# 2、技术方案分析研究

## 2.1 重点研究开发内容和关键技术

### 2.1.1重点研究开发内容

1. 研究权衡互联网质量的相关因素，及其量化表示方法。

主要研究衡量互联网质量的相关因素及其相互之间的联系，建立相应的侦测回馈体系，为网络优化提供数据支撑。

1. 研究如何通过零散的网络数据报文，整合获取网络传输的各项参数。主要研究在不影响互联网业务运营的情况下，通过嗅探零散的连接回应报文来重现连接数据传输的情况。
2. 研究如何处理大量的数据，并从中归纳出规律。

主要研究并行数据处理架构，如何同时处理数十Gbps的业务链路数据，在形成的大数据集中实时提取出连接传输的各项要素并进行统计分析。

1. 研究虚拟群落划分和群体行为分析。

通过类似的上网行为，将实际中的上网人群映射到数字世界的虚拟人群，并进一步多虚拟人群进行挖掘分析，研究相关人群的群体行为。

### 2.1.2重点解决的关键技术问题

1. 获取全网IDC和ISP IP库，精确定位连接传输的地域信息；
2. 大规模并发数据的优化和处理；
3. 单边TCP流量数据的连接重建工作；
4. 混杂模式下的数据报文处理
5. 大数据采集结构和存储结构和分析结构。
6. 优化服务器性能，提高服务器的处理能力；

评析系统要面对庞大的数据量和复杂的处理流程，项目主要面对的风险和技术关键问题主要是服务器数据吞吐量和处理能力以及系统功能模块的伸缩性，服务器数据吞吐量和处理能力决定了系统部署的成本和运营的成本，系统功能模块的伸缩性决定了系统在不同的应用场景是否会出现木桶最短板决定容量的部署难题。

该系统有完全自主开发的高性能数据处理机制，充分的利用目前成熟的SMP服务器结构特点，压榨出服务器的每一点性能。通过修正网卡的驱动程序，将数据报文直接送入用户层，避免不必要的系统协议数据栈处理开销，同时大大的缩短数据报文的处理流程。在功能规划方面采用松耦合的模块组合方式，同时最大化的细化模块的设计，充分保证系统部署的灵活性，消除最短板决定容量的部署难题。

## 2.2主要技术创新点

**2.2.1零拷贝读写基础上的高速数据报文捕捉方法**

互联网质量评析系统是一个大数据为基础的数据分析系统，要获得全面准确的评析报告，数据集的丰富全面是一个必要条件，由此网络评析系统以旁路的方式植入需要检测的目标网络中，全面获取目标网络中所有流动的数据报文。为了最大限度的获取目标网络中的数据报文，使之形成大数据集，我们研究了一种基于零拷贝读写的高速数据报文捕捉方法。

该方法首先在服务器内存中划出连续的一片共享内存区域，并将该片区域按照SMP服务器系统中的CPU数量进行分区，使用修改后网卡驱动中直接暴露出来的数据报文读取接口和硬件DMA通道将流经网卡的数据报文直接复制到系统应用空间中的对应共享内存区域。由于这种方式充分的利用了服务器硬件的特性，使得这种捕捉的方式对CPU的占用非常的少，同时能够面对大规模的并发数据。数据报文被复制到用户空间的共享内存区域时，会依照用户配置对数据报文做了简单的连接分离，解除了各个共享内存中数据报文的依赖关系，使后续报文分析的应用也可以充分利用SMP的CPU处理特性，进一步充分利用服务器系统的系统资源。

下表为传统方式和零拷贝方式的测试结果对比：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 抓包工具 | 抓包率 | 流量 | 报文大小 | 包速率 | CPU | | | |
| 用户 | 核心 | 中断 | 空闲 |
| Libcap | 100% | 1Gbps | 500Byte | 23wpps | 10% | 6% | 5% | 30% |
| Libcap | 60% | 1Gbps | 200Byte | 55wpps | 0% | 0% | 100% | 0% |
| Libcap | 30% | 700Mbps | 60Byte | 110wpps | 0% | 0% | 100% | 0% |
| 零拷贝 | 100% | 1Gbps | 500Byte | 23wpps | 1% | 0.6% | 0.3% | 98% |
| 零拷贝 | 100% | 1Gbps | 200Byte | 55wpps | 2% | 2% | 6% | 90% |
| 零拷贝 | 100% | 700Mbps | 60Byte | 110wpps | 3% | 1% | 20% | 75% |
| 零拷贝 | 100% | 4.9Gbps | 60Byte | 1000wpps | 0% | 100% | 0% | 0% |

上表测试结果表明，使用传统的libcap进行数据报文捕捉时，最大无丢包捕获速率为23wpps，无条件最大捕获速率为110\*30%=33Wpps，而使用零拷贝的方案后，最大无丢包的捕获速率就达到了1000wpps，最终受限于测试环境的发包速率，没有测试进一步的无条件捕获上限，但由表中数据仍然可以看出，与传统的方式对比，可以提升网卡吞吐量30倍以上，并且使CPU的使用降低80%，为应用数据的分析提供了更多的CPU处理时间。

目前这项创新点的技术已经申请专利，专利名:一种基于linux的高性能数据报文捕获方案和装置”[专利申请号：201310352403.8]。

**2.2.2快速报文重组系统**

网络数据报文被复制到系统共享内存区域后，快速报文重组系统将按照网络连接（TCP协议）重组所有的数据报文，并从重组的数据报文中提取出连接传输的各项指标，重组系统使用多进程的工作方式来充分利用SMP的CPU处理特性。系统通过将报文捕捉的共享内存按照SMPCPU的个数设置成对应的段，然后为每一个CPU核心分配一个处理进程，该进程与该CPU进行绑定，并且该进程与某一个共享内存段一一对应，这样就形成了进程数据空间和代码空间在CPU上的完全隔离，减少进程在不同的CPU核心上进行调度的系统开销，充分利用好CPU内部的cache，提升进程运行的效能。这种方式与传统的依靠系统自行调度的方式有很大区别。

下表为传统方式和快速重组方式的测试结果对比：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 采集工具 | 并发连接 | 丢包率 | 报文大小 | 包速率 | CPU | | | |
| 用户 | 核心 | 中断 | 空闲 |
| systemAPI SOCK\_DGRAMsocket | 60W | 30% | 300Byte | 33wpps | 5% | 20% | 75% | 0% |
| 高速重组 | 1000W | 0% | 100Byte | 900wpps | 0% | 100% | 0% | 0% |

上述测试结果表明，在使用传统的方式进行连接重组时，最多可以重组60W连接数的数据报文，并且会造成部分数据被丢弃，而使用高速重组的方式时，系统可以承载900W连接数的重组工作，而且不会照成数据报文丢弃。

目前这项技术已经包含在专利“数据报文转发系统和方法”中[专利申请号：201310315173.8]。

### 2.2.3快速日志记录

大数据集的产生除了收集这一部分，还包括存储记录这一部分，为了应对评析系统所产生的大规模日志数据，我们研究了一种快速的日志记录系统。该系统使用网络传输日志数据，使用零拷贝读写的方式来读取网络日志数据，并在读取过程中完成数据流压缩，经过多次实验调整，计算出最大性能压缩比所需要的各项压缩参数和算法，最终实现了2Gbps文本日志数据的实时数据流压缩，存储数据的体积压缩为原来大小的20%，以方便存储。存储方面使用完全的异步IO来操作磁盘。由于零拷贝读写网络数据报文的系统可以按照配置进行简单的连接分离，所以快速日志记录系统仍然具有天生的SMP结构优越性。

快速日志记录系统的进程调度方式与快速报文重组系统一致，仍然是才用了进程隔离来充分利用SMP系统的多个CPU。在磁盘写入上快速日志系统使用了完全的异步IO方式和全缓存。在一般的操作系统上IO基本都可以分为阻塞IO，非阻塞IO，异步IO三种方式，这三种方式处理数据的效能依次递增，编程复杂度也依次递增。日志系统将异步IO的实现进行简单的封装后应用在了磁盘的数据写入上，经过测试这种方式对比阻塞IO有200%的性能提升，对比非阻塞IO有80%的性能提升。

下表为传统方式和快速重组方式的测试结果对比：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 采集工具 | 报文大小 | 流量 | 丢包率 | CPU | | | |
| 用户 | Wait | 中断 | 空闲 |
| systemAPI SOCK\_UDP（阻塞） | 200Byte | 400Mbps | 50% | 25% | 40% | 15% | 0% |
| systemAPI SOCK\_UDP（非阻塞） | 200Byte | 400Mbps | 20% | 45% | 20% | 15% | 0% |
| 快速日志 | 200Byte | 2Gbps | 0% | 45% | 5% | 0% | 20% |

上述测试结果表明，在使用传统的方式进行日志记录时，最多可以承载400Mbps的日志流量，且不同的记录方式均会带来部分的数据丢失，使用快速日志记录的方案则最大可以承载2Gbps的流量并且还能有部分CPU资源剩余，新的方式使得系统的承载量提升了5倍，并且在实际的环境中也更加的稳定。

目前这项技术已经包含在专利“数据报文转发系统和方法”中。

### 2.2.4分层存储方案

为了能最大限度的提升评析报表对大数据集的实时性和灵活性，我们设计了一种分层的存储方案，该方案通过将底层海量的数据按照上层人机接口的要求按层次进行归纳提纯，将提纯的结果按层次分别存储到关系型数据库和相应的查询缓存中，底层的海量数据则按照文本的方式进行压缩存储。

这种方式在实际应用中测试证明，极端情况下可以提升查询效率数以万倍，使得很多不方便通用实施的查询功能都可以进行通用查询，同时这种方式还能减少数据的存储容量，在实际环境中可以减少五到十倍的存储空间占用。

下表为传统存储方式和分层方式在大批量数据处理中的一些对比：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 实时统计 | 存储规模 | 实时细节查询 | 实时统计查询 | 系统负载 |
| 传统方式 | 不支持 | 大 | 支持 | 不支持 | 高 |
| 分层方式 | 支持 | 传统的5-10倍 | 支持 | 支持 | 低 |

目前这项技术已经包含在专利“数据报文转发系统和方法”中。

## 2.3项目特点

网络评析系统有七大特点：

其一，零拷贝网卡数据报文读写的方式，解决了网卡数据吞吐的瓶颈，可在单台普通四核服务器上的10GE链路的全速数据接收捕获。

其二，自行研究使用的高速日志系统，解决了评析系统日志数据大剂量生产的问题，零拷贝读写和大数据流压缩使得单台普通四核服务器上2Gbps网络质量日志信息的实时数据流压缩存储，并使压缩比达到20%。

其三，分层存储的方式，解决了海量日志数据的存储和查询问题。

其四，快速报文重组，可以快速的将零散的网络数据报文重组成数据连接，并提取连接传输的各项参数，形成评析数据源，重组峰值可达到1000万连接数每秒，报文速率可达900万pps。

其五，完全模块化设计，隔断了系统错误的传播，减低开发和维护的难度，而且模块间可以通过网络进行松耦合，所以整个系统可以按照不同的应用方式进行模块组合，使得系统的应用环境进一步的拓宽。

其六，可提供数据平台的功能，使得用户可以定制自己需要的评析数据。可以针对客户评析数据量大小，要求的结果精度等各方面做到灵活伸缩，有效的控制运营成本而不影响业务正常使用。

其七，系统把请求的处理与数据的分析进行分离，有效提高请求的吞吐量和反应速度，同时增加了系统的稳定性。

## 2.4具体研究开发内容

### 2.4.1零拷贝网卡读写功能实现；

考虑到评析系统在多个地方需要面对大规模的网络数据报文，经过测试发现，传统的操作系统在数据报文协议栈处理上消耗了大量的CPU时间，并且单通道网卡中断的SMP系统中，这些消耗的CPU无法均衡到各个CPU上，从而使得系统的总体吞吐量有限，为了提升系统吞吐量，我们采用一种绕过系统内核处理流程的报文处理机制，通过修改网卡驱动，在驱动中加入报文直接读写的接口，该接口可以在操作系统内核读取到数据报文之前提供一个报文读写的机会给应用程序，并且通过标准使该接口满足PF\_RING函数库的调用规则，由此用户的应用程序可以直接使用PF\_RING的标准库函数进行编程，既可以避免应用开发的复杂度，又能提供足够的报文吞吐性能。

### 2.4.2数据报文捕获功能实现

完成零拷贝网卡驱动修正后，便可以直接使用PF-RING的库函数。考虑到系统运营时将面对庞大的旁路数据，报文捕获模块不仅要面对大数据流的冲击，还需要提供良好的接口给报文分析重组模块。由此我们设计报文捕捉模块使用PF-RING零拷贝操作的模块来绕开操作系统内核的报文协议栈，将网卡的数据报文直接映射到报文捕捉空间，由此解决旁路数据冲击问题。在数据报文分析接口方面，采用共享内存的IPC通讯机制，并且建立简单的负载分发方式，通过配置式，让上层应用可以是根据自己的运算需求和SMP系统的相关参数来定制报文捕获模块的接口，由此充分的挖掘系统资源的使用率，简化应用构建和运营的难度。

### 2.4.3快速报文重组功能实现

互联网的IP数据报文是零散无连接的，为了在无连接的互联网上建立便捷的连接通道，多数应用都使用了UDP/TCP以及其它类似的协议来进一步封装通讯数据报文，为了完成评析系统的原始数据源的建立，我们需要实现一种UDP/TCP的协议重建功能，通过提取数据报文中的IP和UDP/TCP协议头部，依照sock的传输特性，建立相关的连接传输信息档案，然后在建立相应的状态变迁机制和计时器机制，提取出连接的相关传输特征，主要包括：传输时长，传输数据总量，传输的最高速率，最低速率，平均速率，报文往返延时（rtt），源目标IP，源目标端口，源目标地理地址，传输应用，以及额外的应用信息。

### 2.4.4快速日志信息存储

同样为了应对系统中爆炸的数据量，自行开发使用一种日志数据存储模块，依托网卡零拷贝的读写模块和相应的数据报文捕获模块，使用插件的机制来完成日志的存储和初步处理，目前已经完成的插件主要有：异步磁盘写入插件，文本信息流压缩插件，日记数据行分析插件，这些插件使用一个共同的调度核心和报文捕获模块对接，一起协作完成日志数据的分析和压缩存入。

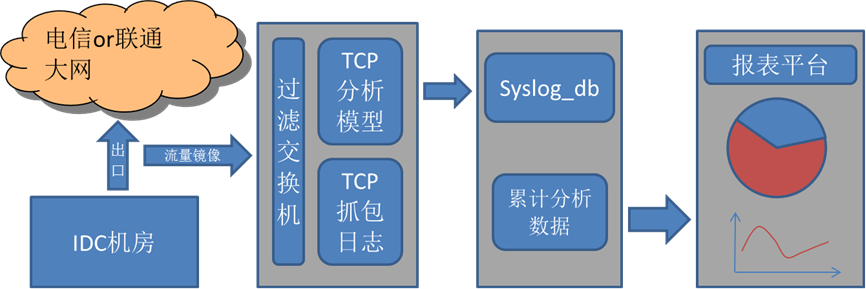
## 2.5项目总体技术方案

### 2.5.1平台架构图如下

构架拓扑



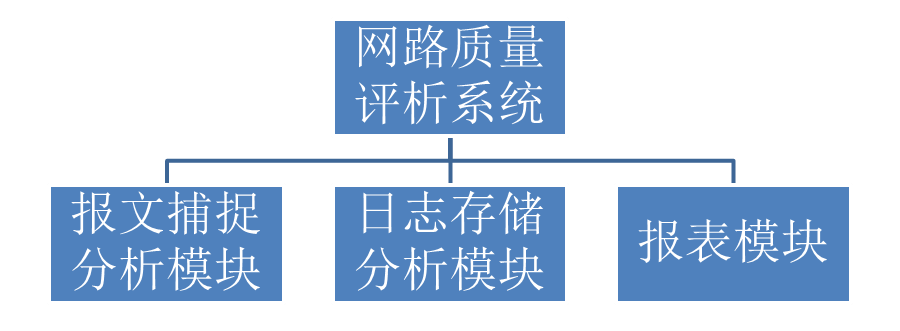
数据流



单元描述

|  |  |
| --- | --- |
| 单元名称 | 单元描述 |
| 镜像流量接收服务器 | 评析系统的数据源部分，负责接受镜像的数据流量，经过重组分析后提供机房出口连接中的各项传输信息。 |
| 日志存储服务器 | 负责接受流量镜像系统中的连接日志数据，进行初步的分析和压缩后将数据存入磁盘。 |
| 报表生成服务器 | 该模块主要为评析系统提供人机接口功能，通过提取日志存储系统中的相关数据，生成可视化，人性化的图标，通过web的方式展示给用户。 |

### 2.5.2系统组成



整个系统由三个核心部分组成，最主要的部分是报文捕捉分析模块，该模块提供了评析系统实现的核心功能。其中日志存储分析模块主要用于存储分析网络链路质量数据，运用自主设计的高吞吐数据处理方式，实时处理海量数据，精确统计网络数据信息，挖掘用户行为，详尽的数据报表，为用户提供运营决策依据。Web管理端主要为用户提供系统设置、报表查看等功能。

## 2.6项目特色及实现方法

### 2.6.1评析数据存储

评析数据存储采用网卡零拷贝读写和异步IO的方式实现。

评析系统需要面对海量的质量评析原始数据，很多的原始数据必须被整合成一个集合进行进一步的提纯运算，由此日志数据间具有一定的特殊依赖性，结合日志数据高频率生成的特点，一些常规的日志存储方案和云存储方案都具有局限性，系统中采用的网卡零拷贝读写和异步IO的方式，使得系统可面对大规模的日志数据冲击，这种实现方式首先在网卡硬件层拦截了日志数据报文，将报文从网卡直接调度到操作系统上层的用户空间，并按照一定的规则将日志数据报文放入不同的SHM贡献内存空间，这一过程的逻辑极为清晰简洁，CPU在处理数据时可以直接调用底层的DMA通道等来降低负载提升效率，从而可以进一步的保证系统的数据吞吐量。

磁盘写入方式，首先使用多进程的工作方式，为系统每一个CPU配置一个SHM共享内存区域，然后为每个SHM共享区域配置一个存储进程，在实际的部署环境中还可以为每个进程分配一个写入的磁盘来进一步提升存储性能，每个存储进程均使用异步IO的方式来操作磁盘。这样的磁盘写入方式，充分的保障了写入的效率，同时为日志数据分析插件提供简单的天然高性能运行环境。

### 2.6.2评析数据分析

评析数据的分析是一个分层次的分析过程。

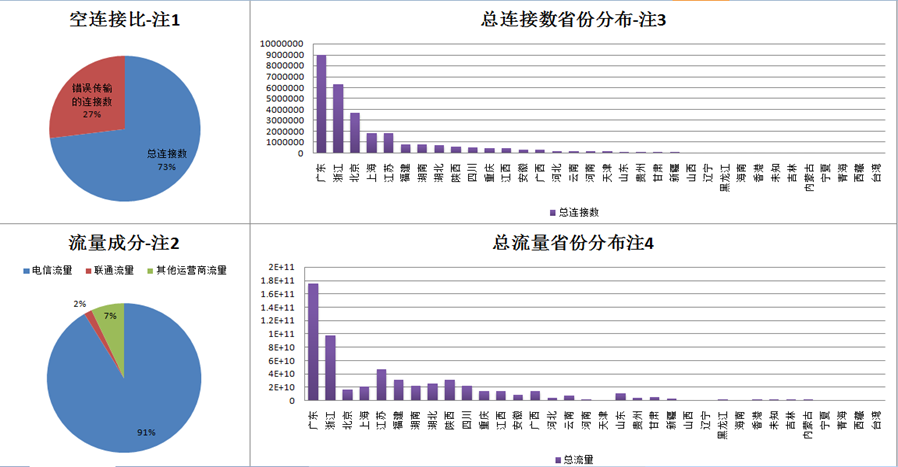
海量的评析数据，首先通过将分析插件插入在存储进程中，并将分析过程置于写入数据之前，由此完成第一层次的粗粒度数据分析，这一层次的分析是简单的，离散的统计，将每一个日志数据按照评析系统人机接口的指标分类，并进行计数。在一定的时间范围类这个统计型的指标将会被汇总存入到关系型数据库中，同时为每一个原始数据的存储文件打下相应的时间戳，以方便引用。

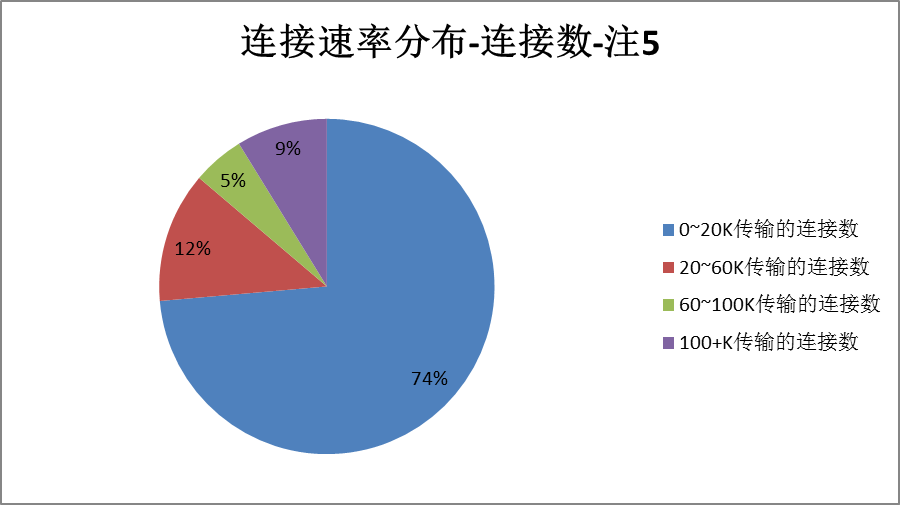
第一层的数据归纳是完全实时同步的，在完成这一层后，就可以归纳人机接口所需要的数据集合了，第一层的归纳既保证了数据集合的简洁小巧，也基本保证了数据的完整齐备，同时为某些特殊的接口建立了时间戳为索引的最小原始文件引用。

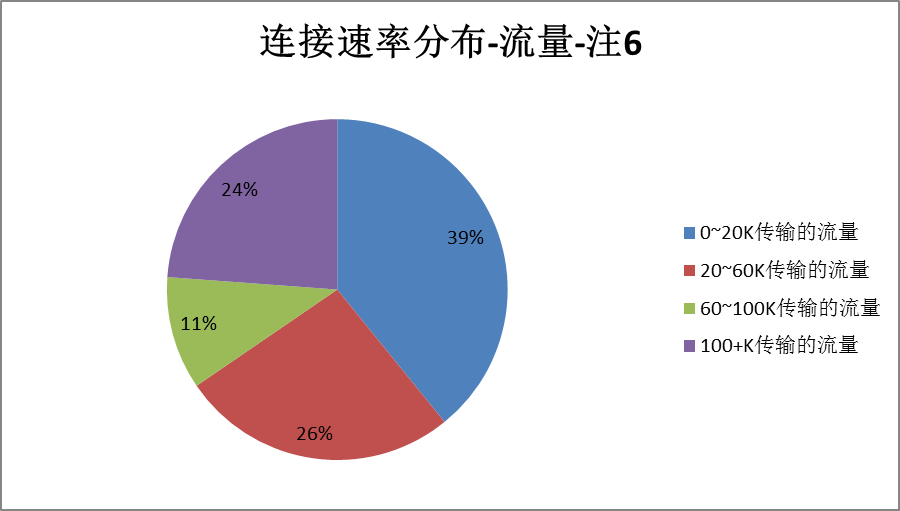
分层数据存储的使用可以使得人机接口的查询在时间上有了保证；也使得单位磁盘空间的存储量大大的增加（初步处理后的日志数据是可以压缩的）；同时引入了关系型数据库存储中间结果，大大的减轻人机接口的编程难度。

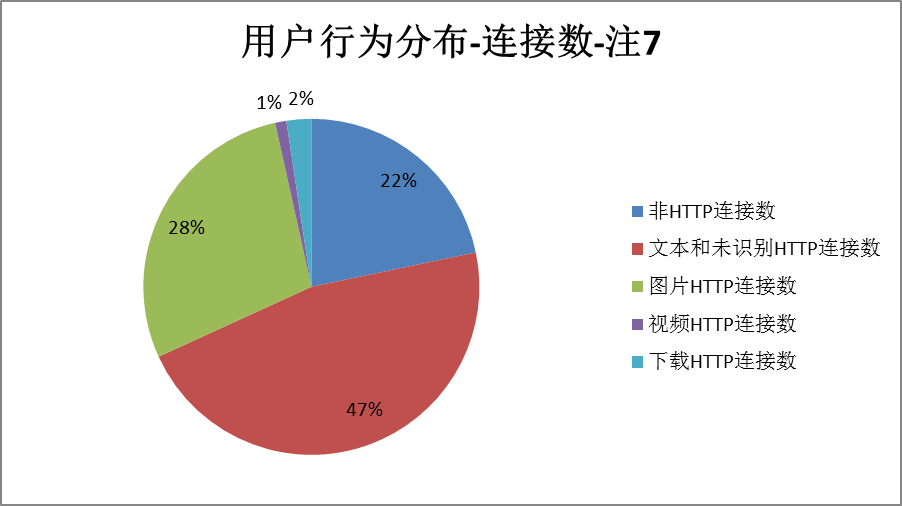
评析数据分析分为实时分析与非实时数据分析两类。实时数据分析，是在日志数据写入磁盘时，即在统计完第一层数据后进行的一种数据归纳，主要评析网络链路的实时变化情况，同时可以在这个点加入一些应用检测的告警插件，此时的数据颗粒是以秒为单位的。而非实时的数据挖掘，是指一段大时间内，对所有的运营数据进行系统的分析统计，通常以小时为单位，得到一些统计的结果。

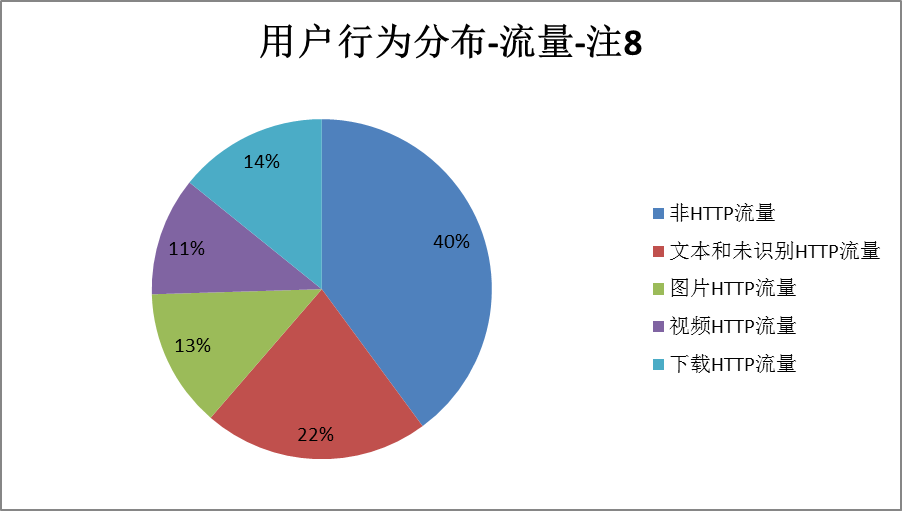
下图为评析对象的全国流量和连接数的分布：

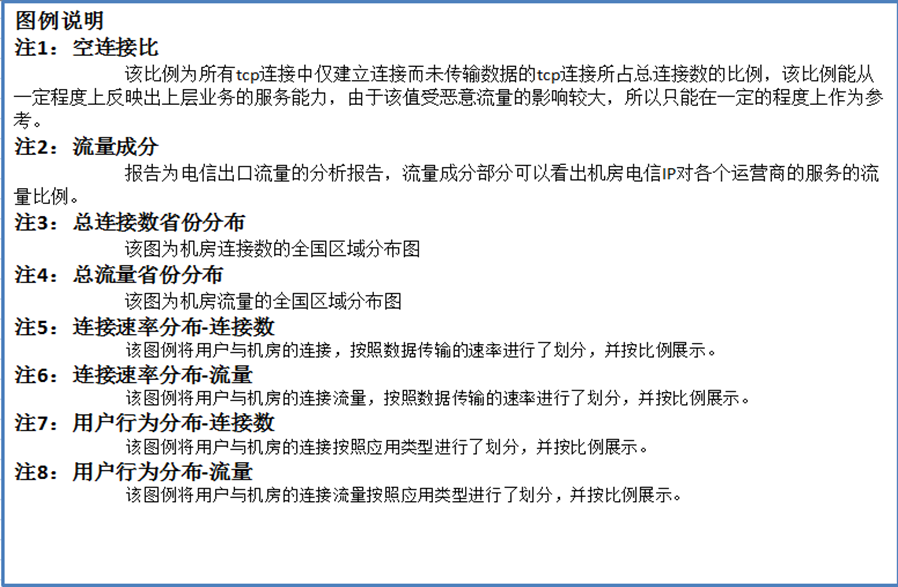




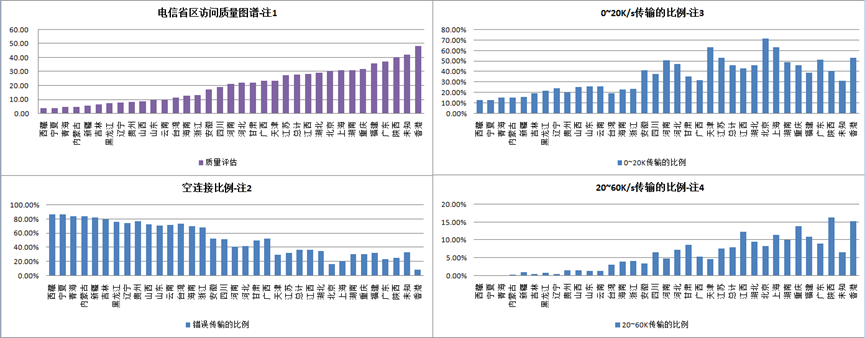


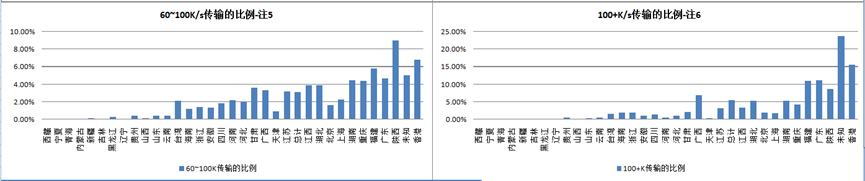


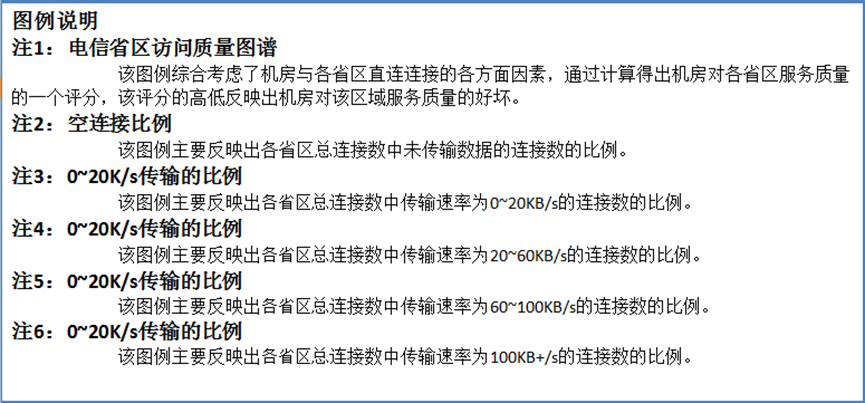




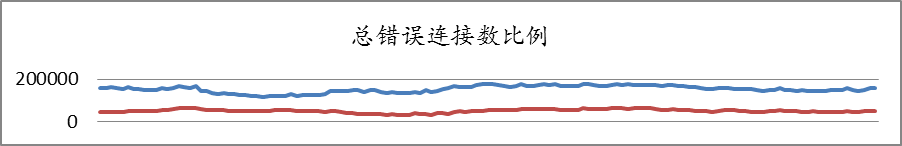
下图为评析对象与全国各地的连接速率统计：

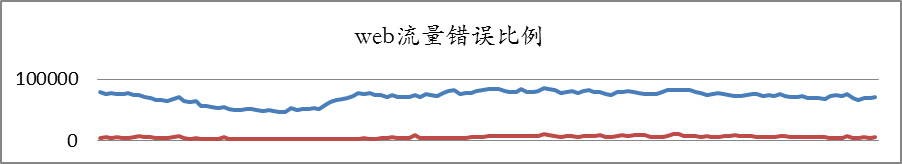


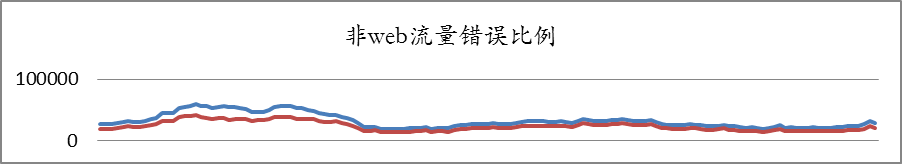




下图和cacti结合后生成的实时质量检测：

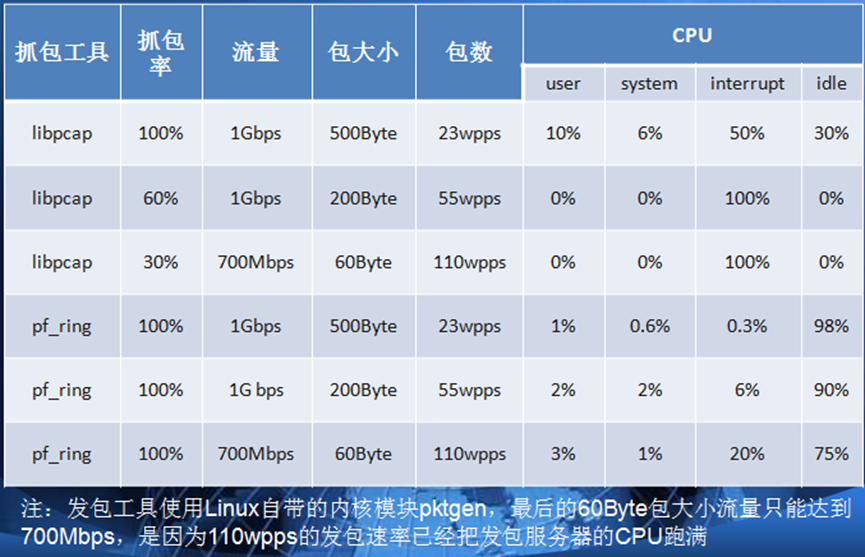






### 2.6.3网卡的零拷贝读写

这个是一个纯底层的技术，为了突破传统操作系统的数据吞吐瓶颈而设计，通过修改网卡驱动和系统中断方式，将数据报文的提取接口直接暴露到操作系统的用户层空间，仅仅利用操作系统的进程调度，其他的协议栈和数据报文分析完全自主可控开发。在零拷贝读写和操作系统默认网络读写间的相关测试数据如下：



## 2.7技术关键点及难点攻关

### 2.7.1技术关键点及亮点

|  |  |
| --- | --- |
| 功能点 | 实现方式 |
| 高速率大容量网络数据的捕捉 | 系统采用直接硬件读写的方式来捕获流经网络数据接口的数据报文，这种方式有着极好的数据处理效率，结合单臂流量的重组功能，基本可以在服务器上处理10GE接口链路上的数据流。 |
| 快速报文重组 | 使用最短协议栈的处理规则，支持是用单臂流量来预测整个连接的交互过程，单服务器数据处理上可以与零拷贝网卡读写的速率相匹配。 |
| 系统架构 | 松耦合的模块设计，细致的模块功能划分，系统支撑各种灵活的部署方式，如分析较多的场合可以扩展分析部分的模块部署，而不需要对整体的部署进行扩容，真正做到最小化投资。 |
| 服务路线 | 为用户提供完备的底层连接数据源，依照用户不同的需求，如是否进行质量告警，使用那些告警参数。是否监测应用层的数据流，是否对数据流中的关键字进行监测，是否需要对单用户连接进行追踪等，可以完全定制人机部分的报表。 |
| 分层日志数据存储 | 分层的日志数据存储为系统整体的人机接口提供了强力的支撑，从实时到非实时，从细颗粒到粗颗粒，所有的数据都能快速定位。 |
| Web管理 | 系统Web管理端，用户可以在该Web管理端进行系统各项参数的设置，可以实现对系统的全面控制，包括默认的报表等同时提供插件功能市场，用户可以在市场中找到自己需要的独特的报表或者监测插件。 |

### 2.7.2技术难点攻关

|  |  |
| --- | --- |
| 功能点 | 实现方式 |
| 高速的网络数据报文捕获和处理 | 很多的时候评析系统的报文处理前端要单独面对整个评析对象的全部数据流量（在一些不方便做负载的场合，或者小预算场景），为了使得评析系统能够尽量适应各种环境，需要开发一套完整的超高速的报文处理系统，这种处理速率有时甚至需要和纯硬件设计的交换机和路由器进行速率匹配。 |
| 高频率大剂量数据存储和分析 | 评析系统的报文数据处理部分会产生大量日志数据，这样的大数据需要采用合理有效的方式进行存储和管理。同时，对这样的海量数据进行采集分析，也需要运用时下领先的数据挖掘技术，数据仓库技术，精确挖掘用户行为，详尽的数据报表，为网络运营商提供运营决策依据。 |

**2.8评析系统性能报告**

**2.8.1前提环境**

互联网质量评析系统,主要目的有三个方面。

其一，帮助大数据流量的客户做整体的流量评析，为客户的整体业务布局提供强力参考。

其二，为运营商评析网络互联互通中的薄弱点，帮助修正网络互联互通的问题。

其三，为一些高安全要求的应用提供应用数据监测功能，使客户可以实时的监测到应用中的数据流动，第一时间发现和定位非法数据在应用中的位置。

因此出于对以上服务的需求，对我们系统的设计和运维是一大考验。所以项目使用一线的服务器厂家为强大的服务后盾，同时使用定制的操作系统和完全自主开发的性能处理部件。性能和功能上大大满足客户应用的需求。

**2.8.2网络环境**

对于评析系统而言，数据源至关重要，合理的引入评析数据是整个评析报告完整，准确的关键。我们评析系统的数据源引入，使用分光器和10GE光模块。直接上联到各大运营商的机房核心设备上，通过分光器，将核心设备上的数据报文直接镜像到评析系统，保证了评析数据源的量，使得评析系统的报表真实，贴近整体的实际情况。

**2.8.3配置信息**

使用睿江网络质量评析系统基本配置，下图为基本配置清单：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 厂家 | 型号 | CPU | 内存 | 硬盘 | 操作系统 |
| Dell | R610 | Intel XeonE5620\*2 | 24G | 80G(SAS) | CentOS 6.2  64bit |

其中，日志存储的服务器，使用服务器标准存储服务器配置，详细信息如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 厂家 | 型号 | CPU | 内存 | 硬盘 | 操作系统 |
| Dell | R710 | Intel Xeon 5620\*2 | 24G | 2T\*6(SATA) | CentOS 6.2  64bit |

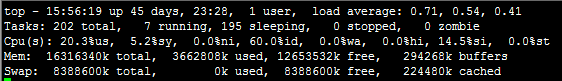
**2.8.4并发**

经过对单台服务器的优化配置，测试单一服务器网络数据报文处理能达到110Wpps，下图是某次的测试数据采样：

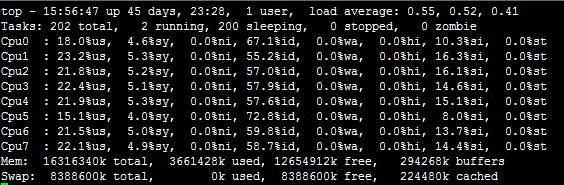


**2.8.5服务器负荷**

经过测试，数据报文捕捉处理西戎，单服务器消耗CPU性能在40%（即使：60%id空域），可以到达110Wpps的并发处理能力。系统总体资源消耗如下图：



经过对系统参数的优化处理，cpu的使用平均负载到各个CPU下，使得软硬件能力得到充分发挥。如下图：



## 2.9项目执行过程中各阶段任务

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 开始日期 | 结束日期 | 主要工作内容 |
| 1 | 2012/6/1 | 2012/9/30 | 进行产品应用的市场调研；  制定产品的总体设计方案；  进行技术准备工作，搭建开发环境等。 |
| 2 | 2012/10/1 | 2012/12/31 | 购买开发、调试、测试产品所需的硬件、搭建网络环境等；  进行新产品代码编写与调试；  对新产品的初步代码进行性能测试；  进行新产品应用部署市场应用的前期准备。 |
| 3 | 2013/1/1 | 2013/3/31 | 根据新产品的实际测试结果进行模型设计及算法设计方面的改进；  将完善后的代码进行产品包装，完成自动安装部署引导程序的编写； |
| 4 | 2013/4/1 | 2013/6/30 | 进入试运营阶段；编写产品用户手册，进行人员培训；  在公司业务中部署该产品，首先为公司的客户提供该服务。 |
| 4 | 2013/7/1 | 2013/8/31 | 根据试运营阶段客户的反馈意见完善系统，增加设计该新型产品的规格种类，扩大其应用的范围及领域；拓展产品应用的国内市场。 |