如 Twitter、新浪微博、腾讯微博)等新兴服务中建立了大量的在线社会网络关系，用图表示人与人之间的关系、基于图研究社区发现等问题、通过图研究信息传播与影响力最大化等问题。

* **Pregel**

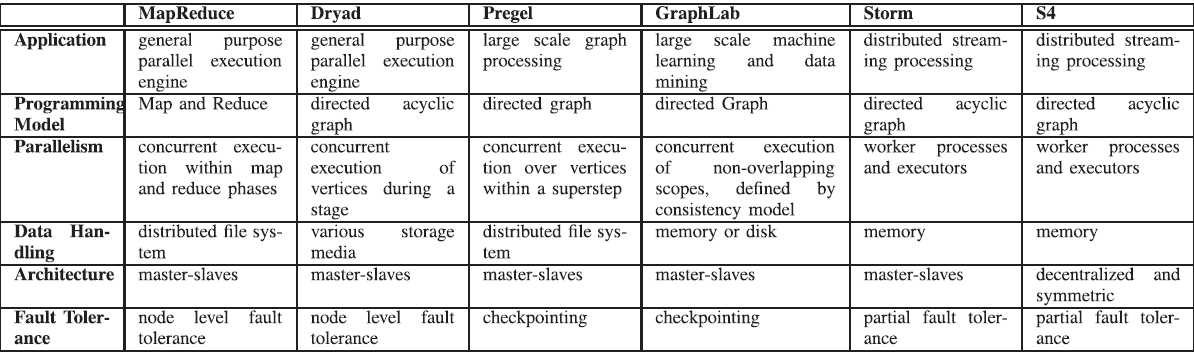
Pregel[28，29]是 Google 提出的基于 BSP(Bulk synchronous parallel)模型的分布式图计算框架，主要用于图遍历(BFS)、最短路径(SSSP)、PageRank 计算等。BSP 模型是并行计算模型中的经典模型，采用的是“计算-通信-同步”的模式。它将计算分成一系列超步(superstep)的迭代。从纵向上看，它是一个串行模式，而从横向上看，它是一个并行的模式，每两个超步之间设置一个栅栏，即整体同步点，确定所有并行的计算都完成后再启动下一轮超步。Pregel 的设计思路是以节点为中心计算，节点有两种状态:活跃和不活跃。初始时每个节点都处于活跃状态，完成计算后每个节点主动“Vote to Halt”进入不活跃状态。如果接收到信息，则激活。没有活跃节点和消息时，整个算法结束。

* **GraphLab**

GraphLab 作为一种新的机器学习算法框架，利用自然图结构(natural graph)表示稀疏的数据依赖关系；采用共享内存、异步迭代计算以及动态计算等机制，提高迭代计算的速度。作为一个广为流行的框架，GraphLab 除了提供通用的 ML 算法的编程模型，目前已实现了一系列的算法，包括聚类算法、图算法、协同过滤的近邻算法以及隐含因子模型算法。 最后，GraphLab 从代码层面对分布式算法设计中需要直面的资源竞争(raceconditions)，死锁(deadlocks)，数据流动状态(distributed state)以及集群的通信协议等方面进行了高层次的抽象，因此人们编程时能够集中精力于算法的设计上，方便地开发分布式算法。

4、各数据处理对比与总结

面对大数据,各种处理系统层出不穷,各有特色.总体来说,我们可以总结出 3 种发展趋势，(1) 数据处理引擎专用化:为了降低成本,提高能效,大数据系统需要摆脱传统的通用体系,趋向专用化架构技术。(2)数据处理平台多样化:自2008年以来克隆了Google的GFS和MapReduce 的 Apache Hadoop逐渐被互联网企业所广泛接纳，并成为大数据处理领域的事实标准。但在全面兼容 Hadoop 的基础上，Spark 通过更多的利用内存处理大幅提高系统性能。而 Scribe,Flume,Kafka,Storm,Drill,Impala,TEZ/Stinger,Presto,Spark/Shark 等的出现并不是取代 Hadoop，而是扩大了大数据技术的生态环境，促使生态环境向良性化和完整化发展。(3) 数据计算实时化:在大数据背景下，作为批量计算的补充，旨在将 PB 级数据的处理时间缩短到秒级的实时计算受到越来越多的关注。



# 第三章 研究目标及内容

## 研究目标

针对公司的网站，系统可以实时监控网站使用情况，实时查看在线人数、Top页面、热门搜索词、热门产品等。通过页面在线时长，来分析具体页面使用情况，从而能更好的确定网站功能性需求。根据分析得出的用户使用情况确定公司的短期或长期发展方向与计划，制定产品宣传的力度、营销策略、优化系统，提升客户的服务方式等。

## 研究内容

### 需求定义、确定Web分析的关键指标

需要实时分析系统在线人数，某页面浏览量，Top页面，热门搜索词，用户地域分布，用户使用环境（浏览器、操作系统、分辨率等）并根据基础指标得出网站趋势分析、来源分析、页面分析、访客分析。

### 系统总体架构

系统按照Web分析标准流程，共涉及四个模块，用户行为数据采集模块，采集数据存储模块，数据处理模块以及数据展示模块。

## 关键问题

### 大规模数据存储

随着互联网的不断扩张和云计算技术的进一步推广，海量的数据在个人、企业、研究机构等源源不断地产生。这些数据为日常生活提供了便利，信息网站可以推送用户定制的新闻，购物网站可以预先提供用户想买的物品，人们可以随时随地分享。但是如何有效、快速、可靠地存取这些日益增长的海量数据成了关键的问题。传统的存储解决方案能提供数据的可靠性和绝对的安全性，但是面对海量的数据及其各种不同的需求，传统的解决方案日益面临越来越多的问难，比如数据量的指数级增长对不断扩容的存储空间提出要求，实时分析海量的数据对存储计算能力提出要求。

### 大量数据处理的实时性

流式大数据是实时产生、实时计算,结果反馈往往也需要保证及时性。流式大数据价值的有效时间往往较短，大部分数据到来后直接在内存中进行计算并丢弃，只有少量数据才被长久保存到硬盘中。这就需要系统有足够的低延迟计算能力，可以快速地进行数据计算,在数据价值有效的时间内，体现数据的有用性。

### 大量数据处理的伸缩性

在大数据流式计算环境中，系统的数据速率具有明显的突变性，并且持续时间往往无法有效预测，这就导致在传统环境中具有很好的理论和实践效果的负载均衡策略在大数据流式计算环境中将不再适用，当前大多数开源的大数据流式计算系统在架构的设计上尚未充分地、全面地考虑整个系统的负载均衡问题,在实践应用中,相关经验的积累又相对缺乏。

# 关键技术路线

系统采用Javascript脚本嵌入目标网页中，使用html事件监听技术收集用户使用情况，如例登录系统时间、打开页面时间、浏览时长等，通过Ajax异步请求将采集的数据发送非关系型数据库中存储，并使用流式计算框架Storm实时分析采集到的数据。

## 数据存储

使用非关系型数据库Hbase存储采集的数据。Hbase是用线下所构建的实时数据库系统，它是Hadoop上的一个NoSQL实现；而NoSQL是对非关系型数据库的这一类数据库统称。Hadoop能有效和可靠地管理分布在数千个服务器上的大规模数据／海量数据，并且以记录的模式将数据存储在表中来管理与处理非结构化、半结构化、多媒体等多种数据。为此系统采用Hbase存储数据。

## 数据处理

系统使用Storm流式计算框架实现实时分析系统使用情况。Storm 是一个开源的分布式实时计算系统，可以简单、可靠地处理大量的数据流。Storm 有很多使用场景，如实时分析、在线机器学习、持续计算、分布式 RPC、ETL，等等。Storm 支持水平扩展，具有高容错性，保证每个消息都会得到处理，而且处理速度很快（在一个小集群中，每个节点每秒可以处理数以百万计的消息）。Storm 的部署和运维都很便捷，而且更为重要的是可以使用任意编程语言来开发应用。介于这些特性，本系统将使用Storm框架完成实时处理数据。

参 考 文 献

[1] 杨瑞林, 李力军. 新型低合金高强韧性耐磨钢的研究. 钢铁. 1999（7）: 41~45.

[2] Schinstock, D.E., Cuttino, J.F. Real time kinematic solutions of a non-contacting, three dimensional metrology frame[J]. Precision Engineering. 2000, 24(1): 70-76.

[3] 温诗铸. 摩擦学原理. 北京: 清华大学出版社. 1990: 296-300.

[4] 贾名字. 工程硕士论文撰写规范[硕士论文].上海: 上海交通大学. 2000.

[2] Storm Project[EB/OL]. http://storm-project.net/.

[3] Yahoo S4[EB/OL].http://incubator.apache.org/s4.