# 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. 研究如何通过对网络层网络报文传输技术的优化，克服现有技术瓶颈，在现有技术基础上为互联网用户提供跟以往相比响应速度更快、延迟损耗更低的DNS解析网络服务。
2. 研究如何搭建更合理的DNS业务模型部署架构，以解决传统DNS服务器单点集成报文组装和解析应答功能造成的硬件资源浪费、业务性能低下、负载均衡实现困难，故障与网络攻击应对能力差等问题。
3. 研究如何通过对大量用户业务日志数据的数据挖掘，从中归纳出不依赖于用户主观能动性录入的有价值的大数据分析结果，并让这些数据在业务中创造更大的用户价值。
4. 研究如何更好地解决运营DNS互联网解析服务时面临的安全性问题，如针对DNS服务器的流量泛洪攻击、域名解析数据库篡改、缓存报文污染、数据库故障和数据丢失、业务服务器故障

等。

### 2.1.2重点解决的关键技术问题

1. 获取精确的各大ISP解析IP地址库，以提供精确的运营商线路智能匹配
2. 大规模瞬时并发DNS请求报文的接收、处理和快速回复；
3. 多套DNS解析设备架构之间的业务数据实时同步
4. 业务流量层和业务逻辑数据层的剥离和分布式部署
5. 用户行为信息大数据的存储、分析和业务价值转化
6. 服务器在业务层和网络层对于攻击和故障的相应防护和保障措施

智能DNS解析服务系统的正常业务运行需要面对瞬时并发量巨大的DNS数据请求报文和繁杂的业务逻辑处理流程。该项目面临的风险和技术关键问题主要是服务器业务数据的吞吐量、数据处理能力、DNS应答数据报文延时、故障灾备应对和所提供相关DNS业务的用户需求覆盖情况。服务器数据吞吐量和处理能力决定了系统的设备部署架构、业务承载量和硬件资源需求；DNS应答数据报文的时延性问题决定了系统作为一个DNS解析服务器最直观的用户体验评价标准；故障灾备的应对能力将直接决定项目业务正常运行的稳定性和可靠性；而DNS相关业务的用户需求覆盖情况很大程度上决定了该项目在市场上的核心竞争力。

该系统通过自主研发的高性能网络数据报处理机制，充分的利用目前成熟的SMP服务器结构特点，通过修正网卡的驱动程序，避免不必要的系统协议数据栈处理开销，从根本上解决了服务器在网络数据吞吐量和报文处理性能存在的瓶颈。同时，通过DNS报文缓存技术、数据应答和报文组装架构分离等技术，大大提升DNS请求数据的应答速度。另一方面，通过数据库隔离、服务器热备、网络链路智能检测、业务模块实时检测等技术，大大保障了项目系统的健壮性，面对网络攻击和突发故障显得更加游刃有余；在业务功能的规划上，充分吸收当前市场现有产品的成熟经验和技术，并在此基础上自主创新，从用户角度出发，全力满足用户对于DNS解析服务的各方面功能诉求。

## 2.2主要技术创新点

#### 2.2.1零拷贝读写方式实现高速DNS数据报文应答

DNS系统是一个被设计成高并发、高可靠性的数据处理系统。高效捕捉业务请求数据是一个基础的、必要的条件，由于传统协议栈难以满足要求，因此DNS系统在不影响原有协议栈工作的情况下，以系统模块的方式植入系统中，全面获取服务器中所有流动的数据报文，并过滤出需要的业务数据。为此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。

传统的 I/O 操作是一种缓冲 I/O，I/O 过程中产生的数据传输通常需要在缓冲区中进行多次的拷贝操作。一般来说，在传输数据的时候，用户应用程序需要分配一块大小合适的缓冲区用来存放需要传输的数据。应用程序从文件中读取一块数据，然后把这块数据通过网络发送到接收端去。用户应用程序只是需要调用两个系统调用 read() 和 write() 就可以完成这个数据传输操作，应用程序并不知晓在这个数据传输的过程中操作系统所做的数据拷贝操作。对于操作系统来说，基于数据排序或者校验等各方面因素的考虑，操作系统内核会在处理数据传输的过程中进行多次拷贝操作。在某些情况下，这些数据拷贝操作会极大地降低数据传输的性能。

对于高速网络来说，零拷贝技术是非常重要的。这是因为高速网络的网络链接能力与 CPU 的处理能力接近，甚至会超过 CPU 的处理能力。如果是这样的话，那么 CPU 就有可能需要花费几乎所有的时间去拷贝要传输的数据，而没有能力再去做别的事情，这就产生了性能瓶颈，限制了通讯速率，从而降低了网络链接的能力。传统的数据拷贝受限于传统的操作系统或者通信协议，这就限制了数据传输性能。零拷贝技术通过减少数据拷贝次数，简化协议处理的层次，在应用程序和网络之间提供更快的数据传输方法，从而可以有效地降低通信延迟，提高网络吞吐率。首先将网卡驱动的缓冲区按照系统中的CPU数量进行分段；然后在服务器内存中划出连续的一片共享内存区域，并将该片区域按照系统中的CPU数量进行分区，利用内存映射机制，将网卡驱动的缓冲区与该共享内存区域进行一一对应关联。使用修改后的网卡驱动中直接暴露出来的数据报文读写接口和硬件DMA通道将流经网卡的数据报文直接复制到系统用户空间中的对应共享内存区域。这种方式应用程序可以直接访问到原始的数据报文，避免了数据报文在内核态和用户态之间迂回。

下表为传统方式和零拷贝方式的测试结果对比：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 文件大小 | 报文大小 | 传输耗时（ms） | CPU | | | |
| 用户 | 核心 | 中断 | 空闲 |
| 传统方式 | 20MB | 100Byte | 337 | 0% | 0% | 100% | 0% |
| 传统方式 | 360MB | 200Byte | 3631 | 0% | 0% | 100% | 0% |
| 传统方式 | 1GB | 500Byte | 18399 | 10% | 6% | 5% | 30% |
| 零拷贝 | 20MB | 100Byte | 128 | 0% | 100% | 0% | 0% |
| 零拷贝 | 360MB | 200Byte | 1762 | 2% | 2% | 2% | 90% |
| 零拷贝 | 1GB | 500Byte | 8537 | 1% | 0.6% | 0.3% | 98% |

如表所见，与传统方法相比，零拷贝方式大约减少了 65% 的时间。这就极有可能提高了需要在 I/O 通道间大量拷贝数据的应用程序的性能。

#### 2.2.2基于业务大数据的故障后备服务和数据异常回滚

考虑到DNS系统在网络服务中的重要性，以及用户对于DNS业务解析稳定性和可靠性的要求，为了能最大化提高系统业务运行的健壮性，使系统在极端情况下（如数据库失效、模块内部通信失败等）也能提供可靠的服务，我们通过采用Hadoop、MongoDB、Hbase等大数据运算和存储技术，设计出一套基于DNS系统运行日志数据挖掘的DNS数据报文组装模块，作为报文组装核心模块的后备服务。该系统通过收集DNS系统的运行日志，并写入分析系统的分布式文件系统中，定时进行大数据分析提取出组包系统所需的各类基本信息，如域名、记录类型、记录数据等，并以有序、高效的方式写入后备系统的数据集群。为DNS系统随时提供故障后备服务。

同时，该系统还提供一套完备的外部数据调用接口，能根据需要提取出域名在某个时段的实时数据，如提取出某指定域名在1小时前的A记录地址信息。这样大大便了故障排查，以及降低了系统的二次开发的难度。

#### 2.2.3 请求应答和报文组装模块架构分离

在传统的DNS服务解析系统中，一般都会把记录解析参数配置、数据报文组装、数据报文应答三大主要功能部分单点集成在同一个服务器上。当采用这种业务架构形式，服务器每次业务应答都要进行数据报文的重复组装，不但造成重复无谓的系统性能开销，同时还可能出现因为网络层报文收发造成的性能开销影响到应用层业务逻辑的正常运行效率。由于业务层与网络层单点紧密聚合的缘故，使得服务系统对于网络流量攻击的防护能力薄弱，数据库不能得到充分地与外部隔离而很有可能遭到外部的攻击与数据篡改，存在很大的安全隐患；另一方面，这种架构形式对硬件要求更高、消耗更大，不利于进行大规模集群扩展和热备部署。

鉴于上述原因，我们研究出了一种有别于传统单点式部署的分布式部署架构DNS业务系统。在该DNS业务系统架构中，请求应答、报文组装和数据库存储将分别部署在三部分各自互联的服务器集群上协同工作。采用该分布式架构后，实现了网络层数据报文收发和应用层DNS业务逻辑的角色剥离，使得请求应答服务器的角色功能更简单专注、快捷高效；同时，核心报文组装和数据存储模块可以不再对外部公开，作为内部通信服务，得到有效的外部隔离和安全性保证。所有外部的异常流量和入侵行为都由行使应答服务的缓存服务器来承担，使得安全策略和热备部署更加简单。同时，该架构支持多台缓存服务器围绕单台报文组装和核心数据库的快速扩张部署，大大降低了大规模集群扩展和后期集中管理难度。

#### 2.2.4主动智能预同步的DNS报文缓存技术

为了使用户的DNS请求获得更快的响应回答，一般DNS解析系统都会搭建缓存服务器对DNS报文进行本地缓存。对于传统的DNS缓存服务器来说，一般是采用代理的方式，即收到用户的DNS请求后，将会查阅本地缓存中是否存在该请求的相关应答报文，如果存在则直接答应；但如果不存在的话，则需要通过转发的方式到下一级的DNS解析服务器来获取相应的解析结果，待得到结果返回后才能再回复到用户，大大增加了数据业务路径的长度和应答延时。同时，该模式需要靠DNS报文的TTL值来保证数据的实时刷新，即使得缓存报文在一定时间后失效，需要重复消耗额外的网络开销来不断重复上述操作。

为了优化上述问题，本项目在缓存服务器上采用了主动智能预先同步的设计方案。在缓存服务器初始化或数据发生变更的时候，通过增加智能Manager中间模块进行事件的监听和数据的实时同步调度，让缓存服务器在用户的请求到来之前已经主动请求并缓存好相应的DNS数据应答报文，使得用户DNS请求响应速度能得到最大程度的优化，而不需要再被动进行反复的迭代请求过程。同时，由于智能Manager模块的存在，使得缓存服务器能时刻保证缓存数据报文的实时性，不再需要依赖于传统的TTL机制，即能做到一次缓存，永久生效。

#### 2.2.5无人值守的DNS故障智能自发现及自处理技术

对于DNS解析系统来说，系统的稳定性和可用性是最需要关注的焦点问题。为此，我们通过把DNS业务需求与NQA技术进行修改融合与创新，利用Track技术把网络监控事件和策略路由配置能够相互响应，使得可以通过在网络层路由器中部署相应监控模块，实现智能流量监测与流量自动切换功能。

通过部署应用该技术，能够在无人值守的情况下，自动实时监控对外服务的网络链路是否正常运行、服务器是否正常提供业务层服务；当其中任何一条链路或服务器发生不可用的情况，能够在无人操作的情况下第一时间自动识别问题链路或服务器，并把本来指向该链路或服务器的DNS网络业务流量智能自动选择并切换到可用的网络链路中去，最大程度保证系统对于用户的稳定性和可用性，而且一切操作均基于程序智能自动运行，大大节省人力资源配置。

## 2.3具体研究开发内容

#### 2.3.1修改网卡驱动

把网卡的收发队列按SMP系统中CPU数目划分好，在用户空间中同样申请SMP系统中CPU数量的内存段，然后通过内存机制做好关联。修改网卡驱动，把到达网卡收发队列的原始数据报文根据一定的负载均匀策略分发到特定的内存段中，这样就达到绕过系统协议栈的目的，一步到位地到达用户空间，这时上层应用可以直接针对原始数据报文做处理。

#### 2.3.2零拷贝高速收发接口实现

应用程序得到了原始数据报文后要做一系列的解包、打包工作，为了降低使用难度，需要开发一套完整的收发、以及集成路由功能的接口。

对于接收数据报文流程，首先按照tcp/ip协议从原始数据报文中取出各种基本元素（协议类型、源地址、目标地址等信息），然后经过过滤得出业务数据报文，最后把数据报文提交到上层业务层做进一步处理。

对于发送数据报文流程，首先按照tcp/ip协议把上层应用提交的实现负载数据加上有效的头部信息，然后根据目标地址，通过路由模块匹配出最合适的路由策略，最后提交到网卡发送队列。

#### 2.3.3快速报文透传模块实现

网络数据报文被复制到进程用户空间后，数据报文捕捉模块从数据报文中提取出数据报类型、目标地址、源地址等信息，过滤出业务数据报文并提交上层处理。其中，对于DNS业务数据报文，直接通过零拷贝读写接口提供业务层直接处理，减少由于网络包拷贝而造成的性能开销；但对于非业务数据报文，必须找到一个办法使得数据报文能重新回到系统网络协议栈，否则将会让除DNS业务外的所有网络功能受到影响。

鉴于以上情况，项目中需要通过研发对应透传模块把非业务数据报文透传到系统协议栈，以保证非业务数据能与服务器正常交互。快速报文透传模块使用多线程的工作方式来充分利用SMP的CPU处理特性。系统先分配一个与系统协议栈通讯的线程，并分配好缓冲区，按照SMPCPU的个数设置成对应的段，然后为每一个CPU核心分配一个处理线程，该线程与该CPU进行绑定，并且该线程与某一个共享内存段一一对应。当非业务数据报文到达时，系统把报文拷贝到对应的缓冲区，再通知系统协议栈通讯的线程取出数据，通过修改后的驱动零拷贝写接口提交到系统协议栈。这样就形成了线程数据空间在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，使非业务数据正常通讯。

#### 2.3.4基于业务大数据的故障后备服务和数据异常回滚实现

利用零拷贝高速收发接口实现一个收集日志的模块（该模块的能抵抗并发量是1000万qps），高效地把日志颁发到日志存储系统，把收集到的日志上传到hadoop hdfs；然后每5分钟调用日志分析程序分析系统运行日志，提取出各个请求的相关信息（域名、请求类型，域名对应的返回数据等），经过归并后，存储在mongodb集群中。

接着实现后备组包模块，实现一套完整的DNS协议解析接口，并开放出网络端口给缓存代理服务器使用。当接收到缓存代理服务器的请求后，系统会根据所请求的域名信息查找mongodb集群，匹配出最近一次该域名类型的记录，再组织好一个DNS回复包返回给缓存代理服务器。并实现一套查询接口，该接口可实现根据域名、记录类型和时间段查找出记录数据的功能，为数据回流提供支撑。

#### 2.3.5DNS报文组装模块实现

通过把出口线路（电信、移动、联通及自定义线路）归集起来结成“线路出口表”，为每条线路定义全局唯一标识——线路ID。每个用户的DNS记录信息都能任意从现有的线路出口表中选择一个出口。同时，在配置业务数据时建立好“子域名”与“主域名”的对应关系，生成域名信息表。把用户的DNS记录按线路划分好，用“主域名”和“线路ID”标识一个用户在某一线路上的DNS记录，形成“域名区域表”命名为ZONE，给每一个ZONE定义全局唯一标识——ZONEID。

按照DNS请求类型把用户配置的记录分成以下类别：A、AAAA、CNAME、MX、NS、TXT，每种类别用一个数据表存储。表里的每一行数据都与“域名区域表”关联起来。这样只要通过“主域名”和“线路ID”就能找到该ZONE下的所有配置数据。

通过建立一个独立的线程监听通用的DNS请求端口53，建立线程池处理DNS请求。监听线程接收到“高速代理服务器”发送的DNS请求后通知线程池的一个线程进行处理。

处理线程根据标准的DNS协议（RFC1035）解释好请求报文，得出子域名、请求类型（A、AAAA等）和线路ID等信息。根据子域名从“域名信息表”中检索出对应的主域名；根据主域名和线路ID从“域名区域表”检索出ZONEID；根据ZONEID和子域名从对应的请求类型表里检索出对应的配置数据；最后根据标准的DNS协议组织好回复报文返回出调用者。

#### 2.3.6DNS报文高速缓存实现

为了保证系统对于用户外部DNS请求响应速度的及时性，需要部署DNS报文缓存节点服务器来实现DNS请求的高速应答。首先，我们通过在缓存服务器应用零拷贝数据报文收发技术，最大程度提升缓存服务的网络吞吐量。缓存服务器以DNS数据报文的类型来进行基本列表区分，并以请求域名为主键构建哈希列表，并在哈希节点中按照智能视图做好基本分类。当缓存节点每次启动或者受到数据变更的通知，缓存节点都会根据新增的信息构建DNS请求包去模拟用户请求来发往核心报文组装服务器，并把核心模块应答的报文整个预缓存到内存。当收到用户请求的时候，缓存服务器对请求包进行拆解，并根据数据包中的关键数据在内存中进行哈希索引。一旦发现符合的数据报文，便直接提取并拷贝到数据报文中直接回复，而不用再重复进行数据报文的组装，系统性能花销和算法时间复杂度极低，保证了业务的高效性。对于不存在相关记录的请求，即视为业务无关的攻击流量包，进行屏蔽丢弃以保证不影响正常业务的有效运行。

#### 2.3.7DNS报文请求应答和报文组装模块架构分离实现

与普通的DNS服务器不同，本智能DNS解析服务系统作为DNS记录解析的根服务商，必须时刻保证解析业务的高效性和解析结果数据的实时性和准确性。

为了保证解析业务的高效性和安全性，我们研发出了一种报文请求应答和报文组装模块分离的部署架构。基于此架构下，业务流量请求的应答工作全部由缓存节点来承担和完成，缓存服务器并不关心用户配置数据的增删和准确性，即并不关注业务逻辑本身，而更多地作为一个单纯网络数据收发的主要流量接口；而报文组装模块本身并不具有很高的网络高并发可用性，它更多的是关注于业务逻辑的本身，把系统资源都主要花费在报文的组装和用户数据的实时读取上。实施这种角色分离的好处是显而易见，使得各个角色模块的职能更加单一高效，通过协同运转来使得整个系统的性能更加高效。由于本项目系统提供的是支持智能线路匹配的DNS解析服务，而不单单是传统的简单DNS解析，请求应答和报文组装角色的分离使得我们必须提供一个数据渠道来让两者能够通信智能线路匹配的相关信息。因此，我们在传统DNS标准协议（RFC1035）的基础上，实现了一种向下兼容的私有协议，使得两者之间的数据通信能够在一次简单的DNS通信中完成而不需要额外开销，保证系统运作效率。

系统运行过程中，如何协调报文组装和缓存应答服务器之间的数据交互同步是一个很重要的问题，这将直接影响到系统的可用性。因此，我们通过引入一个智能调度管理的中间层角色（以下简称Manager），来负责协调缓存应答与报文组装模块之间的数据交互同步。Manager将会不断监听用户数据、智能视图IP库等关键参数，当这些参数发生变化时，Manager将会作为通知者把更新信息推送到缓存服务器，让缓存服务器主动进行报文更新同步，以保证用户配置生效的实时性。

#### 2.3.8在现有DNS通信协议（RFC1035）上实现支持智能线路匹配的私有通信协议

在业务过程中，缓存服务器需要不断与报文组装核心模块进行通信，以保证缓存服务器所缓存的数据报文与核心数据库中的数据实时同步。对于缓存服务器与报文组装核心之间的通信数据，不仅要包含传统RFC1035所规定的协议信息，同时因为存在智能线路匹配的需求，需要在传输的DNS数据报文负载中加入IP库智能匹配的相关信息，以使得报文组装能够根据智能线路匹配的情况组装并返回相对应的解析结果。因此，我们在现有DNS通信协议（RFC1035）的基础上，实现了一套包含智能解析相关信息的私有通信协议，且该协议能够向下兼容传统的基于RFC1035的DNS通信协议。

应用这套私有通信协议后，不仅使得基于此私有协议设计的内部模块之间的能够通过DNS协议传递智能视图匹配信息，同时亦能使各功能模块向下兼容传统的DNS协议报文，直接响应所有标准DNS协议的请求，方便服务器内部测试，同时使得项目中的缓存服务器和核心报文组装能够和任何基于标准DNS协议的服务设备随时进行直接对接和替换，大大增加了项目的灵活性和后期可扩展性。

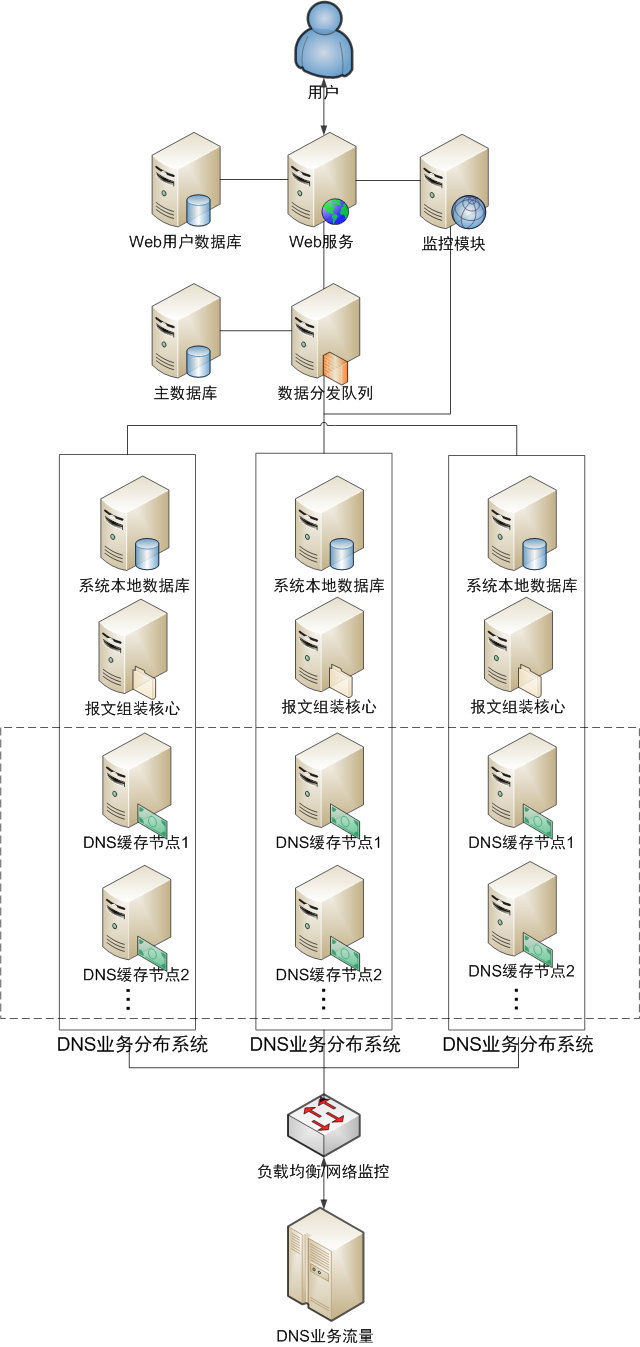
#### 2.3.9DNS业务智能监控实现

由于DNS解析系统经常作为网络攻击的首要攻击对象，因此，对于运行智能DNS解析业务的本系统平台来说，系统的安全性、可用性和故障抗击性是优先要考虑的问题。

为此，我们通过运用NQA网络质量监控技术，并创新性地把它与DNS业务流程机制进行有机结合。通过在路由网络设备不断对缓存服务器集群节点进行有效性检测轮询，以在最小时间粒度的情况下判断服务器和网络链路的可用性。一旦发现DNS业务发生故障，通过Track联动功能以通知到应用模块来改变当前策略路由，把指向问题链路的网络流量自动切换到可用的服务器链路节点上。整个监控过程基于智能事件调度，不需要人工干预，最大化保证业务系统运行的稳定性和可用性。

## 2.4项目总体技术方案

### 2.4.1平台架构图



单元描述

|  |  |
| --- | --- |
| 单元名称 | 单元描述 |
| 负载均衡/网络监控节点 | 根据项目安全性、稳定性扩展性的需求，系统架构采用分布式架构，该节点为业务流量的统一出入口，对于外部请求流量，通过在该节点配置网络路由策略，把流量分担到后方多台处理服务器上面，极大提高系统的并发承载能力；同时通过部署自主研发的故障自识别和自切换智能监控模块，一旦发现后方服务器或者链路出现问题，则可第一时间自动进行路由策略变更，保证业务正常运行 |
| DNS报文组装核心 | DNS解析系统功能的核心部分，通过读取系统本地数据库中的用户配置信息，实现全局DNS解析报文的组装业务逻辑，智能解析判断与实现。同时在后台进行日志统计与分析，当组装功能出现异常时，能启动基于业务大数据的故障后备解析服务和数据异常回滚 |
| DNS缓存节点 | 该模块主要是提供域名解析结果的高速缓存应答功能，该模块直接与外网DNS业务流量进行通讯，具有很高的数据并发能力。同时该模块承担对外部网络流量的攻击流量防护的职责，不同的缓存节点之间能互为热备。同时对于业务日志进行后台记录和暂存，集合后交由Hadoop进行用户数据挖掘以支持Web的用户数据报表等功能。 |
| Web用户数据库 | 数据库主要保存用户账号信息，用户系统账号配置设置参数，添加域名记录，监控地址列表及报警配置等数据。 |
| 主数据库 | 该数据库中存放所有用户在系统平台上配置的相关DNS解析记录和域名记录，作为系统全局数据库的最高优先级别存储。 |
| 数据分发队列 | 通过消息队列的方式把主数据的变更操作分量同步到各分布式业务系统的本地数据库中，以保证各分布式系统的数据一致性和实时性 |
| 系统本地数据库 | 该数据库部署在每一套分布式DNS解析系统中，为每一套分布式系统中DNS业务实际读取的实际数据来源，其数据均是来自于主数据库，为其一份子副本，且不断保证与主数据库中数据的同步性和实时性，该本地数据库的存在使得就算分布式系统与主数据库发生网络连接中断，业务也能继续正常运行，大大加强了DNS业务系统的稳定性和可用性。 |
| Web管理 | 系统Web管理端，用户可以在该Web管理端进行域名管理和DNS解析记录设置，实现对DNS智能解析匹配结果的全面控制，具体包括用户视图的智能划分、域名的智能解析、域名解析访问情况的监控，域名解析量情况统计分析报表等。 |
| 监控模块 | 该模块分为两部分，一部分为面对用户使用的业务层监控，用户可以通过在用户web系统中配置监控模块需要处理的监控任务，对已存在的解析记录解析结果的实际访问情况进行实时监控，并报警提醒；另一部分为业务系统内部使用的系统监控，该监控不断监视着整个系统中各个功能角色是否在正常运行，当发生故障时能第一时间发现并自动进行报警和实施配置策略。 |

### 2.4.2系统组成

整个系统由三个核心部分组成，最主要的部分是DNS核心服务模块，该模块提供了DNS智能解析实现的核心功能；业务监控模块主要用于检测保证系统的正常运行，运用自行研发设计的多种高可用性监控模块和网络链路规则设置，实时掌握系统当前运行情况，精确统计网络数据信息，最大可能维护系统的可用性和稳定性。Web管理端主要为用户提供DNS域名管理、解析记录设置、报表查看、实时监控等可配置功能。

## 2.5项目特色及实现方法

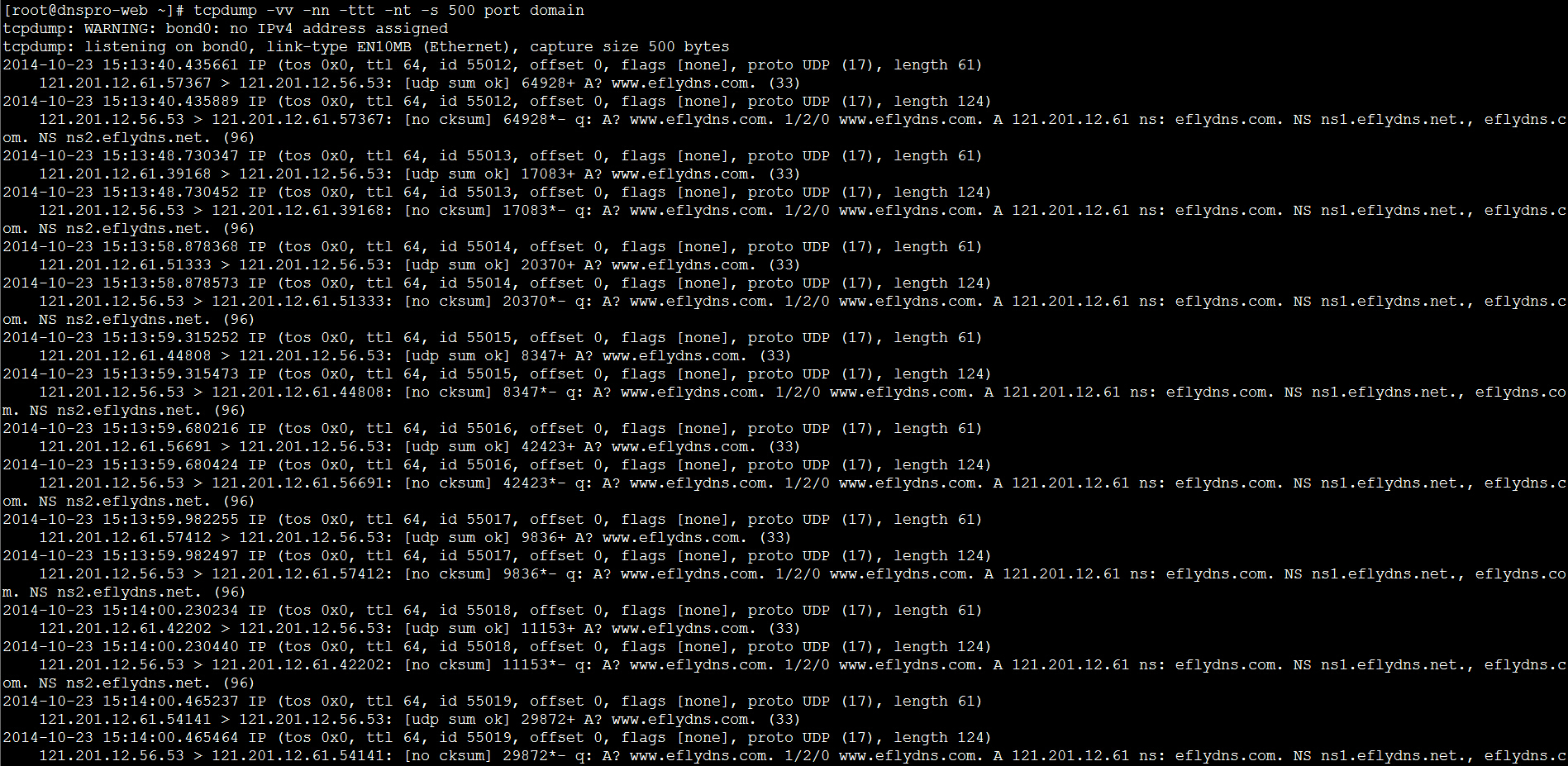
### 2.5.1高速DNS报文应答

对于DNS智能解析服务系统来说，最首当其冲，同时也是最容易反映DNS服务质量的就是用用户使用该服务系统时，DNS解析请求报文的响应速度。为了使本系统的DNS报文响应速度达到行业领先水平，使系统面向用户的用户体验达到最佳，系统采用了多方面的技术手段来进行业务提升。

首先，系统通过网络包零拷贝技术，在系统底层进行基于NETMAP的网卡驱动优化，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，大大提升了系统网络数据的收发速度，突破了传统操作系统的数据吞吐瓶颈，使得系统对于数据报文的收发响应速度、最大承载量远远高于传统的系统协议栈，在技术基础上为DNS报文的高速应答实现做了很好的铺垫与提升，大幅降低响应延时。

其次，在业务架构上采用分布式业务架构，通过把报文组装和报文应答进行角色分离，并采用搭建本地高速缓存应答节点服务器的方式，在应用层进行业务功能优化，使得用户每次请求的数据都能通过缓存节点直接返回，而不需要进行繁琐的迭代请求刷新和等待数据报文组装，大幅缩短报文应答响应时间。

通过采用系统标准的tcpdump抓包和编写测试软件进行压力测试，我们可以清晰地看到系统对于每个DNS请求的响应答复时间都能保持限制在0.2毫秒之内，达到业界先进水平。



### 2.5.2智能DNS解析服务

有别于传统的DNS域名解析商，本系统所提供的DNS解析服务能够基于智能线路进行匹配。即用户能够对旗下的域名进行智能解析记录设置，使得系统能够智能匹配DNS请求的来源线路状况，来返回用户所设置的相应ISP线路的对应解析结果，从而使得用户的域名访问能够实际指向更合理、更快速的对应同线路段的服务器，大大提升该域名的用户访问体验。

研发项目团队经过多年在相应行业的技术业务积累和数据挖掘，整理出一个全面权威的ISP线路地址库，DNS业务的智能匹配功能正是通过此地址库为数据支撑依托来实现业务逻辑。通过在缓存服务器节点中导入此地址库，对DNS请求数据包进行拆包并提取关键数据，然后利用地址库进行智能匹配，根据匹配结果返回相应的缓存记录信息。

智能匹配不仅支持标准的DNS报文通信协议（RFC1035），同时支持越来越受到重视的edns-client-subnet协议（RFC2671），即不根据数据报文的源IP，而是通过解析在RR记录的Addition字段中的相关协议记录来进行智能地址库匹配，为中转DNS解析服务器或代理返回以用户源网段为准的DNS应答报文。

用户当需要使用我们的智能DNS服务时候，只需要通过互联网访问我们的网站服务平台，并进行简单的用户注册，然后把域名的解析指向到本系统提供的域名地址并设置好相应的解析记录，即可使用我们DNS智能解析服务功能。

下图为DNS智能解析服务网站平台各功能截图:



网站首页

用户注册

域名管理



解析记录

### 2.5.3用户实时域名状态监控

对于系统使用用户来说，在进行了域名和解析记录的相关配置及授权后，很多时候需要获得相关的途径和功能模块去了解使用本系统后的业务运行情况。因此，本系统在进行多次用户需求统计和论证后，设计出多种人性化且贴合实际的业务监控报表功能模块以供用户选择和使用，让用户获得更好的用户体验。

用户域名实时状态监控包含多个细分功能模块，具体如下：

1）用户域名解析量实时统计

对于用户来说，域名解析量是最为关心也是一个最为基础的数据，该数据将实时反映当前域名解析业务情况是否正常、对于域名访问用户解析请求是否生效，并能让域名拥有者能及时了解到进行智能解析业务域名及其子域名的访问情况。

为此，我们采用基于分布式大数据存储的实时数据存储框架，在各缓存节点直接对解析访问情况进行实时记录并归并到数据存储服务中。经过数据分析和处理后，即可让用户实时了解当前各域名及子域名的解析量统计报告。在业务满负载的情况下，依然可以保证用户查看到延迟在一小时以内的解析量统计信息，高效稳定。

用户可以通过多种条件组合的方式，查看当前已经设置解析记录的域名解析统计总量和各子域名各自的解析统计，时间跨度支持涵盖当天、最近一周、最近一月、最近一季、最近半年、最近全年等，时间精度最小以小时为单位。使得用户能根据业务需要精确定位到解析统计的各个重要时间节点，如峰值定位等。

下图为域名解析量实时统计功能截图:



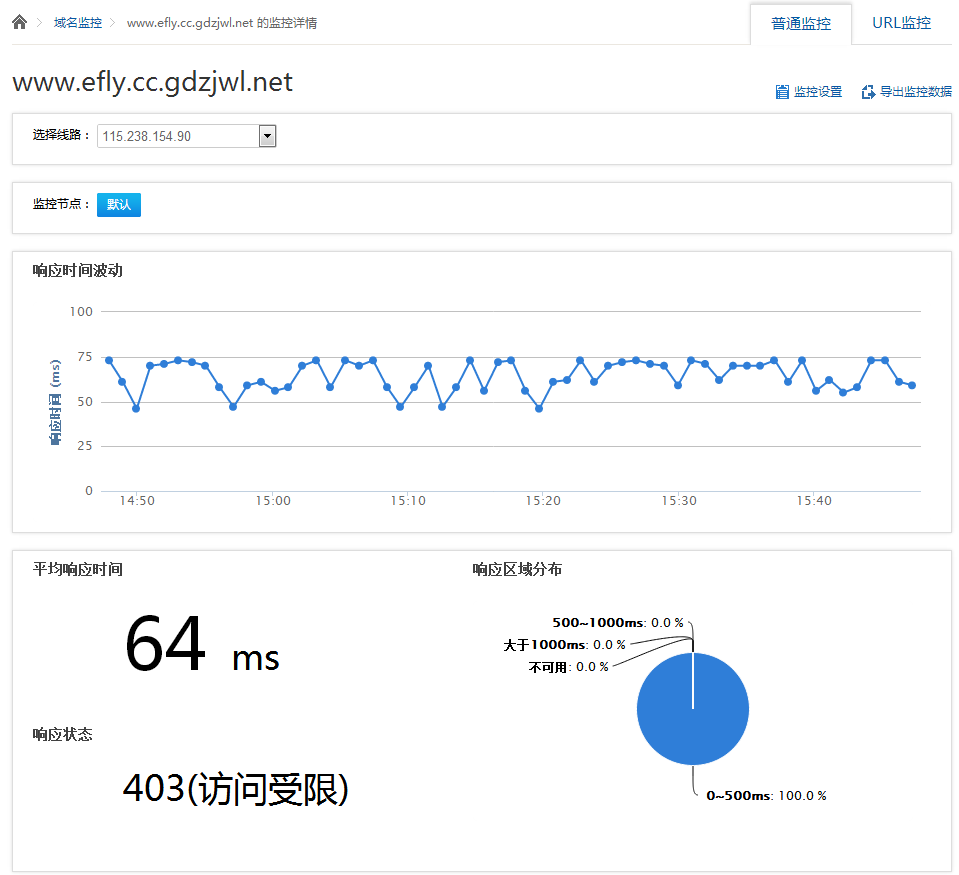
2）用户域名访问状态实时监控

为了保证用户设置的解析记录的有效性，使业务能够有效正常运行，系统提供该增值服务让用户能够对当前域名的访问情况有更好的驾驭能力，同时当服务器出现问题时，用户能够在第一时间获得相关提示信息以及时处理故障，保障业务正常运行。

系统提供用户域名访问状态实时监控模块，让用户能够对已托管到平台的域名下的所有已设置解析记录进行实时监控，监控信息包括解析指向服务地址最近一小时的精确响应时间白标、当前http状态、当前是否连通、最近一小时平均响应时间等。用户在设置响应监控域名后，能够简单方便地从平台上查看指定域名的访问有效性，确保业务的正确运行，并及时修正设置错误的解析记录设置。

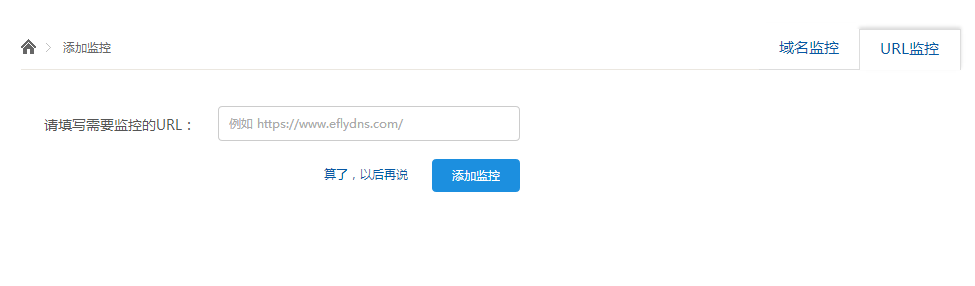
同时，域名访问状态实时监控模块还支持自动报警提醒功能，当用户进行监控配置后，系统会定期对指定域名的有效连通性进行测试，当发现域名访问状态出现问题时，将把出错提示信息根据用户配置自动发送警告邮件提醒用户，让用户能够第一时间知悉域名访问状态实时情况，避免因业务瘫痪造成损失。同时，智能自动报警提醒的功能存在使得用户在配置后不需要派驻相关人员长期对系统报表进行监控，为用户节省业务运营人力资源成本。下图为域名访问状态实时监控功能截图:





3）自定义url访问状态实时监控

除了对于已设置解析记录指向地址的实时监控，用户很多时候可能还需要根据业务情况对指定url地址进行访问状态的实时监控（如cname设置指定地址不在本系统解析、域名网站中对某些url有强制依赖等情况）。因此，我们除了为在系统中当前有解析记录的地址提供状态实时监控外，还支持用户对指定自定义的url进行监控。用户只需要在监控模块中填入需要监测的url地址，系统即可直接对其开展监控，并如域名记录状态检测一样支持智能报警功能。下图为监控添加界面：



url对应的数据报表功能与界面和上文第二节中用户域名访问状态实时监控类似。具体如下图所示：



### 2.5.4域名智能诊断和指向检测

对于新用户或者对DNS业务流程不熟悉的用户来说，初次使用本智能解析系统并不是一件容易的事，因此，系统中特意根据用户实际需求场景，为用户量身定制出了一些能辅助用户上手的人性化智能模块，让用户能够更便利地使用本系统。

1)域名指向检测

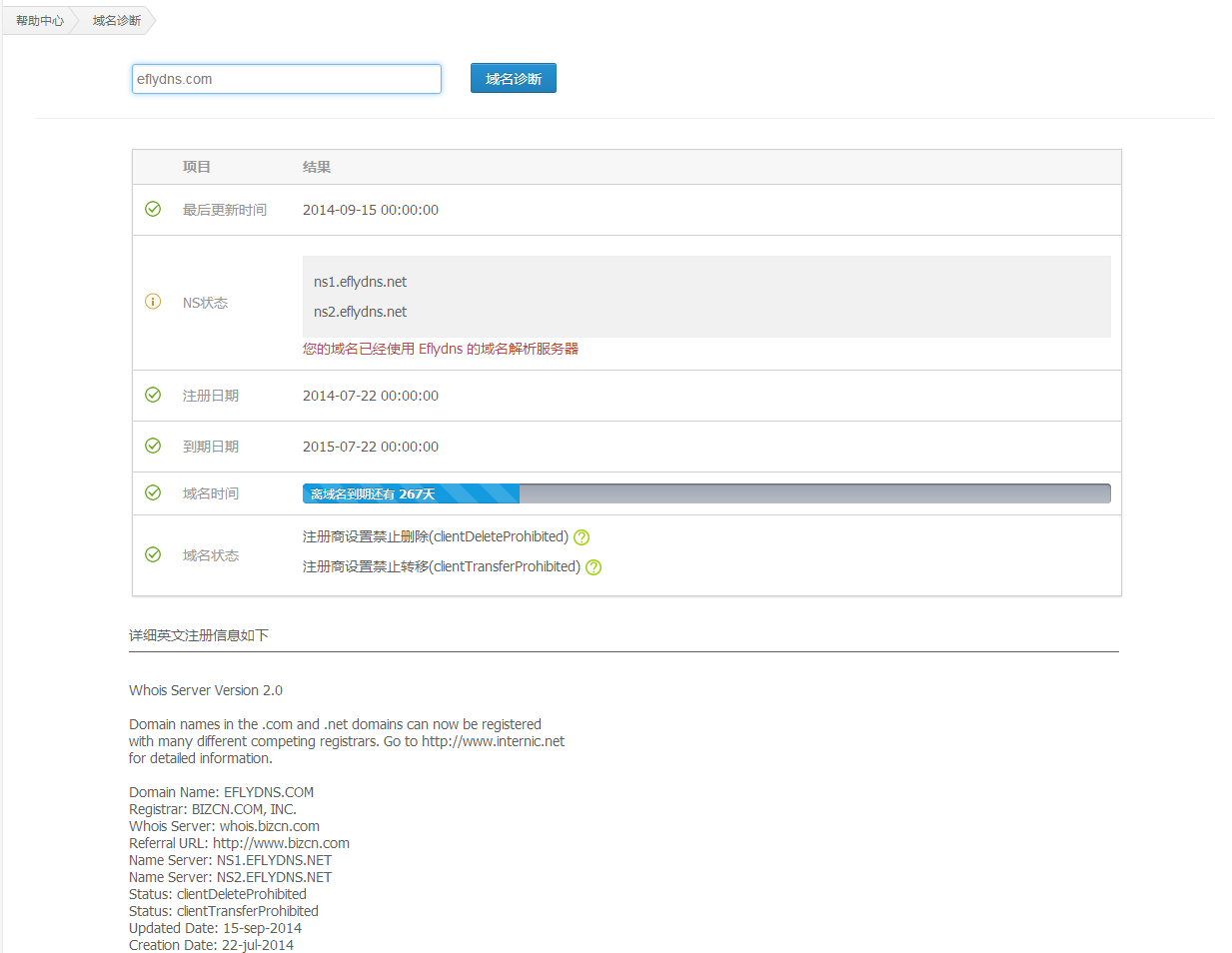
当用户初次使用本系统时，第一步首先要做的就是把域名的解析授权到本系统。但是，由于域名的源注册商不同，导致域名授权修改的步骤各有难易不同，很容易让用户手足无措；同时，很多时候由于用户的操作没有到位或者缓存的原因，导致用户自以为已经把域名解析授权到本系统，但实际上却并不是这样，从而造成域名不能够正确解析。因此，我们除了为用户编写量身定制的用户手册指导文献以外，还加入了一个智能域名指向检测系统。当用户把一个新域名添加到我们的系统中后，系统将会实时对新添加域名的指向状态进行智能检测，当发现当前指向地址不是本系统时，将会向用户显示相应提醒，让用户能够掌握当前域名的授权更改是否已经生效，从而更好地使用域名解析服务。下图为智能域名指向检测功能截图:



2)域名智能诊断

对于用户来说，对注册域名的基本消息的获取和掌控是非常重要的；同时，系统需要一个客户自助接口来让用户检查自己当前域名是否存在问题。因此，我们为用户提供一个智能域名诊断工具，当用户输入域名后，该工具能自动为客户分析该域名是否存在问题，检查域名当前是否已使用本系统的解析服务，同时为用户罗列出该域名的所有详细基本相关信息，让用户能够对自己域名的基本状态一目了然，从而可以自己排除一些简单的错误问题，大大减轻了客服资源运维消耗。

域名诊断工具能够为用户检索域名的最后修改时间、NS授权状态、注册日期、到期日期、域名状态、Whois原文信息等，一键智能完成，非常简单方便。



## 

## 2.6技术关键点及难点攻关

### 2.6.1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，使得服务器能处理10GB接口链路上的数据流，网络流量峰值达到1200万pps。 |
| 快速报文透传 | 使用最短协议栈的处理规则，维持非业务流程的完整性，让非业务通讯正常进行。 |
| 路由优化 | 为零拷贝高速接口提供路由支撑，使之跟系统路由配置具有同的功能，灵活应对业务需求。 |
| 请求应答和报文组装角色分离 | 请求应答、报文组装和数据库存储将分别部署在三部分各自互联的服务器集群上协同工作。使得请求请求应答服务器的角色功能更简单专注、快捷高效；核心报文组装和数据存储模块可以不再对外部公开，作为内部通信服务，得到有效的外部隔离和安全性保证。外部的异常流量和入侵行为都由请求应答服务器节点来分别承抗。 |
| 高速缓存功能 | 根据用户配置智能预缓存，并根据DNS协议定义对应匹配规则提高检索速度，通过高速网络收发接口作为系统低层支撑架构最大化业务报文的吞吐量 |
| 故障智能切换 | 通过把DNS业务需求与NQA技术进行修改融合与创新，利用Track技术把网络监控事件和策略路由配置能够相互响应，使得可以通过在网络层路由器中部署相应监控模块，实现智能流量有效性监测与故障指向流量自动切换功能。 |
| 基于业务大数据的故障后备服务 | 利用零拷贝高速收发接口收取缓存代理服务器产生的日志，保存到hadoop hdfs上；然后利用hadoop进行数据分析，提取出所需的业务数据，并做好归并存放到mongodb集群中。  后备服务程序通过实时查询mongodb集群匹配出该域名类型最近一次访问的记录，并以标准的DNS协议格式返回给缓存代理服务器。 |

### 2.6.2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 | 很多时候DNS系统要处理大量的高并发的请求，为了提高系统的吞吐量，以充分发挥硬件的性能，需要开发一套完整的高速的报文处理系统。 |
| 高速报文转  发 | 经过大量的试验我们研究出了一种可以有效提高业务报文的处理速度，同时又不影响非业务报文的处理流程的方法，这种方法让DNS系统的数据吞吐量有了质的提升。 |
| 私有通信协议研制 | 设计私有通信协议时，因为存在智能线路匹配的需求，不仅要在在传输的DNS数据报文负载中加入IP库智能匹配的相关信息，同时还要兼容传统RFC1035所规定的协议信息。 |
| Edns-client-subnet协议支持 | 根据当前互联网发展趋势，本系统除了支持传统的DNS标准协议外(RFC1036)外，还需要支持应用日益广泛的Edns-client-subnet协议（RFC2671） |
| 故障流量智能自动切换 | DNS解析服务系统对于业务的稳定性要求非常高，为防范不可免因素造成的故障发生，需要实现一套方案使得当DNS架构系统中部分设备出现业务中断时，业务流量能被自动引导到可用服务器 |
| 路由优化模块 | 为了适应多变的网络环境和业务需求，同时做到与现有系统无缝接合，需要实现一套高速的路由匹配规则。 |

**2.7系统性能报告**

**2.7.1前提环境**

DNS智能解析服务系统,主要目的有三个方面。

其一，为所有互联网企业与个人用户提供强大的智能DNS智能解析功能，改善用户域名的访问速度和业务质量，提升优化整个互联网网络访问质量环境。

其二，根据业务运营中收集到的业务记录日志，通过大数据数据挖掘技术从中提取出有价值的信息，通过这些信息为用户创造出更多有价值的功能和数据参考，形成数据良性循环。

其三，为用户域名解析业务提供实时状态数据监测功能，使客户可以实时的监测到当前域名中的统计数据和访问质量，能够第一时间发现和定位域名解析业务流程中出现的问题。

因此出于对以上服务的需求，对我们的服务器稳定运行也是一大考验。所以项目使用一线的服务器厂家为强大的服务后盾，同时对操作系统和软件做一次开发或二次开发。性能和功能上大大满足客户应用的需求。

**2.7.2网络环境**

对于服务器来说，网络环境至关重要。有一个良好的网络环境，为客户服务解决的一个个资源紧张，带宽不足。特别对于DOS（拒绝服务攻击）充足的带宽十分重要。我们节点网络接入，使用千兆二层交换机（H3C S5120-EI）。直接上联到各大运营商的机房核心设备上，通过千兆网络的互连，一来解决了服务器对外服务的带宽需求；二来遇到DOS(拒绝服务攻击)时候,解决带宽的紧张问题。

**2.7.3配置信息**

使用睿江DNS智能解析服务系统基本配置，下图为基本配置清单：

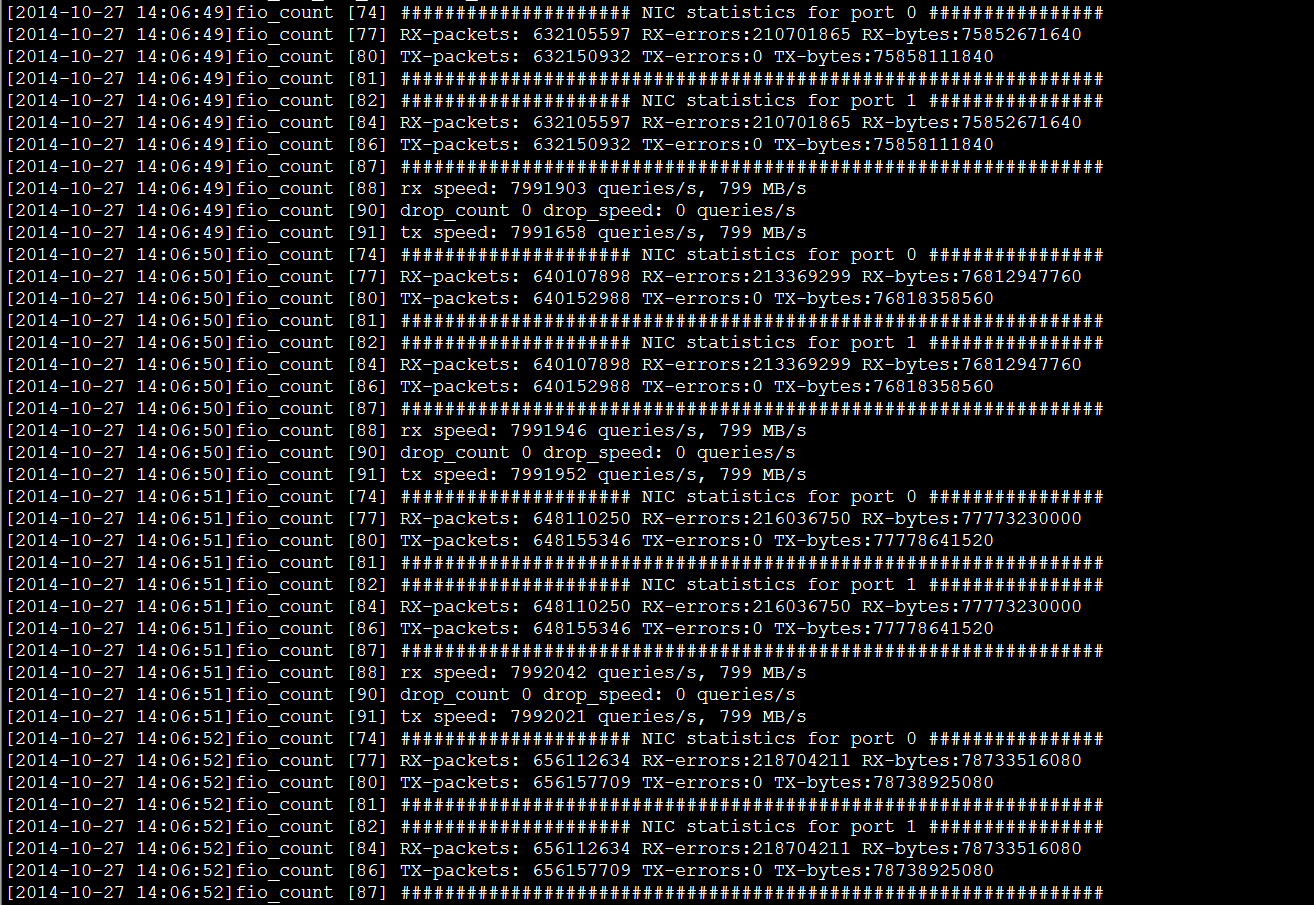
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 厂家 | 型号 | CPU | 内存 | 硬盘 | 操作系统 |
| Dell | R610 | Intel XeonE5620\*2 | 24G | 146G\*2 SAS | CentOS 6.5  64bit |

其中，日志存储、数据库存储的服务器，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 | R610 | Intel Xeon E5620\*2 | 24G | 300G\*2 SAS+1T\*2 SAS | CentOS 6.5  64bit |

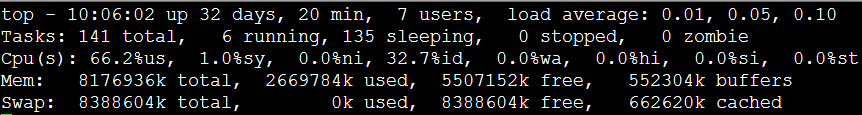
**2.7.4并发数据**

经过对单台服务器的优化配置，测试单一缓存节点服务器的网络数据报文处理能达到800Wpps，下图是某次的测试数据采样：

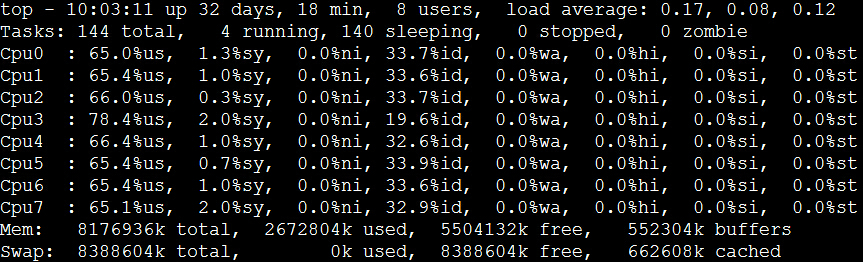


**2.7.5服务器负荷**

经过对业务缓存服务器进行压力测试，单服务器消耗CPU性能在70%（即使：30%id空域），可以在业务网络环境到达200万pps的业务数据并发处理能力。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：



## 2.8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 | 2014/3/1 | 2014/3/31 | 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 | 2014/4/1 | 2014/5/31 | 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 | 2014/6/1 | 2013/6/31 | 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 | 2014/7/1 | 2014/7/30 | 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 | 2014/8/1 | 2014/9/30 | 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。 |