DASFlow: Uma Arquitetura Distribuída de Processamento e Armazenamento para Dados de Monitoramento de Rede

Diego J. Hoss[1,3], **Christian Lyra**[2], **Carmem S. Hara**[1]

[1]Departamento de Informática – Universidade Federal do Paraná (UFPR)
Caixa Postal 19.081 – 81.531-990 – Curitiba – PR – Brasil

[2]Ponto de Presença da RNP no Estado do Paraná – PoP-PR
UFPR – Centro de Computação Eletrônica

[3]Departamento de Infraestrutura em TI
Universidade Tecnológica Federal do Paraná (UTFPR)

{djhoss,carmem}@inf.ufpr.br, lyra@pop-pr.rnp.br

Abstract. *Network monitoring is an activity that demands an increasing storage and processing capacity. This follows from the continuous growth of the data volume that are transmited daily on the Internet. Monitoring tools are generally based on a centralized architecture. The centralized model has limitations associated with scalability that are inherent to the architecture, such as the lack of redundancy and a single point of failure. In this paper, we propose a distributed architecture called DASFlow. This solution aims at overcoming these limitations and providing scalability for the NfSen network monitoring tool.*

Resumo. *O monitoramento de redes é uma atividade que demanda cada vez mais capacidade de armazenamento e processamento. Isto decorre do contínuo aumento no volume de dados que trafega diariamente na Internet. Para realizar esta atividade, são utilizadas ferramentas de monitoramento que em geral possuem arquitetura centralizada. O modelo centralizado possui limitações associadas à escalabilidade que são inerentes à arquitetura, tais como: falta de redundância e ponto único de falha. Neste trabalho, é proposta uma arquitetura distribuída chamada DASFlow. Esta solução visa contornar os problemas decorrentes da arquitetura centralizada e prover escalabilidade para a ferramenta de monitoramento de rede NfSen.*

1. Introdução

A contínua pervasividade da tecnologia nos diversos setores da sociedade acarretou uma mudança de paradigma no modo como é feita a comunicação entre pessoas, empresas e máquinas. Essa nova era trouxe um aumento exponencial no volume de dados que trafega diariamente na Internet [Cisco 2013]. Por conta desse acréscimo e também da evolução da tecnologia, as redes de comunicação tornaram-se cada vez mais complexas em sua estrutura. Desta forma, monitorar e gerenciar grandes redes tornou-se uma tarefa desafiadora, especialmente pela dificuldade em lidar com enormes volumes de dados.

No contexto da gerência de redes, o monitoramento é realizado utilizando-se diversas ferramentas conhecidas como ferramentas de monitoramento de rede. Grande parte destas ferramentas possuem limitações em relação à manipulação de grandes massas de

dados. Uma das principais é a arquitetura com a qual efetuam o armazenamento e processamento dos dados de monitoramento, que em geral é centralizada. O modelo centralizado possui limitações inerentes à arquitetura, podendo ser destacados a falta de redundância, um ponto único de falha e a ausência de balanceamento de carga.

Atualmente, alguns trabalhos encontrados na literatura buscam prover escalabilidade para a atividade de monitoramento de rede e contornar estes problemas [Gao et al. 2011, Li et al. 2011]. Estes trabalhos diferem do proposto neste artigo por direcionarem seus objetivos para a escalabilidade de apenas um aspecto do monitoramento, como processamento e recuperação [Morariu et al. 2010], armazenamento [Morariu et al. 2008] ou coleta [Deri and Fusco 2013].

Uma das ferramentas utilizadas para a atividade de monitoramento é o NfSen/Nfdump [Haag 2009]. Ela realiza as funções de coleta, armazenamento, processamento e apresentação do tráfego de rede. Sua interface gráfica fornece um ambiente intuitivo e de fácil utilização. Por meio de consultas pré-existentes, ela é capaz de separar e apresentar os resultados em forma de gráficos para diferentes objetivos. A ferramenta é muito utilizada em diversas instituições por ser gratuita e possuir ampla documentação. Embora bem aceita pelos administradores de rede, o NfSen/Nfdump possui arquitetura centralizada e está sujeita, portanto, às limitações deste modelo.

Neste sentido, o objetivo deste trabalho é investigar soluções a fim de prover escalabilidade para a atividade de monitoramento de rede. Para isto, criou-se uma arquitetura distribuída chamada DASFlow aplicada à ferramenta NfSen. Ela é composta de um conjunto de servidores de armazenamento, de forma que o processamento sobre estes dados possa ser executado em paralelo. O funcionamento da arquitetura é baseado no desenvolvimento de dois módulos, chamados StoreDAS e QueryDAS, que atuam em todos servidores, permitindo sua integração e cooperação para a execução das tarefas. O armazenamento dos dados de monitoramento é realizado utilizando um sistema de arquivos distribuído (SAD).

Com isso, é esperado que a arquitetura proposta neste trabalho contemple os três aspectos do monitoramento, fornecendo uma solução que permita ao NfSen realizar de modo escalável as etapas de coleta, armazenamento e processamento dos dados de monitoramento de rede.

O restante deste artigo está organizado da seguinte forma: na seção 2 são descritas definições e objetivos do monitoramento de rede. Na seção seguinte são apresentados trabalhos que investigam soluções para o problema da escalabilidade no monitoramento de rede. A seção 4 apresenta a arquitetura distribuída como alternativa ao modelo centralizado. A implementação da solução proposta é descrita na seção 5. Os estudos experimentais e resultados são apresentados na seção 6. A seção 7 conclui o artigo indicando trabalhos futuros e possíveis melhorias na proposta.

2. Monitoramento de Rede

O monitoramento de rede é uma atividade relacionada à gerência de redes e é composta por quatro etapas: coleta, armazenamento, processamento e apresentação do tráfego monitorado [Brownlee et al. 1999].

Muitos são os motivos que levam as organizações a monitorarem seu tráfego de

rede. As operadoras de telecomunicações, por exemplo, usam a fase de monitoramento para obter informações a fim de realizar a cobrança pelo *link* comercializado. Ainda por meio desta etapa, elas otimizam a utilização da banda com técnicas de qualidade de serviço (do inglês, *Quality of Service - QoS*). Questões relacionadas ao desempenho e segurança também são consideradas. Com o correto monitoramento do tráfego, pode-se entender o comportamento da rede e prevenir, ou minimizar, problemas de ataques de negação de serviço (do inglês, *Denial of Service - DoS*).

Para realizar o monitoramento, diversas ferramentas podem ser utilizadas, tais como: Tcpdump [TCPDUMP/LIBPCAP 1999], Wireshark [Wireshark 2006], Ntop [Deri 2001], Flow-tools [Fullmer 2000] e NfSen/Nfdump [Haag 2009].

O NfSen/Nfdump é uma ferramenta de código aberto capaz de realizar o armazenamento, processamento e apresentação dos dados de monitoramento. Os dados de monitoramento utilizados pelo NfSen são conhecidos como fluxos IP. Um fluxo IP contém informações contabilizadas sobre uma comunicação de rede unidirecional ou bidirecional, entre uma origem e um destino em um intervalo de tempo [Claise 2004]. A estrutura de um fluxo IP possui variações de acordo com o protocolo utilizado. Todos os protocolos no entanto, possuem ao menos as seguintes características: IP origem, IP destino, porta de origem, porta de destino, protocolo de comunicação (TCP, UDP, ICMP) e quantidade de dados trafegado (em *bytes*) [Claise 2013, sFlow Protocol 2004].

O NfSen é responsável pela interface amigável com o usuário. Seu funcionamento consiste em fornecer recursos de visualização sobre os dados de monitoramento. Para isto, o administrador interage com a ferramenta por meio de consultas pré-determinadas, e obtém os resultados com o auxílio de gráficos e relatórios. Além disso, o NfSen provê mecanismos para que o usuário defina sua própria consulta, atuando em conjunto com as consultas pré-existentes no sistema.

O Nfdump é a ferramenta base para o funcionamento do NfSen. Ela é responsável pela manipulação dos fluxos IP. Isto é, ela realiza a coleta, trata o armazenamento e responde as consultas requisitadas pelo NfSen. Para a tarefa de coleta e armazenamento dos dados, o Nfdump agrupa informações de acordo com o tempo em que foram recebidas. Por exemplo, os fluxos IP recebidos no dia 15/02/2015 entre às 11:00 e 11:05 da manhã, são agrupados e armazenados em um arquivo chamado `nfcapd.201502151105`.

Na etapa de processamento, a ferramenta aceita parâmetros que podem modificar o formato de saída de uma consulta. Estes parâmetros podem ser utilizados, por exemplo, para realizar a ordenação e agregação dos resultados. Um dos parâmetros do Nfdump utilizado para modificar o formato de saída é o '-w'. Ele permite que o resultado de uma consulta seja armazenado em um arquivo com o mesmo formato de entrada. Isto significa que resultados parciais podem ser direcionados para um arquivo e reutilizados nas próximas consultas que envolvam o mesmo conjunto de dados. O NfSen/Nfdump embora muito utilizado, possui limitações relacionadas à escalabilidade porque atua de modo centralizado nas etapas de coleta, processamento e armazenamento dos fluxos IP.

Na próxima seção, são apresentados alguns trabalhos que buscam soluções para melhorar o desempenho das etapas envolvidas na atividade de monitoramento de redes.

3. Trabalhos Relacionados

Uma solução escalável para o monitoramento de redes deve oferecer condições para que as tarefas de coleta, armazenamento e processamento possam ser realizadas ajustando-se ao aumento do volume de dados manipulados.

Em [Deri et al. 2013], os autores apresentam uma solução de coleta para redes de 10 Gbps, que tem como objetivo coletar todos os dados sem perdas. A abordagem foca na otimização dos recursos computacionais e classificação dos dados a serem capturados. Embora a proposta obtenha uma capacidade maior de coleta em ambientes com alto tráfego de dados, esta solução não provê escalabilidade, pois preserva um único nodo para coletar os dados de monitoramento. Outra abordagem para coleta é apresentada em [Morariu and Stiller 2008]. Este trabalho adota uma solução distribuída para a coleta através de uma rede *Peer-to-Peer* (P2P). Ainda que seja dinamicamente expansível, esta solução limita-se a apenas efetuar a coleta dos dados, não considerando as demais atividades do monitoramento.

Alguns trabalhos concentram esforços em soluções de armazenamento. [Li et al. 2011] apresenta uma proposta de armazenamento ágil dos dados. Neste trabalho, os autores buscam coletar grandes quantidades de dados, tornando-os disponíveis para as ferramentas de monitoramento o mais rápido possível. Para aumentar a capacidade de armazenamento, a solução é capaz de manipular vários discos simultaneamente. No entanto, ela preserva a figura de um único ponto de armazenamento, limitando-se a uma otimização do modelo centralizado. Em contrapartida, soluções de armazenamento distribuído são adotadas em [Morariu et al. 2008], [Deri and Fusco 2013] e [Lee et al. 2010]. Nestes trabalhos, os autores utilizam um conjunto de nodos interligados entre si de modo a prover um armazenamento expansível dinamicamente. Ainda que sejam soluções escaláveis, elas atentam-se para o desempenho no armazenamento dos dados de monitoramento, não considerando outras etapas como processamento ou apresentação.

Em termos de escalabilidade de processamento, tem-se em [Morariu et al. 2010] uma proposta que distribui o processamento dos fluxos IP entre vários nodos. Cada nodo atende conjuntos específicos de consultas, isto é, associadas a alguns tipos de parâmetros, tais como: protocolos, endereçamento e tipo de serviço. Nesta solução, várias consultas podem ser realizadas simultaneamente sem sobrecarregar os nodos de processamento. Em [Lee and Lee 2013], a solução proposta é semelhante ao encontrado em [Morariu et al. 2010]. Neste trabalho porém, os autores utilizam o Hadoop [Hadoop 2005] para o processamento distribuído das consultas. Observa-se em ambas as propostas que a solução não considera as etapas de coleta e armazenamento.

É possível observar nos trabalhos apresentados que os autores atentam para o fato de lidar com grandes volumes de dados. Contudo, as abordagens utilizadas geralmente tratam as etapas de forma isoladas. Isto é, não buscam resolver o problema da escalabilidade de modo geral. Neste artigo é proposta uma solução que investiga como prover escalabilidade para todas as atividades do monitoramento de rede utilizando a ferramenta NfSen/Nfdump.

4. A Arquitetura DASFlow

As ferramentas de monitoramento de rede por vezes demandam mais recursos computacionais do que possuem à disposição. Isto é percebido quando arquivos são descartados por falta de espaço, ou, quando consultas tornam-se mais demoradas à medida em que se aumenta o volume de dados requisitados. Uma das abordagens possíveis para melhorar o desempenho na execução das consultas e aumentar a capacidade de armazenamento é utilizar um ambiente distribuído. Esta solução no entanto, deve considerar o objetivo para o qual os dados serão recuperados para determinar a forma de distribuí-los. Para isto, é necessário saber os tipos de consultas que são aplicadas sobre os dados e com que frequência elas ocorrem.

Em geral, a ferramenta NfSen é utilizada para responder dois tipos de consultas: gerenciais e investigativas. Consultas gerenciais envolvem dados históricos, também conhecidos como dados *frios*, relativos a semanas, meses e anos. Elas fornecem informações sobre o comportamento da rede monitorada, atendendo questões como: qual a evolução da utilização do *link* de dados? Qual protocolo vem crescendo em utilização nas últimas semanas? Qual o tráfego trocado entre Sistemas Autônomos (do inglês, *Autonomous System - AS*). Este tipo de consulta auxilia na compreensão e planejamento dos investimentos destinados à infraestrutura de rede.

As consultas investigativas estão relacionadas a incidentes de segurança. Para estes casos, as consultas executadas visam investigar e encontrar os causadores de situações anômalas. Estas consultas geralmente são realizadas sobre dados recentes, ou dados *quentes* e envolvem períodos curtos que duram entre 30 minutos e 12 horas. As consultas mais comuns para este objetivo respondem questões como: qual rede ou IP com maior tráfego? Qual porta e protocolo mais utilizados? Qual o motivo de haver muito tráfego de várias origens para um único destino?

Tanto consultas investigativas bem como consultas gerenciais usualmente utilizam o recurso de filtragem dos dados. Para o NfSen/Nfdump, o filtro é um parâmetro utilizado na consulta que pode ser aplicado sobre algumas das características dos fluxos IP, tais como: endereços IP de rede ou dispositivo, portas de comunicação, protocolos, tipos de serviço, quantidade de pacotes e *flags* de comunicação.

Um sistema que trata a questão de escalabilidade no contexto descrito deve considerar diferentes práticas de distribuição e replicação dos dados. A adoção de diferentes práticas tem como objetivo melhorar o desempenho das consultas submetidas ao sistema, mesmo com o aumento do volume de dados armazenados de forma distribuída. A arquitetura do sistema proposta neste trabalho, chamada DASFlow, é apresentada na Figura 1.

A arquitetura considera um conjunto de nodos. Cada nodo desempenha o papel de cliente, servidor ou ambos. O cliente é responsável por receber consultas submetidas pelo administrador de rede, distribuí-las pelos servidores, processar os resultados parciais enviados pelos servidores e retornar o resultado da consulta. Os servidores são responsáveis pelo processamento parcial da consulta sobre os dados armazenados no próprio servidor.

O componente que controla a distribuição e replicação dos dados de monitoramento é o módulo de metadados. Ele associa intervalos de tempo a um conjunto de nodos servidores a partir de um arquivo de configuração dado como entrada. O arquivo

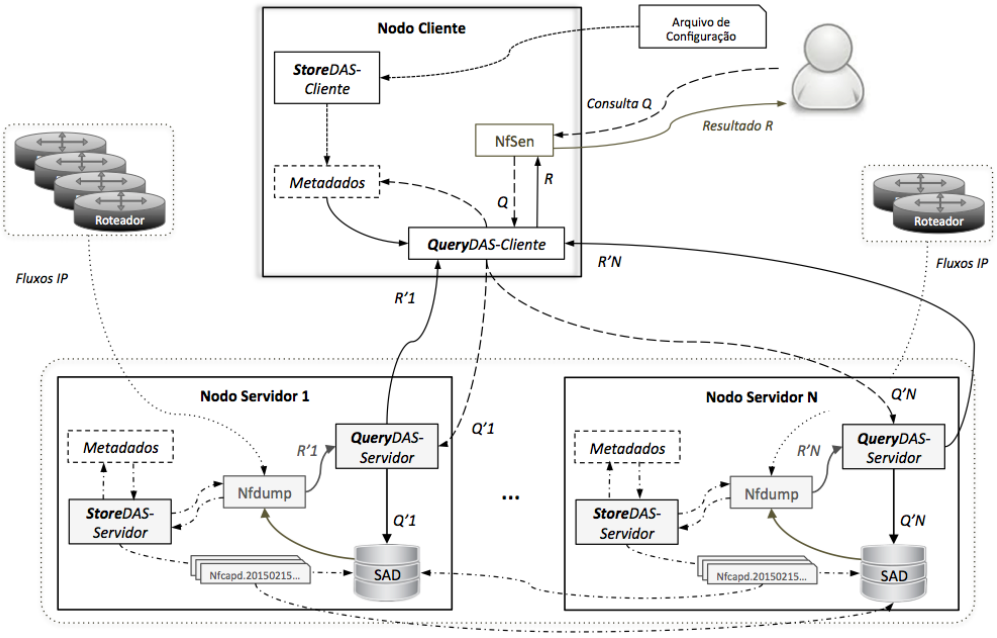

Figura 1. DASFlow: Arquitetura distribuída de processamento e armazenamento.

de configuração pode, por exemplo, determinar que os dados das últimas 12 horas sejam replicados em um conjunto de servidores {S1,S2, S3}, enquanto apenas uma cópia dos dados mais antigos sejam mantidos divididos entre todos os servidores. É importante observar que os metadados são replicados em todos os nodos do sistema. Assim, o servidor que recebe os dados de monitoramento sabe para onde os arquivos devem ser encaminhados para armazenamento sem consultar um nodo cliente. A arquitetura prevê a existência de diversos servidores, associados a um ou mais roteadores. Dessa forma, à medida que o tráfego aumenta, é possível adicionar novos servidores ao sistema, ou seja, prover escalabilidade de coleta. Os componentes internos dos nodos são detalhados a seguir.

4.1. Nodo cliente

Um nodo cliente possui quatro componentes: a ferramenta NfSen, o módulo de metadados, os módulos StoreDAS-Cliente e QueryDAS-Cliente. A seguir são apresentados detalhes do seu funcionamento.

Ferramenta NfSen: Este componente é a interface de comunicação entre o administrador de rede e a arquitetura. Ele é responsável por receber a consulta sobre os dados de monitoramento e apresentar os resultados. É importante observar que para um administrador que já utiliza a ferramenta NfSen, não há nenhuma alteração na interface com a qual ele já está habituado, sendo as questões de escalabilidade tratadas internamente pela arquitetura.

Metadados: Os metadados contém informações sobre a distribuição e replicação dos dados de monitoramento. Na arquitetura proposta, optou-se por não alterar a forma nativa do Nfdump de armazenar dados de fluxo em arquivos identificados por intervalos de tempo (como `nfcapd.201502151105`). Assim, os metadados associam intervalos de tempo (t) a conjuntos de servidores (S). Ou seja, arquivos gerados dentro de um

determinado intervalo t são replicados em todos os servidores do conjunto S.

StoreDAS-Cliente: Este é o componente que recebe o arquivo de configurações que determina a distribuição e replicação dos arquivos e atualiza os metadados em todos os nodos no sistema. O StoreDAS-cliente também é responsável por monitorar o tempo de vida dos arquivos armazenados em cada servidor. Assim, dados que eram considerados quentes e passaram a ser frios podem ser removidos dos servidores que não são os responsáveis por manter os dados históricos daquele intervalo de tempo.

QueryDAS-Cliente: Este módulo coordena o processamento de consultas. Para isso, ele acessa os metadados para encontrar os servidores que armazenam os dados envolvidos em uma consulta Q e gera subconsultas (Q') sobre cada subconjunto de dados. Estas subconsultas são enviadas para processamento para os servidores, que retornam os resultados parciais (R'). Os resultados parciais são armazenados em arquivos temporários criados por meio da execução do Nfdump em cada nodo servidor. O módulo QueryDAS-Cliente executa então o processamento final sobre as saídas parciais para gerar o resultado completo (R). O resultado final é gerado pela execução da consulta completa (Q) sobre os resultados (R') recebidos dos nodos servidores.

4.2. Nodo servidor

O armazenamento dos arquivos de monitoramento, bem como o processamento de consultas sobre estes arquivos são realizados pelos nodos servidores. Os componentes que atuam nestes nodos são apresentados a seguir.

Nfdump: este componente faz parte da ferramenta NfSen/Nfdump e é responsável por realizar a coleta e processamento parcial das consultas.

Sistema de arquivos distribuído (SAD): este elemento está disponível em todos os nodos servidores formando um ambiente de armazenamento distribuído. Ele é responsável por permitir o acesso aos arquivos a partir de qualquer nodo da arquitetura, através de um espaço de nomes de arquivos comum.

StoreDAS-servidor: Este módulo é responsável por prover escalabilidade de coleta à arquitetura, atuando como um intermediador entre a coleta realizada pelo Nfdump e o sistema de arquivos distribuído. Assim que o Nfdump gera o arquivo contendo os fluxos IP, o StoreDAS-servidor adiciona ao seu nome informações de identificação, isto é, a qual roteador o tráfego pertence. Este arquivo é então armazenado em um ou mais servidores de armazenamento, de acordo com as informações obtidas dos metadados.

QueryDAS-servidor: Este módulo é responsável por receber uma consulta parcial (Q') enviada pelo QueryDAS-cliente, e, buscar os dados necessários para atendê-la. Após esta etapa, o módulo solicita ao Nfdump que execute o processamento sobre os arquivos recuperados do SAD. Neste passo, o Nfdump utiliza o parâmetro '-w' para criar uma saída temporária que será armazenada em um arquivo. Esta saída possui o formato de entrada do Nfdump, logo, é possível reutilizar este resultado para novas consultas. Ao concluir o processamento, o resultado parcial (R') é enviado ao nodo cliente. O QueryDAS-servidor e QueryDAS-cliente são os componentes responsáveis por prover a escalabilidade de processamento de consultas.

Um dos requisitos esperados da arquitetura DASFlow é a portabilidade e continuidade do projeto. Assim, foi determinado que a implementação da solução seria na forma

de módulos independentes da ferramenta original. Deste modo, eles podem ser anexados à ferramenta NfSen exigindo alterações mínimas no código original. Adicionalmente, a arquitetura proposta visa atender de modo escalável e na mesma solução os três aspectos do monitoramento. A coleta e o armazenamento são atendidos pelo sistema de arquivos distribuído e pelos módulos StoreDAS. Para a escalabilidade de processamento, os módulos QueryDAS dividem uma consulta (Q) em subconsultas (Q') para obter a cooperação de cada nodo servidor na execução da consulta completa.

5. Implementação

A implementação do módulo QueryDAS-cliente foi realizada utilizando a linguagem de programação PHP, a mesma da ferramenta NfSen. A integração com a ferramenta ocorre por meio da adição de uma linha em um dos arquivos originais. Após esta modificação, o módulo está pronto para ser utilizado. O módulo QueryDAS-servidor foi implementado também com a linguagem PHP. Nesta versão porém, ele não atua integrado ao NfSen, mas sim como um serviço que fica "ouvindo" no nodo servidor. Para acessar o SAD, os módulos interagem com o sistema de arquivos por meio de sua API. Para o armazenamento dos dados de monitoramento, foi utilizado o sistema de arquivos distribuído Ceph. A escolha deste sistema deve-se a algumas de suas características descritas a seguir.

O Ceph é um sistema de arquivos distribuído de código aberto sob a licença LGPL. Este sistema possui flexibilidade em relação a sua utilização porque pode atuar como um sistema de objetos, blocos ou arquivos distribuídos. Isto é possível devido à implementação do RADOS [Weil et al. 2007]. Seu funcionamento consiste em criar uma plataforma de armazenamento totalmente distribuída. Esta camada de armazenamento provida pelo RADOS é chamada pelo Ceph de *Ceph Storage Cluster (CSC)*. O acesso ao CSC pode ser feito com a utilização da biblioteca **librados**. Esta biblioteca está disponível nas linguagens: C, C++, Java, Python, Ruby e PHP.

Para o armazenamento dos dados, o Ceph cria uma camada de abstração entre o armazenamento físico e o lógico. Essa abstração é equivalente a uma unidade de armazenamento e é chamada de *pool*. Os *pools* são mapeados pelo algoritmo CRUSH [Weil et al. 2006] e o resultado é conhecido como CRUSH *Map*. Neste mapa, constam as informações sobre a associação que existe entre os *pools* (unidades lógicas) e seu armazenamento nos servidores (unidades físicas). Desta forma, o Ceph baseia-se no CRUSH *Map* para determinar quando, quantas e onde as réplicas devem ser mantidas. Além disso, por meio da integração das APIs com o CRUSH *Map*, o Ceph fornece controle de localidade dos dados armazenados.

Com a utilização deste recurso, pode-se organizar o armazenamento dos dados de acordo com sua utilização, classificando os arquivos como frios e quentes como exemplificado nas seções anteriores. Para isso, é criado no arquivo do CRUSH *Map* um conjunto de regras. Estas regras são associadas às unidades físicas de armazenamento (Disco, Máquina, Rack), por exemplo: DiscoA = *regra1*; DiscoB = *regra2*. Assim, adiciona-se também no CRUSH *Map* a associação entre a regra e o *pool*, exemplo: $pool1$ = *regra1*. Logo, pode-se determinar que todos os dados pertencentes ao $pool1$ estão armazenados no DiscoA, obtendo desta forma o controle de localidade dos dados.

Na arquitetura proposta na seção 4, o mapeamento entre o arquivo de configuração do metadados e o CRUSH *Map* deve ser realizado pelo módulo StoreDAS-cliente. Na

versão atual da ferramenta este módulo ainda não foi implementado e a distribuição dos dados é diretamente configurada no sistema de arquivos distribuídos Ceph. Para isso, foi incluído no CRUSH *Map* uma regra chamada $replicaTotal$ que abrange todas as unidades físicas de armazenamento, e em seguida, foi associado um *pool* chamado $dadosQuentes$ à esta regra. Desta forma, todos os dados armazenados neste *pool* estão disponíveis em todas as máquinas do ambiente. Embora as funcionalidades do módulo StoreDAS-cliente não estejam disponíveis na sua totalidade nesta versão da implementação, os resultados experimentais apresentados na próxima seção mostram o potencial da arquitetura como um todo.

6. Estudo Experimental

Para validar a solução proposta neste artigo, foi criada em escala reduzida a arquitetura DASFlow apresentada na seção 4. Esta arquitetura possui 3 máquinas executando a implementação dos módulos QueryDAS-cliente e QueryDAS-servidor. Embora possam atuar na mesma máquina, para esta avaliação, os módulos cliente e servidor são executados em máquinas distintas. Para a etapa de coleta e armazenamento, este trabalho não considera a existência dos módulos de metadados e StoreDAS-servidor, portanto, os dados de monitoramento são replicados em todos os nodos servidores.

O nodo cliente possui um processador de 2.6 GHz, 2GB de memória RAM e 500 GB de disco a 7.200 RPM. Ele executa o módulo QueryDAS-cliente, isto é, recebe a consulta, envia para os demais nodos, executa o processamento final e devolve o resultado para o NfSen. As máquinas servidores (1 e 2) são idênticas em configuração: possuem processador de 3.2 GHz, 4 GB de memória RAM e 300 GB de disco a 7.200 RPM. Elas executam o módulo QueryDAS-servidor, logo, atendem as requisições vindas do nodo cliente. Além disso, elas possuem o Nfdump para processar as consultas parciais.

Os experimentos consideram dados originários do tráfego do PoP-PR. Foram utilizadas duas bases de dados:

T1: Arquivos gerados no período entre 00:05 e 02:00 do mesmo dia, totalizando 800 MB de dados.

T2: Arquivos gerados no período entre 00:05 e 08:30 do mesmo dia, totalizando 3,5 GB de dados.

Foram executadas dois tipos de consultas sobre T1 e T2: (SF) sem filtro; (CF) com filtro, selecionado apenas o tráfego HTTP (porta 80). A tabela 1 mostra uma síntese das consultas utilizadas no testes considerando o volume de dados por nodo servidor.

Tabela 1. Informações sobre as consultas utilizadas nos experimentos.

Volume de dados processado por nodo servidor	Consulta T1		Consulta T2	
	CF	SF	CF	SF
Qtd. de arquivos	12		51	
Tamanho (Bytes) Entrada	400 MB		1,75 GB	
Tamanho (Bytes) Saída	150 MB	1,3 GB	700 MB	5,3 GB

6.1. Resultados

Os resultados obtidos nos experimentos são apresentados na tabela 2. Os valores referem-se ao tempo de resposta das consultas em segundos. Para obtê-los foram realizadas 5

consultas, sendo as duas primeiras executadas como aquecimento. A média aritmética das 3 últimas consultas é o tempo de resposta apresentado. Para comparar os resultados, a mesma consulta é aplicada no NfSen original. O NfSen original é a ferramenta sendo executada na sua formatação clássica, em apenas um nodo. De modo a manter a igualdade nos experimentos, ela foi testada no nodo cliente, o mesmo nodo onde é executado o NfSen da arquitetura DASFlow.

Tabela 2. Tempo de resposta (em segundos) das consultas executadas.

Consulta	T1/CF	T1/SF	T2/CF	T2/SF
NfSen Original	12.343	21.251	69.155	114.461
DASFlow	10.203	58.327	59.142	269.692
Diferença	**2.140 (17,4%)**	**-37.076 (174,5%)**	**10.013 (14,5%)**	**-155.231 (135,6%)**

Observa-se que a consulta do tipo sem filtro (SF) quando distribuída, obtém tempo de resposta muito alto em relação a consulta original. Tanto para T1 como para T2 o tempo de resposta mais que dobrou com a execução distribuída do DASFlow.

Este resultado explica-se pela forma como o Nfdump executa o processamento parcial. Por não haver filtros, cada nodo servidor processa sua entrada e gera como saída padrão uma massa de dados aproximadamente 3 vezes maior que a entrada. Esta saída possui um alto custo relacionado ao envio do resultado para o nodo cliente e o processamento final que ele executa sobre os dados.

Em contrapartida, quando a consulta é executada com filtro (T1/CF e T2/CF), observa-se que a saída do processamento parcial é um conjunto de dados consideravelmente menor que a entrada. Para este tipo de consulta nota-se que há um ganho em tempo de resposta. Embora não seja tão expressivo, o valor é considerado importante dadas as condições de distribuição, com apenas 2 nodos executando em paralelo. O ganho tende a ser maior com a paralelização de consultas em mais nodos servidores.

Para minimizar a deficiência das consultas sem filtro, é proposta na implementação do módulo QueryDAS-cliente um controle para identificar consultas deste tipo. Deste modo, quando uma consulta é executa sem filtro, o módulo pode determinar que apenas um dos nodos que possua os dados possa executá-la. Assim, a geração de resultados parciais é minimizada e consequentemente haverá menor tempo de transmissão e processamento final pelo nodo cliente. A consulta T2/SF foi executada com apenas um nodo que possui todos os dados envolvidos na consulta e o resultado é apresentado na tabela 3.

Tabela 3. Resultados (em segundos) da consulta T2/SF com um nodo.

Consulta	T2/SF
NfSen Original	114.461
DASFlow com 1 nodo	147.317
Diferença	**-32.856 (22,7%)**

Observa-se que na execução da consulta sem filtro sobre apenas um nodo, a diferença entre a consulta original e da arquitetura distribuída diminui consideravelmente. Isto ocorre porque o nodo 1 envia apenas o resultado final para o nodo cliente, diminuindo desta forma o tempo de envio e evitando um processamento adicional dos resultados parciais. O resultado final é a estrutura de dados que deve ser impressa na tela e que contém

a saída esperada da consulta. O maior tempo de resposta da consulta distribuída está relacionado ao custo de operação do sistema de arquivos distribuído, que corresponde a 20% do acréscimo. Adicionalmente, tem-se o tempo de envio que soma aproximadamente 2% do custo.

Os resultados apresentados demonstram que consultas gerenciais, geralmente realizadas sobre dados antigos e sem o parâmetro filtro, devem ser controladas para que apenas os nodos que possuam estes dados a processem. Em um cenário real no entanto, são as consultas investigativas que demandam mais agilidade no processamento, e estas por sua vez, utilizam na maioria dos casos algum tipo de filtro. Isto permite observar que a solução proposta pode ser utilizada de forma eficaz para fornecer maior capacidade de processamento para consultas investigativas, além da escalabilidade de armazenamento.

7. Conclusão e Trabalhos Futuros

Neste artigo foi apresentada uma arquitetura distribuída para fornecer escalabilidade de processamento, armazenamento e coleta para dados de monitoramento de rede. Esta arquitetura é projetada e aplicada para a ferramenta NfSen. Ela é concretizada pela implementação dos módulos StoreDAS e QueryDAS. Os módulos integram nodos remotos extraindo seus recursos de processamento e armazenamento fazendo-os trabalhar de forma cooperativa.

A avaliação da solução traz dois aspectos principais. Os resultados indicam que as consultas sem filtro (SF) devem ser controladas para que sejam executadas apenas pelos nodos servidores que possuem os dados envolvidos nas consultas. Para isto é necessário obter controle de quais nodos estão aptos a respondê-la. Neste sentido, o sistema de arquivos distribuído Ceph mostrou-se uma escolha interessante por possuir o recurso de controle de localidade.

O outro aspecto está relacionado as consultas com filtros (CF). Os resultados obtidos nos testes mostraram a viabilidade imediata da solução para o processamento de consultas com caráter investigativo, ou seja, aquelas que buscam informações relacionadas a incidentes de segurança. Ainda que o ganho não tenha sido expressivo, é um resultado que demonstra o potencial da ferramenta visto que os experimentos consideraram a paralelização com apenas 2 servidores.

Para os trabalhos futuros, são planejados estudos experimentais considerando a existência de mais nodos de processamento a fim de determinar seu impacto no tempo de resposta das consultas. É proposta também a implementação da módulo de metadados com interface para a integração com o módulo QueryDAS-cliente. Desta forma, o módulo será capaz de decidir dinamicamente qual nodo responderá as consultas.

Para a etapa de coleta, a interface com o módulo metadados irá permitir o agrupamento dinâmico dos dados coletados de acordo com a importância temporal. À medida em que novos dados são coletados, os antigos serão movidos e armazenados propositadamente somente em alguns nodos, evitando réplica total e garantindo o aumento na capacidade de armazenamento. Este recurso será provido pela implementação conjunta do módulo de metadados com os módulos StoreDAS-cliente e StoreDAS-servidor.

Adicionalmente, esta solução pode ser utilizada para integrar diversas redes permitindo que o NfSen tenha acesso a um conjunto maior de dados de monitoramento. Para

isto, os módulos cliente e servidor somados ao metadados, atuarão para integrar a coleta e armazenamento de todos os nodos de tal forma que os fluxos IP estarão disponíveis para qualquer consulta, ou seja, sobre dados recentes ou antigos. Desta forma, a partir de uma única consulta da ferramenta NfSen poderá ser obtida uma visão geral de todas as redes monitoradas, não sendo necessárias várias consultas para contemplar todos os dados coletados.

Referências

Brownlee, N., Mills, C., and Ruth, G. (1999). Traffic flow measurement: Architecture. `http://tools.ietf.org/html/rfc2722`. Acessado em junho de 2014.

Cisco, W. P. (2013). Cisco visual networking index: Forecast and methodology, 2012-2017.

Claise, B. (2004). Cisco systems netflow services export version 9. rfc 3954 (informational).

Claise, B. (2013). Ipfix protocol. `http://tools.ietf.org/pdf/rfc7011.pdf`.

Deri, L. (2001). Ntop software. `http://www.ntop.org/products/ntop/`. Acessado em março de 2014.

Deri, L., Cardigliano, A., and Fusco, F. (2013). 10 gbit line rate packet-to-disk using n2disk. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 441–446. IEEE.

Deri, L. and Fusco, F. (2013). Microcloud-based network traffic monitoring. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 864–867. IEEE.

Fullmer, M. (2000). Flow-tools software. `http://www.splintered.net/sw/flow-tools/docs/flow-tools.html`. Acessado em março de 2014.

Gao, L., Yang, J., Zhang, H., Zhang, B., and Qin, D. (2011). Flowinfra: a fault-resilient scalable infrastructure for network-wide flow level measurement. In *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*, pages 1–8. IEEE.

Haag, P. (2009). `http://nfsen.sourceforge.net`. Acessado em março de 2014.

Hadoop (2005). The hadoop distributed file system. `http://aosabook.org/en/hdfs.html`. Acessado em maio de 2014.

Lee, Y., Kang, W., and Son, H. (2010). An internet traffic analysis method with mapreduce. In *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pages 357–361. IEEE.

Lee, Y. and Lee, Y. (2013). Toward scalable internet traffic measurement and analysis with hadoop. *ACM SIGCOMM Computer Communication Review*, 43(1):5–13.

Li, J., Ding, S., Xu, M., Han, F., Guan, X., and Chen, Z. (2011). Tifa: Enabling real-time querying and storage of massive stream data. In *Networking and Distributed Computing (ICNDC), 2011 Second International Conference on*, pages 61–64. IEEE.

Morariu, C., Kramis, T., and Stiller, B. (2008). Dipstorage: Distributed architecture for storage of ip flow records. In *Proc. of the 16th Workshop on Local and Metropolitan Area Networks*.

Morariu, C., Racz, P., and Stiller, B. (2010). Script: a framework for scalable real-time ip flow record analysis. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 278–285. IEEE.

Morariu, C. and Stiller, B. (2008). Dicap: Distributed packet capturing architecture for high-speed network links. In *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, pages 168–175. IEEE.

sFlow Protocol (2004). `http://sflow.org/sflow_version_5.txt`.

TCPDUMP/LIBPCAP (1999). `http://www.tcpdump.org`. Acessado em março de 2014.

Weil, S. A., Brandt, S. A., Miller, E. L., and Maltzahn, C. (2006). Crush: Controlled, scalable, decentralized placement of replicated data. In *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, page 122. ACM.

Weil, S. A., Leung, A. W., Brandt, S. A., and Maltzahn, C. (2007). Rados: a scalable, reliable storage service for petabyte-scale storage clusters. In *Proceedings of the 2nd international workshop on Petascale data storage: held in conjunction with Supercomputing'07*, pages 35–44. ACM.

Wireshark (2006). `http://www.wireshark.org/docs/`. Acessado em março de 2014.