

piPEs-BR: Uma arquitetura para a medição de desempenho em redes IP

Leobino Sampaio¹, Ivo K. Koga¹, Herbert Souza¹, Ivan K. Koga¹, Guilherme E. Rhoden², Fausto Vetter², Gerson Leiria², José A. S. Monteiro¹, Edison Melo²

¹Núcleo Interdepartamental de Pesquisas em Redes de Computadores – NUPERC
Universidade Salvador (UNIFACS)
Rua Ponciano de Oliveira, 126 - Rio Vermelho 41950-275 – Salvador – BA – Brazil
leobino@unifacs.br, ivo.koga@gmail.com, herbert_monteiro@yahoo.com.br,
ivankoga@gmail.com, suruagy@unifacs.br

²Laboratório de Interoperabilidade em Redes de Computadores – INTEROP
Universidade Federal de Santa Catarina (UFSC)
Campus Universitário, CEP 88010-970 - Florianópolis-SC Brasil
{rhoden, fvetter, leiria, melo}@npd.ufsc.br

Abstract. *It is constantly increasing the demand for performance measurement infrastructures where measured data can be used for a wide range of tools and, additionally, being able to cover different network administrative domains. Nowadays, there is a strong trend in the adoption of Web Services for deploying flexible and adaptable network environments. However, many questions remain open when trying to deploy monitoring environments. These questions are primarily related to test, storage, scheduling, authorization, interface, and detection/advising. This paper presents an End-to-End performance measurement architecture, named piPEs-BR, which deals with these questions and which was implemented and tested at RNP's backbone.*

Resumo. *É cada vez maior a demanda por infra-estruturas de medição de desempenho em que os dados medidos possam ser utilizados por um amplo conjunto de ferramentas, podendo, inclusive, pertencer a domínios administrativos de redes diferentes. Atualmente, existe uma forte tendência para a adoção de Serviços Web como forma de implantação de ambientes de medições flexíveis e adaptáveis. Contudo, muitas questões permanecem em aberto quando se deseja implantar ambientes de monitoramento desta natureza, principalmente as relacionadas ao teste, armazenamento, agendamento, autorização, interface e detecção/aconselhamento. Assim, este trabalho apresenta uma arquitetura para a medição de desempenho fim-a-fim denominada piPEs-BR, que trata destas questões e foi implementada e testada no backbone da RNP.*

1. Introdução

Nos últimos anos, o número e a complexidade das aplicações da Internet tiveram um aumento considerável. Tal fato fez com que as redes IP, projetadas inicialmente para serviços simples, passassem a prover serviços para aplicações com requisitos de desempenho bem mais específicos e difíceis de serem atendidos. Mesmo em redes de alta velocidade como a Abilene (Internet2) [Internet2 2005] e a Géant2 [Géant2 2005b], os

usuários não estão livres de problemas de desempenho que podem ser gerados por problemas na rede local, problemas de roteamento e de interconexão entre as redes. Esta constatação tem levado ao desenvolvimento de iniciativas visando identificar problemas de desempenho fim-a-fim que possam ser úteis para os usuários destas redes. Dentre estas devem ser mencionadas a iniciativa de desempenho fim-a-fim (E2Epi — *End to End performance initiative*) da Internet2 [Internet2 2004] e a atividade de pesquisa em Medição e Monitoração de Desempenho (JRA1) da Géant2 [Géant2 2005c], que buscam utilizar e desenvolver mecanismos de averiguação e acompanhamento que auxiliem no diagnóstico mais preciso dos problemas, bem como na sua caracterização.

A monitoração de redes, destas e de outras iniciativas, envolve o uso de ferramentas de medição que geralmente são utilizadas de acordo com a plataforma de *hardware* e *software* disponível, o nível de detalhamento das medições (granularidade) desejada e as métricas de desempenho de interesse, o que, de certa forma, resulta na implantação de ambientes caracterizados pela diversidade de ferramentas e técnicas escolhidas de acordo com a conveniência dos gerentes de rede. Como exemplo, pode-se citar o grupo da Internet2 que desenvolveu inicialmente o ambiente piPEs (*performance initiative Performance Environment system*) [Internet2 2003] que utiliza diversas ferramentas para coletar e disponibilizar dados de medições.

Acontece que ambientes que utilizam ferramentas diversificadas, como o piPEs, se caracterizam pela dificuldade de integração de novas ferramentas, novas técnicas e novos serviços. Esta dificuldade se torna ainda mais evidente quando os ambientes de monitoramento ultrapassam as barreiras dos seus domínios administrativos. Por este motivo, existe uma carência por infra-estruturas de monitoramento que sejam flexíveis e adaptáveis às aplicações de usuários finais bem como às ferramentas que fazem uso dos dados das medições, tais como as ferramentas de gerenciamento, visualização, etc. Estas razões levam a uma forte tendência para a utilização de Serviços Web (do inglês *Web Services*) [Roy and Ramanujan 2001], como forma de reduzir os problemas de interoperabilidade das aplicações, fazendo com que cada ferramenta exponha as suas funcionalidades de negócios de forma bem definida e em forma de serviços, utilizando a Web como o principal meio de comunicação.

A perspectiva de utilização de Serviços Web para a integração das ferramentas de medição motivou a realização de diversos trabalhos na área. Sampaio e Monteiro [Sampaio and Monteiro 2004] propuseram um modelo de medições por fluxo de tráfego orientado a serviços em que foram definidos e implementados serviços de manipulação dos fluxos de aplicações em redes IP. A Internet2 e a Géant2 iniciaram um trabalho conjunto no sentido de definir um ambiente comum de medição que pudesse ser implantado nas NRENs (*National Research and Education Networks*) associadas. Dentro deste esforço, foi elaborado um documento, chamado de GFD (*General Framework Design*) [Géant2 2005a], detalhando uma arquitetura orientada a serviços (SOA — *Service Oriented Architecture*) [Papazoglou 2003] para uma infra-estrutura de medições de desempenho em redes IP. Atualmente, estão desenvolvendo um protótipo batizado de perfSONAR (*PERformance Service Oriented Network monitoring ARchitecture*) [Hanemann et al. 2005] com o objetivo de testar, de forma concreta, os serviços definidos no GFD e que servirá de base para a infra-estrutura a ser implantada nas NRENs.

Mesmo que os trabalhos mencionados tenham dado contribuições significativas

em prol de um ambiente de monitoramento flexível, ainda existe uma carência muito grande por uma infra-estrutura completa envolvendo funcionalidades relacionadas ao teste, armazenamento, agendamento, autorização, interface e detecção/aconselhamento. Neste contexto, a RNP, através do grupo de trabalho de medições (GT-Medições) [Monteiro 2004], especificou, desenvolveu e implantou uma arquitetura para a medição de desempenho orientada a serviços, denominada piPEs-BR, seguindo os princípios do piPEs e algumas das especificações do GFD. Na arquitetura piPEs-BR toda a infra-estrutura responsável por obter as medidas de desempenho é vista como um único sistema, dando total transparência aos detalhes específicos da técnica e ferramenta de medição e da rede. Além disso, a arquitetura foi concebida para que os resultados das medições realizadas possam ser utilizados por outras infra-estruturas de medição.

Portanto, este trabalho se propõe a apresentar a arquitetura piPEs-BR e a sua implementação através de um ambiente de medição que leva o mesmo nome. Este ambiente foi desenvolvido e testado no *backbone* da RNP durante os experimentos do GT-Medições. O ambiente e os resultados alcançados validaram a arquitetura concebida e forneceram um referencial técnico para trabalhos futuros.

Este artigo está organizado da seguinte maneira: na Seção 2, a arquitetura piPEs-BR é apresentada, onde são descritos e detalhados os seus módulos de funcionamento. Na Seção 3, são abordados os principais aspectos de implementação do ambiente piPEs-BR. Na Seção 4, são descritos os experimentos realizados. Por fim, a Seção 5 é dedicada à conclusão e aos trabalhos futuros.

2. A arquitetura piPEs-BR

A arquitetura piPEs-BR é composta por um conjunto de módulos que visam contemplar funcionalidades de: teste, armazenamento, agendamento, autorização, interface e detecção/aconselhamento. Estes módulos interagem entre si através do uso de Serviços Web, fornecendo e solicitando serviços com funcionalidades específicas e bem definidas. Além dos módulos, estão previstas na arquitetura interações com três tipos de usuários: usuários finais, avançados e administradores do ambiente. A Figura 1 apresenta a arquitetura, seus módulos e usuários.

Um dos principais objetivos da arquitetura piPEs-BR é que o usuário final, interessado nas informações sobre a rede, não necessite lidar com detalhes específicos das ferramentas utilizadas nas medições. Ao receber a solicitação de uma determinada informação sobre a rede, o próprio ambiente de medições deve ser capaz de recuperar as informações que o usuário necessita e disponibilizá-las num formato que pode variar de acordo com o tipo de usuário. Outra característica da arquitetura proposta é que o ambiente de medições implantado deve ser inteligente o suficiente para gerenciar seus testes, fazendo o agendamento de novos testes de forma a minimizar a ocorrência de testes repetidos, minimizando a utilização da rede e evitando que esses venham a prejudicar seus próprios resultados.

Além disso, a arquitetura piPEs-BR visa definir um sistema que possibilite a sua expansão estando preparado para incorporar novas funcionalidades sob a demanda de seus usuários, através da integração de novas ferramentas. Assim, visa definir um sistema que contém em seu núcleo básico os processos de coleta de dados de medições, podendo ser integradas outras ferramentas para, por exemplo, processar os dados coletados voltados a uma aplicação específica.

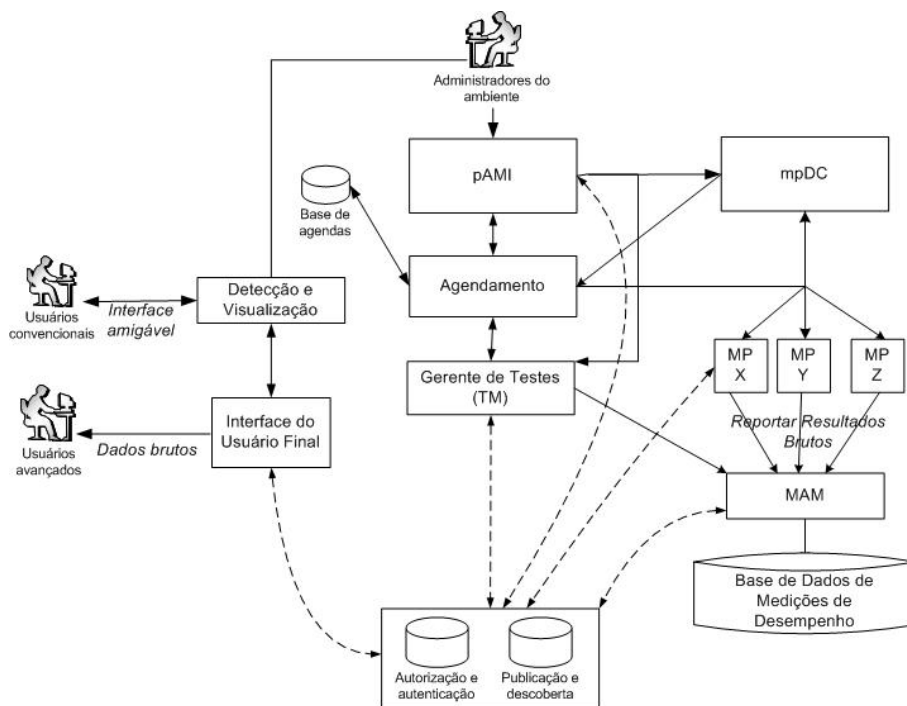


Figura 1. Arquitetura piPEs-BR.

2.1. Módulos da arquitetura

A arquitetura piPEs-BR é composta por módulos especializados de software. A seguir é descrito cada um desses módulos.

MP (*Measurement Point – Ponto de medição*): os pontos de medição são os módulos responsáveis pela execução de uma ou mais ferramentas de medição. Num MP está presente o módulo de teste, que se encarregará de receber as agendas dos testes para aquele MP, realizar os testes solicitados e por fim armazenar localmente os resultados de uma medição. O serviço do MP é o de coletar os dados das medições (ativas e/ou passivas). Por este motivo, o MP é uma peça fundamental no funcionamento de todo o ambiente. Em intervalos de tempo pré-estabelecidos, o MP envia os dados armazenados localmente ao MAM (descrito a seguir) que se encarrega de armazenar os dados por um período maior de tempo. Para isso o MP possui um módulo interno denominado mpMC (Consolidador de Medidas do MP).

mpDC (*Measurement Point Dynamic Configurator – Configurator Dinâmico dos Pontos de Medição*): é o módulo responsável por receber o registro de um novo ponto de medição e suas capacidades de medição. É através dele que um ponto de medição se anuncia para o ambiente piPEs-BR podendo então receber agendamentos para a realização de testes com as ferramentas que possui. Nele está o registro de todos os pontos de medição e suas capacidades que são consultadas antes da realização de um agendamento de testes.

pAMI (*piPEs-BR Administration and Management Interface – Interface de gerenciamento e administração*): é responsável por fornecer toda a interface necessária para a realização das atividades dos administradores do ambiente. É através do pAMI que os administradores podem realizar agendamentos e configurar os MPs.

MAM (*Measurement Archive Manager* – Gerente do Arquivo de Medições): este módulo é um gerenciador de banco de dados (SGDB) com as funções de acesso aos dados e disponibilização através de Serviços Web. Os pontos de medição podem armazenar os resultados das medições localmente e, de tempos em tempos, enviá-los para o MAM de forma a disponibilizar uma série histórica dos dados das medições.

Publicação e Descoberta (LS – *Lookup Service*): este módulo é o responsável por manter uma base de dados com informações sobre todos os serviços disponibilizados pelo ambiente, de forma que tais serviços possam ser descobertos e utilizados pelas ferramentas usuárias de forma dinâmica.

Autorização e Autenticação (AA): os serviços de autenticação e de autorização envolvem a criação de mecanismos de controle de acesso aos demais serviços disponibilizados com base nas credenciais apresentadas pelos usuários do ambiente.

Gerente de testes (TM – *Test Manager*): é responsável pelo controle no uso dos recursos da arquitetura. Este toma a decisão de quais testes deverão ser realizados e/ou agendados conforme as informações presentes no MAM e nas solicitações realizadas pelas aplicações usuárias. Este componente faz o controle de acesso aos recursos disponibilizados nos MPs e no MAM.

Agendamento: faz o agendamento de testes conforme as solicitações do gerente ou da interface dos administradores. Além disso, é o módulo que faz a correlação entre as agendas e controla o disparo dos testes.

Interface do usuário final: módulo que fornece a interface com o usuário final, provido as informações das medições através do uso de Serviços Web. Ele apresenta os dados coletados através de um formato padrão (ex. XML) que pode ser utilizado pelos aplicativos pertencentes ao módulo de detecção e visualização ou pelo usuário avançado que pode escrever a sua própria aplicação.

Detecção e visualização: é o módulo composto por ferramentas para acesso e visualização dos dados das medições. Podem ser utilizadas ferramentas já existentes tal como a MonALISA [Legrand et al. 2004].

2.2. Usuários

A arquitetura piPEs-BR especifica um ambiente de medições que visa atender as necessidades de diversos usuários, que podem estar localizados em domínios administrativos diferentes. Por isso, optou-se em classificar os usuários basicamente em três tipos principais: usuários convencionais, usuários avançados e administradores do ambiente.

Os usuários convencionais (ou leigos) são aqueles que interagem com o ambiente para ter acesso aos resultados das medições de forma padronizada e mais amigável. Esse acesso se dá tipicamente através de páginas Web pré-definidas disponibilizadas pelo ambiente para tais usuários. Os usuários convencionais também podem requisitar medições através do Gerente de testes, que por sua vez entra em contato com o módulo de Agendamento para realizar a operação desejada.

Usuários avançados, por outro lado, são caracterizados por não apenas estarem interessados na visualização gráfica dos resultados provenientes das medições, mas também, por estarem interessados nos dados brutos de medição coletados pelo ambiente. Nesse caso, os usuários avançados têm acesso aos dados brutos para que eles próprios possam realizar suas análises específicas.

Por fim, os administradores do ambiente são usuários especializados cuja função é manter a infra-estrutura (software e hardware) de medições operante, bem como manter os processos de medição a serem executados sobre a infra-estrutura sendo executados de forma adequada para o fornecimento de serviços aos demais usuários.

2.3. Interação entre os módulos da arquitetura

Sobre a interação entre os módulos da arquitetura, considera-se inicialmente um usuário leigo que esteja interessado em conhecer o estado da rede e/ou visualizar gráficos de desempenho da mesma. Para isto ele deve possuir uma ferramenta de detecção/visualização que fará acesso ao ambiente e este poderá solicitar uma autenticação. Após a autenticação e recebendo a autorização adequada ao seu tipo de usuário, a ferramenta deverá fazer então um acesso ao módulo de Publicação e Descoberta para descobrir o serviço que irá fornecer os dados solicitados pelo usuário. Através desta consulta, ela descobrirá que deverá fazer acesso ao Gerente de Testes, para obter os dados desejados. Os dados desejados poderão já estar disponíveis na base de dados (MAM) ou poderá ser necessário agendar testes para obtê-los, fazendo uso do módulo de Agendamento. No primeiro caso, basta acessar o MAM e recuperar as medidas que estão guardadas na base de dados de medições. Já no segundo caso, é necessário possuir autorização necessária para realizar o agendamento solicitando ao módulo de Agendamento para que este insira uma nova agenda na base de agendas para os MPs envolvidos. Quando os dados solicitados estiverem disponíveis, eles serão enviados em forma bruta do MAM para o Gerente de testes. Este, por sua vez, fará um processamento nos dados e depois os enviará para a ferramenta de detecção/visualização, gerando o gráfico que será apresentado ao usuário.

A requisição dos dados feita pelo usuário avançado ocorre de forma semelhante, diferindo apenas no aspecto de que o usuário avançado pode fazer uma chamada direta ao Gerente de testes e obter os dados diretamente para realizar as suas análises específicas. Para isso, o Gerente de testes utiliza os serviços dos módulos de Autenticação e Autorização e Publicação e Descoberta para verificar a conformidade da informação de acesso dos usuários e também para descobrir e escolher os serviços que podem fornecer dados e informações úteis para estes tipos de usuários.

O acesso de um administrador se dá através do pAMI. Após a autenticação e definição do seu nível de acesso e com o conhecimento dos serviços disponíveis, o pAMI poderá se conectar diretamente ao Gerente de testes solicitando que novos testes sejam realizados ou acessar diretamente o módulo de Agendamento podendo gerenciar diretamente as agendas.

3. Implementação do ambiente piPEs-BR

No intuito de avaliar a viabilidade de implantação de ambientes de medições em conformidade com a especificação da arquitetura piPEs-BR, foi desenvolvido um protótipo que fez parte do escopo das atividades do GT-Medições da RNP. Nesta seção serão apresentados os detalhes de implementação deste protótipo descrevendo, inicialmente, os módulos de teste (MP, mpDC e pAMI). Depois serão apresentados os módulos de Agendamento, Arquivo, Visualização de dados. E, por fim, o de Publicação e Descoberta.

3.1. Módulo de Testes (MP, mpDC e pAMI)

Uma vez que, normalmente, as ferramentas de medição são invocadas passando-se um conjunto de parâmetros e retornam uma seqüência de resultados, pode-se abstrair qual ferramenta específica está sendo utilizada e considerar apenas uma ferramenta genérica.

Sendo assim, de acordo com os objetivos e especificações da arquitetura, os pré-requisitos para o desenvolvimento do módulo de testes foram os seguintes:

Independência da ferramenta de medição de desempenho: o módulo deve ter alta extensibilidade referente ao acoplamento de novas ferramentas de medição, através da utilização de interfaces genéricas para a comunicação entre as variadas ferramentas e a infra-estrutura piPEs-BR. Para isso foram utilizados arquivos de configuração para cada ferramenta. Nestes arquivos foram definidos os parâmetros necessários e os tipos de resultados que a ferramenta retorna ao executar um teste.

Facilidade de integração: a facilidade de integração com novas ferramentas visa diminuir os problemas decorrentes das constantes atualizações. Portanto, a solução mais viável e adotada foi a utilização de arquivos de configuração para as ferramentas e a criação de *scripts* para a coleta de resultados.

Agenda interna: a agenda interna tem por objetivo evitar que testes não deixem de ser realizados por causa de falhas de comunicação na rede. Esta agenda é responsável pelo controle da execução dos testes em cada ponto de medição.

Armazenamento local dos resultados dos testes: é recomendado que o armazenamento dos dados das medições em um MP seja realizado localmente, uma vez que ao centralizar os resultados dos testes, existe o risco de se perder informações caso a base centralizada esteja indisponível logo após a execução de um dado teste.

Com base nesses pré-requisitos, o protótipo foi desenvolvido conforme a Figura 2, a qual ilustra os módulos e suas interações. Os serviços disponibilizados por estes módulos são incumbidos da realização de testes e do controle do estado de execução dos componentes. Observa-se também na Figura 2 que a comunicação entre os componentes desta camada se dá através de Serviços Web, representada através das caixas com a sigla WS (abreviação da palavra em inglês *Web Service*).

O módulo MP desenvolvido é o responsável pela execução dos testes, ou seja, provê a interface do ambiente para as ferramentas de medições (Owamp, iperf, ping e ping6). Cada MP é previamente registrado no mpDC (Configurador Dinâmico de MPs) para que possa ser conhecido pelo ambiente, tanto em aspectos de capacidades como de infra-estrutura, e para vir a ser utilizado pelos outros módulos. Para a realização dos testes, deve ser utilizado um módulo externo de agendamento, que deve informar os testes a serem executados e os períodos de execução. Os dados coletados pelos MPs são enviados ao MAM e armazenados em uma base de dados. Para a comunicação com o MAM, os MPs utilizam um componente interno chamado mpMC.

Observa-se também na Figura 2 que cada módulo, de modo geral, é composto por diversos componentes internos que provêm funcionalidades para que cada atividade seja realizada. Além disso, pelo fato da comunicação entre os módulos ser realizada através de Serviços Web, o ambiente pode ser facilmente integrado a outras infra-estruturas de

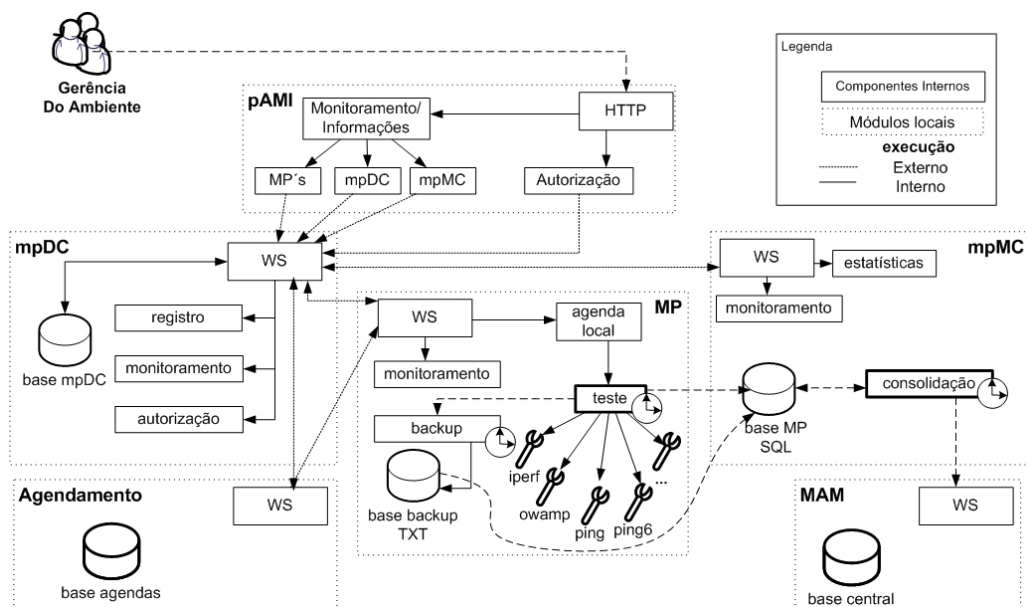


Figura 2. Visão detalhada da implementação dos módulos de teste.

medição. Como exemplo, pode-se citar o módulo MP que disponibiliza serviços para agendamento de testes, autorização dos pontos e medição. Estes serviços (alguns listados na Tabela 1) podem ser utilizados por qualquer outra ferramenta através de Serviços Web.

Tabela 1. Serviços Web do MP para o Agendamento.

Serviço	Parâmetros de entrada
new_schedule	schedule_id, minute, hour, day, month, weekday, period, exclusion
new_test	test_id, name, description, destination_id, timeout, wait_time, tool_id, parameters
add_test_to_schedule	test_id, schedule_id
delete_schedule	schedule_id
delete_test	test_id
delete_test_from_schedule	test_id, schedule_id

3.2. Agendamento

No contexto do piPES-BR, o módulo de Agendamento garante que os testes sejam executados de acordo com o que for solicitado pelo módulo Gerente de testes ou os usuários administradores do ambiente analisando também a disponibilidade de recursos na rede para a realização dos mesmos. Através de informações disponibilizadas pelo mpDC, o módulo de agendamento possui o conhecimento das ferramentas que estão disponíveis no domínio de medição.

As solicitações externas passam por um controlador de recursos denominado *Resource Control Manager* (RCM), no qual verifica a possibilidade de agendamento do teste requisitado de acordo com os recursos disponíveis em conjunto com as agendas marcadas. Após passar pela análise dos recursos, o teste também é submetido a uma análise das agendas, evitando assim que testes redundantes sejam alocados.

A verificação das agendas, realizada através do *Schedule Control Manager* (SCM), se faz necessária a fim de reduzir o número de testes de um ponto de medição.

Isso evita a sobrecarga de processamento nos pontos de medição e da rede, evitando a influência dos mesmos nos resultados das medições.

Uma vez que os testes tiverem passado pelo controlador de recursos e pelo controlador de agendas, eles estão prontos para serem criados pelo agendador nos pontos de medição.

3.3. Gerente do Arquivo de Medições (MAM)

As informações recuperadas pelos MPs são enviadas ao Gerente do Arquivo de Medições (MAM) que fornece serviços para o registro de dados históricos das medições realizadas. Este módulo possui os serviços de inserção e recuperação de dados armazenados em um SGBD.

Para a implementação dos serviços do MAM utilizou-se o servidor de aplicações Tomcat [Tomcat 2005] juntamente com o Apache Axis [Axis 2005]. O SGBD escolhido para armazenamento dos dados foi o MySQL [MySQL 2005]. A escolha do Tomcat e o Apache Axis deveu-se à facilidade e à quantidade de ferramentas disponíveis nestas plataformas voltadas para o desenvolvimento de Serviços Web. Já o MySQL foi escolhido a partir de estudos comparativos com o ROOT [Rademakers and Brun 1998] que é um *framework* de análise de dados orientado a objetos que permite a inserção e recuperação de dados. O *framework* ROOT se mostrou um pouco mais veloz para acesso aos dados, porém a sua instalação e criação de estruturas de acesso ao ambiente se tornaram uma restrição quanto à flexibilidade e dificuldade perante as ferramentas que foram utilizadas. Assim, a partir deste teste, o MySQL provou ser a melhor opção como base de dados por fornecer uma interface flexível e amigável, além de possuir diversas ferramentas para a sua manipulação e conexão com o ambiente escolhido.

Desta maneira, um cliente/serviço que fizer uma chamada para um serviço do MAM terá que utilizar uma mensagem SOAP encapsulada num pacote HTTP com a requisição. Neste caso, o Tomcat processará o HTTP e desencapsulará a mensagem SOAP para que o Axis faça o tratamento do conteúdo e chame o serviço adequado.

Para atender a esta requisição é necessário o uso dos objetos das camadas inferiores para a recuperação das medidas disponíveis no MySQL e, após a finalização desta tarefa, os objetos retornam os dados para o Axis que se encarrega de retorná-los para o cliente, encapsulando-os em uma mensagem SOAP que será enviada via HTTP pelo Tomcat.

Os serviços desenvolvidos para o MAM realizam a inserção dos dados de testes provenientes dos MPs do piPEs-BR e disponibilização dos dados brutos e de média para acesso externo. Pode-se verificar como cada componente do MAM se relaciona através da Figura 3.

Os serviços disponibilizados para recuperação de dados são o `rawData` e o `avgData`. O primeiro retorna os dados brutos entre dois pontos de medição e o segundo retorna a média dos dados em um determinado intervalo de tempo. Já a inserção de dados pode ser feita linha a linha (inserção simples) utilizando o serviço `dataInsert`, ou em massa, utilizando uma matriz de dados, com o serviço `dataArrayInsert`.

Nas Tabelas 2 e 3 estão apresentados, respectivamente, os parâmetros que são passados para cada serviço (de recuperação e inserção) e o retorno esperado.

gráficos para a visualização de dados medidos no ambiente piPEs-BR. A MonALISA contém uma visualização 3D que utiliza pontos com coordenadas geográficas que são considerados os fornecedores das informações. Essa forma de disposição das informações tem algo em comum com a proposta do piPEs-BR que disponibiliza informações também em pontos de medição (os MPs).

A ferramenta MonALISA foi utilizada como parte do módulo de Detecção e Visualização do piPEs-BR. A Figura 4 ilustra a integração da MonALISA com o piPEs-BR, onde o módulo criado para o *MonALISA Service* utiliza os Serviços Web do piPEs-BR e disponibiliza os dados para o *MonALISA Client* que, por sua vez, gera os gráficos.

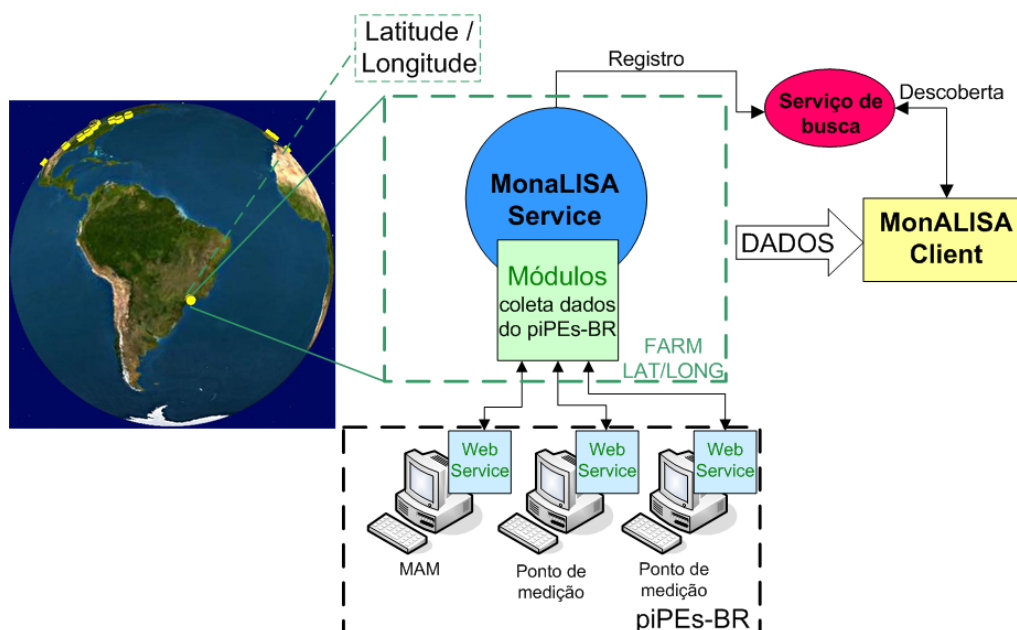


Figura 4. Integração da MonALISA com o piPEs-BR e o usuário final.

Para adicionar essa nova funcionalidade à MonALISA, foi construído um módulo composto por classes Java, herdeiras da classe `lia.Monitor.monitor.cmdExec` que implementam a interface `lia.Monitor.monitor.MonitoringModule`. Essas classes têm como objetivo recuperar os dados do piPEs-BR, tratá-los e formatá-los de acordo com as necessidade da ferramenta MonALISA.

3.5. Publicação e Descoberta

Devido ao piPEs-BR disponibilizar serviços de forma distribuída, e também por ser dividido em diversos componentes, foi necessário o desenvolvimento de um componente responsável pela localização dos serviços disponibilizados pelos módulos do piPEs-BR. De forma dinâmica, os consumidores de serviços do ambiente necessitam de informações sobre o ponto de acesso do serviço, parâmetros de entrada e saída, a localização e outras informações mais específicas.

Como os módulos do ambiente utilizam Serviços Web para comunicação, optou-se por utilizar uma tecnologia consolidada para publicação e descoberta de Serviços Web. Além disso, foi necessário construir bibliotecas que facilitassem o desenvolvimento das aplicações, fazendo com que as aplicações pudessem ser desenvolvidas ou adaptadas de

forma mais rápida. Sendo assim, a tecnologia usada para o módulo de Publicação e Descoberta foi a ferramenta UDDI (*Universal Description, Discovery, and Integration*) [Sleeper 2002], dado que em uma análise prévia, verificou-se que esta tecnologia possui ampla aceitação e é bastante utilizada nestes tipos de soluções. O UDDI consiste numa plataforma de suporte à publicação e descoberta de recursos, com o objetivo de fomentar a interoperabilidade e o uso de Serviços Web.

A utilização do UDDI é de extrema importância para a integração do piPEs-BR com outras infra-estruturas de medição (tais como o perfSONAR), já que as informações sobre os serviços disponibilizados serão atualizadas e utilizadas de forma dinâmica e pela Web.

Para o piPEs-BR foi construída uma biblioteca na linguagem Java para descobrir os serviços por nome ou por um identificador. Na criação dessa biblioteca foi utilizada a API *UDDI4J* [Hewlett-Packard and IBM 2005], que contém métodos de conexão com o UDDI. Essa biblioteca contém classes em Java que recebe como parâmetros de entrada: o nome do *business* que contém o serviço desejado (no nosso caso o piPEs-BR), o nome ou identificador do serviço e o *tModel Name*, que no escopo do piPEs-BR representa o MP desejado. As classes ainda retornam como resposta das chamadas características do serviço, tais como: URL de acesso, URL do documento WSDL, parâmetros, etc. Essa biblioteca basicamente cria um objeto do tipo `UDDIProxy` que utiliza as URL's de *inquiry* e *publish* para conectar no UDDI.

Para utilizar um serviço através da biblioteca desenvolvida, a primeira ação a ser tomada é a busca do *Business* que contém o serviço desejado. Para isso é utilizado o método `proxy.findbusiness()` que retorna um objeto com todas as informações contidas no *business* e que contém também as informações dos serviços. Depois se comparam os nomes dos serviços retornados pelo *proxy* com o nome do serviço desejado, para verificar se o serviço está contido neste *business*. Se o serviço existir no *business* procura-se o *tModel* que contém as informações de acesso ao serviço. Caso o serviço não exista é retornada uma mensagem de erro.

Através da biblioteca de publicação e descoberta o desenvolvedor não necessita saber detalhes sobre a conexão com o UDDI, o que facilita a busca e utilização dos serviços disponibilizados. Como exemplo, pode-se citar a ferramenta MonALISA que pode ser considerada como uma consumidora dos serviços do piPEs-BR, que utiliza esta biblioteca para consultar a URI referente aos serviços disponibilizados.

4. Experimentos Realizados

A principal contribuição deste trabalho é a concepção de uma arquitetura de medições de desempenho que possibilita reunir as melhores características das ferramentas de monitoração existentes. Essas contribuições ficaram mais evidentes após a avaliação do protótipo que foi desenvolvido e testado no *backbone* da RNP. Deste modo, essa Seção será dedicada a apresentar os resultados alcançados com o desenvolvimento do protótipo.

4.1. Realização de testes com ferramentas de medição

Após o desenvolvimento do MP, foram realizados diversos testes para verificar sua estabilidade e a correta integração com os outros módulos. Os testes que envolveram

a realização de 28 testes distintos de medições no intervalo de três meses foram satisfatórios. Para comprovação, foram selecionados 30 dias consecutivos no intervalo de 11/05/2005 a 10/06/2005. Estes testes demonstraram a capacidade do ambiente na realização de medições na rede e na recuperação e visualização dos dados medidos.

4.2. Integração com ferramentas de visualização

A fim de verificar a usabilidade do ambiente piPES-BR para o usuário final, foi necessária a utilização de ferramentas de visualização que possuíssem interfaces mais amigáveis. Por isso, a implementação do módulo de detecção e visualização da arquitetura do piPES-BR se deu através do uso da ferramenta MonALISA que demonstrou ser a melhor escolha para realização dos testes de visualização, já que esta possui uma interface bem avançada assim como suporte a Serviços Web.

Os dados da MonALISA são disponibilizados em cada ponto (coordenada geográfica – Figura 5(a)), que representa um MP, no ambiente piPES-BR.

Além da visualização dos dados coletados, notou-se que era necessário o desenvolvimento de uma ferramenta de visualização que fosse flexível e proporcionasse o acesso a todas as funcionalidades do ambiente piPES-br. Assim optou-se pelo desenvolvimento da ferramenta ICE (*Internet Computer network Eye*), que apesar de ainda se encontrar num estágio inicial de desenvolvimento, já apresenta os seus primeiros resultados. A ICE foi desenvolvida utilizando Java, JNI, Axis, JFreechart e os testes realizados com a ferramenta validaram o funcionamento do piPES-BR, possibilitando aos usuários a recuperação de medidas instantâneas e históricas através de sua interface.

Na Figura 5 são apresentados exemplos de utilização das ferramentas MonALISA (a) e ICE (b) no ambiente do piPES-BR.

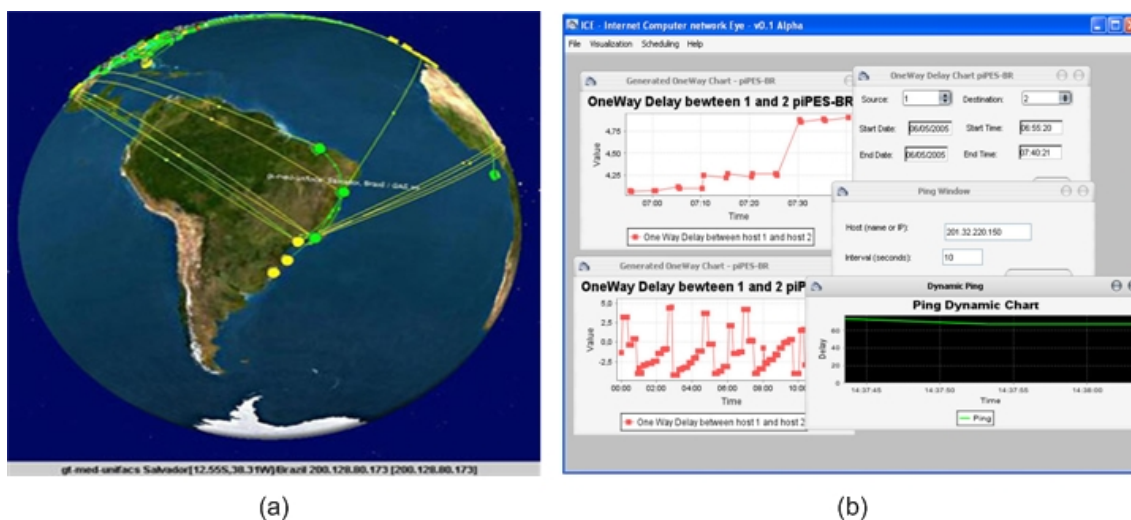


Figura 5. Ferramentas de visualização utilizadas no ambiente piPES-BR.

4.3. Gerenciamento do ambiente

Em termos de gerenciamento do ambiente, foi implementada a interface web pAMI. O objetivo foi construir uma interface para que o gerente pudesse visualizar, monitorar, acessar e modificar todas as informações referentes aos MPs, mpDC além de outras que estão

relacionadas ao ambiente piPEs-BR. Nesse sentido, podem ser encontrados no pAMI uma interface de gerenciamento completo para o ambiente em que são mostradas estatísticas de desempenho, *uptime* dos MPs e mpDC, situação atual dos MPs no ambiente entre outros, como é exemplificada pelas Figuras 6 e Figura 7. No pAMI também existe um esquema de autenticação/autorização através do qual é verificado se o usuário ou gerente tem permissão para executar uma determinada tarefa evitando que pessoas não autorizadas manipulem o ambiente.

The screenshot shows the pAMI web interface in a Mozilla Firefox browser. The page title is "pAMI piPEs BR Administration and Management Interface". The main content area displays "MPs Status Information" as a table with the following data:

ID	Name	Status	Authorized	Up-Time
1	GPS-PoP-SC	UP	YES	14 days, 21 hours, 33 minutes, 47 seconds
2	MP-UFSC	UP	YES	18 days, 17 hours, 56 minutes, 29 seconds
3	NDT-UFSC	DOWN	YES	0
4	NDT-Pop-SC	UP	YES	3 days, 0 hours, 7 minutes, 52 seconds
5	MP-RCT	DOWN	YES	0
6	MP-ROOT-PoP-SC	UP	YES	14 days, 21 hours, 41 minutes, 56 seconds
7	MP2-Bahia	UP	YES	10 days, 3 hours, 42 minutes, 58 seconds
8	MP-piPEs-BR	UP	YES	18 days, 17 hours, 56 minutes, 4 seconds
9	MP-Bahia-Owamp	SERVICES DOWN	YES	2 days, 11 hours, 6 minutes, 40 seconds
11	MP-Bahia-ProjEsp-Unifacs	DOWN	NO	0

Figura 6. Interface do pAMI que apresenta o estado dos pontos de medição.

A Figura 6 apresenta o estado dos pontos de medição contendo informações dos MPs, se existe algum problema com o ponto, se o mesmo está autorizado e o tempo que o serviço foi inicializado. O monitoramento detalhado de um ponto pode ser visualizado na Figura 7.

5. Conclusão e trabalhos futuros

As principais contribuições da arquitetura piPEs-BR podem ser avaliadas sob dois pontos de vista: a integração de diversas ferramentas de medição e o ambiente de monitoração voltado para o usuário final.

Por ser concebida para reunir as melhores características das ferramentas de medição existentes conforme a necessidade do usuário, a arquitetura piPEs-BR especifica módulos especializados que quando utilizados em conjunto, fornece um ambiente sofisticado de monitoração, abrangendo desde as ferramentas de medições já consolidadas até as ferramentas avançadas de visualização. Isso ficou comprovado através da implementação e testes com o protótipo no *backbone* da RNP. O segundo ponto a ser avaliado é que o usuário final, seja ele administrador do ambiente, sofisticado ou convencional, pode ter um ambiente de monitoração que o livra de preocupações com detalhes específicos da rede sob observação.

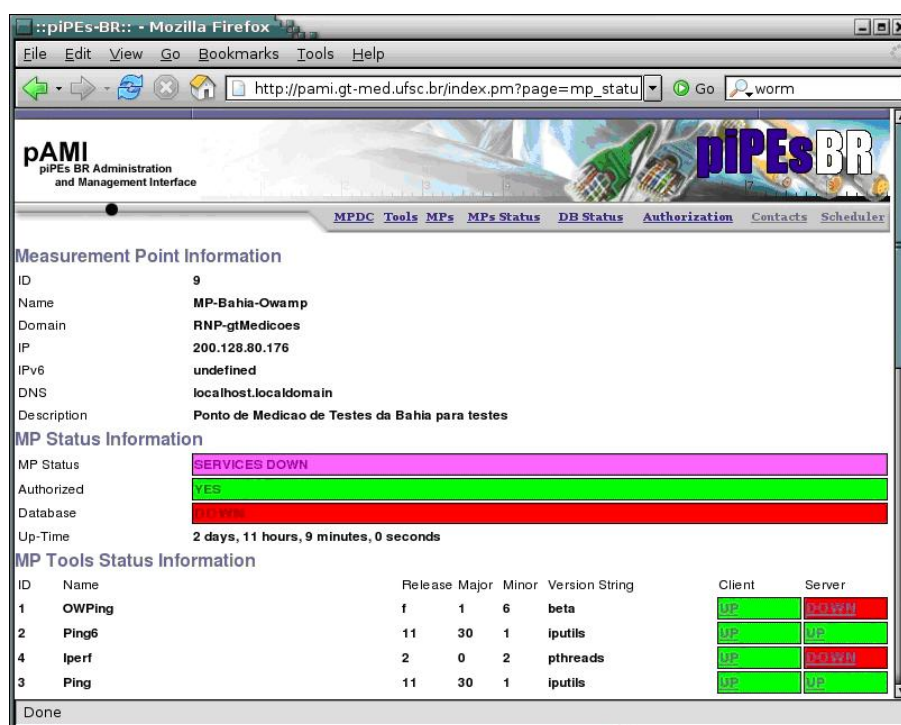


Figura 7. Interface do pAMI que apresenta os detalhes do monitoramento.

Como trabalhos futuros nesta área, vale destacar o esforço do JRA1 da Géant2 no desenvolvimento do protótipo perfSONAR que tem como meta principal prover aos grupos de usuários dados de desempenho de múltiplos domínios. Neste caso, uma contribuição significativa seria dedicar um esforço no sentido de fazer um estudo para a interoperação deste protótipo com o ambiente piPEs-BR, viabilizando testes de desempenho entre a RNP, Géant2 e Internet2.

Além da interoperação com as outras infra-estruturas de medições também existe a necessidade de implementação e aperfeiçoamento dos módulos já previstos pela arquitetura atual do piPEs-BR tais como os módulos de Agendamento, Gerente de testes, Autenticação e Autorização.

Agradecimentos

Os autores agradecem todo o apoio que a RNP vem dando ao GT-Medições com o fornecimento da infra-estrutura técnica necessária para a realização deste trabalho.

Referências

- Axis, A. (2005). Apache axis. <http://ws.apache.org/axis>.
- Chinnici, R., Gudgin, M., Moreau, J.-J., and Weerawarana, S. (2003). Web services description language (WSDL) version 1.2 part 1: Core language. <http://www.w3.org/TR/2003/WD-wsdl12-20030611>.
- Géant2 (2005a). Deliverable D.J.1.2.1: General Framework Design. <http://www.geant2.net>.
- Géant2 (2005b). Géant2. <http://www.geant2.net>.

- Géant2 (2005c). JRA1 project main page. http://monstera.man.poznan.pl/wiki/index.php/Main_Page.
- Hanemann, A., Boote, J., Boyd, E., Durand, J., Kudarimoti, L., Lapacz, R., Swany, M., Trocha, S., and Zurawski, J. (2005). Perfsonar: A service oriented architecture for multi-domain network monitoring. In *Proceedings of the Third International Conference on Service Oriented Computing (ICSOC 2005)*, pages 241–254. ACM Sigsoft and Sigweb.
- Hewlett-Packard and IBM (2005). Uddi4j. <http://uddi4j.sourceforge.net/>.
- Internet2 (2003). E2EpiPEs: End-to-End Performance Initiative Performance Environment System Architecture. <http://e2epi.internet2.edu/E2EpiPEs/e2epipe11.pdf>.
- Internet2 (2004). E2EPI:Internet2 End to End Performance Initiative. <http://e2epi.internet2.edu>.
- Internet2 (2005). Internet2. <http://www.internet2.edu>.
- Legrand, I., Newman, H., Voicu, R., Cirstoiu, C., Grigoras, C., Toarta, M., and Dobre, C. (2004). Monalisa: An agent based, dynamic service system to monitor, control and optimize grid based applications. In *Computing in High Energy and Nuclear Physics (CHEP)*, Interlaken, Switzerland. CERN.
- Microsystems, S. (2005). Jini network technology. Technical report, SUN.
- Monteiro, J. A. S. (2004). GT-Medições: Documento de Diagnóstico e Alternativas. Technical Report P2.1, RNP.
- MySQL (2005). Mysql. <http://www.mysql.com/>.
- Newman, H., Legrand, I., and Bunn, J. (2001). A distributed agent-based architecture for dynamic services. *Conference for Computing in High Energy and Nuclear Physics*. Beijing.
- Papazoglou, M. (2003). Service-oriented computing: Concepts, characteristics and directions. In *Keynote for the 4th International Conference on Web Information Systems Engineering (WISE 2003)*. IEEE Computer Society.
- Rademakers, F. and Brun, R. (1998). Root: an object-oriented data analysis framework. *Linux J.*, 1998(51es):6.
- Roy, J. and Ramanujan, A. (2001). Understanding web services. *IT Professional*, 3(6):69–73.
- Sampaio, L. and Monteiro, J. A. S. (2004). SFM3: A Service-based Flow Traffic Measurement Management Model for IP Networks. In *The 2nd Workshop on End-to-End Monitoring Techniques and Services (E2EMON)*, pages 55–61, San Diego, California.
- Sleeper, B. (2002). The evolution of uddi: Uddi.org white paper. http://www.uddi.org/pubs/the_evolution_of_uddi_20020719.pdf.
- Tomcat, A. (2005). Apache tomcat. <http://tomcat.apache.org/>.