

# SDI - Sistema de Diagnóstico Instantâneo

**Bruno César Ribas<sup>1</sup>, Diego Giovane Pasqualin<sup>1</sup>, Vinicius Kwiecien Ruoso<sup>1</sup>,  
Marcos Castilho<sup>1</sup>, Fabiano Silva<sup>1</sup>, Luis C. E. de Bona<sup>1</sup>,**

<sup>1</sup>Departamento de Informática – C3SL – Universidade Federal do Paraná  
Caixa Postal 19.081 – CEP 81.531-980 – Curitiba – PR – Brasil

{ribas, dgp06, vkr07, marcos, fabiano, bona}@inf.ufpr.br

**Abstract.** *To manage a large computers network the ability to monitor of anomalies and carry out tasks of maintenance is essential. This paper proposes an Instant Diagnose System, which allows the monitoring of large computational grid through a web interface and automatically execute corrective actions with a flexibly and customized way. The proposed tool has been implemented with significant results in Paraná Digital project network, which has more than 2100 servers.*

**Resumo.** *Para gerenciar uma grande rede de computadores a capacidade de monitoração de anomalias e de realização de tarefas de manutenção é fundamental. Este trabalho propõe um Sistema de Diagnóstico Instantâneo, que permite monitorar grandes parques computacionais através de uma interface web e executar automaticamente ações corretivas de maneira flexível e personalizada. A ferramenta proposta foi implementada com resultados expressivos na rede do projeto Paraná Digital, que conta com mais de 2100 servidores.*

## 1. Introdução

Redes de computadores estão a cada dia mais baratas e mais comuns, seja em pequenas e grandes empresas, governos, e centros acadêmicos. O custo de manutenção destas redes se torna impraticável em certo momento. O responsável deve estar atento a anomalias diversas, como falha na disponibilidade de serviços, queda de energia ou de conexão, além de outros problemas específicos a cada sistema, como alterações em hardware e software, análise de comportamento de usuários e capacidade em disco.

Saber qual o estado atual das máquinas em uma rede e tomar ações preventivas é crucial para o bom funcionamento do parque. Receber informações de cada máquina ou enviar programas para correção de problemas, em tempo real, deve ser a principal característica do sistema proposto.

O Sistema de Diagnóstico Instantâneo (SDI) provê uma interface simples para observação dos dados coletados e meios rápidos e seguros para propagar automaticamente soluções desenvolvidas pelo administrador aos problemas encontrados em sua rede, isso integrado a um modelo de autogerenciamento e autoconfiguração em grandes redes proposto por [Bona et al. 2008].

O SDI proporciona um ambiente de fácil utilização e personalização, permitindo a execução de scripts de coleta de informações e verificação com ação preventiva, exibindo os resultados através de uma interface web.

## 2. Soluções encontradas

Aqui se analisam propostas alternativas encontradas na literatura.

O *Parallel SSH* [Pssh ] permite a execução de comandos em um determinado conjunto de *hosts*<sup>1</sup> criando túneis [Ssh ] temporários entre o servidor e os *hosts*. Ferramenta similar foi desenvolvida e rejeitada por dois motivos: (1) desempenho, pois há um overhead grande para recriar os túneis a cada nova requisição; e (2) confiabilidade, devido à potencial indisponibilidade de *hosts* no instante da consulta.

Na solução proposta pelo [ClusterSSH ] os túneis se mantêm abertos entre os servidores e *hosts* todo o tempo, cada túnel conectado a um terminal X<sup>2</sup> diferente, permitindo assim que comandos sejam enviados para *hosts* específicos, ou todos ao mesmo tempo, com melhor custo e confiabilidade que a solução anterior. Essa solução foi descartada por não fornecer interface simples para visualização dos *logs*.

A solução [ClusterIt ] tem o foco no processamento de *jobs* em paralelo. Esta solução foi descartada por divergir em uso ao objetivo do nosso projeto.

## 3. Sistema de Diagnóstico Instantâneo

O diferencial do SDI quanto as aplicações citadas é o processo de organização das informações recebidas, separada em histórico e atual. A disposição de uma [API ] flexível para criação de novos *scripts*, e a visualização dos dados de modo personalizável e em tempo real, são as características em que o sistema é baseado.

A comunicação entre o servidor e os *hosts* é feita através de uma conexão *ssh* que se mantêm aberta todo o tempo, fator esse que reduz drasticamente o *overhead* do estabelecimento de conexão a cada consulta. O SDI se encarrega então de enviar os *scripts* e comandos através destes túneis até os *hosts* e posteriormente recuperar os resultados. O *ssh* foi escolhido por ser um protocolo largamente utilizado para gerenciamento remoto de máquinas unix, além de possibilitar uma comunicação constante entre duas máquinas de maneira segura e flexível.

A API do SDI fornece um conjunto de variáveis, funções e principalmente conceitos que criam um ambiente simplificado para uso do potencial da ferramenta em seus sistemas de coleta, sumarização e tratamento dos dados.

O Algoritmo 1 apresenta o fluxo de execução do SDI. É mantido um túnel *ssh* para todos os *hosts* que devem ser monitorados (linha 12). Ao criar a conexão, os *scripts* habilitados para execução na conexão são enviados (linha 13), após isso o SDI monitora dois arquivos de comandos para enviar suas modificações ao destino (linha 14). As mensagens que chegam no servidor são interpretadas pela função *parse* (linha 15), que as trata e armazena. Caso a conexão seja interrompida, o *host* em questão é colocado em estado *offline* (linha 16) e o SDI aguarda um tempo aleatório para tentar se conectar novamente (linha 17) - isso é feito para evitar um excessivo aumento de carga no servidor. Este processo é repetido até que o usuário decida parar a execução do sistema.

O funcionamento do SDI pode ser separado nos seguintes passos: coleta dos dados, tratamento das informações, sumarização e exibição em uma interface web. As

---

<sup>1</sup>Máquinas ou nós em uma rede

<sup>2</sup>Terminal de visualização e de entrada de dados para aplicações clientes do X Window System.

seções a seguir descrevem estes passos.

### 3.1. Coleta

Na fase de coleta de dados o SDI se responsabiliza por receber as informações e executar funções específicas de acordo com a configuração do sistema. O administrador é responsável por definir parâmetros e regras sobre a criação de *scripts*, ordem e periodicidade de execuções, assim como habilitação e desabilitação dos mesmos.

O túnel *ssh* aberto pelo SDI aguarda por comandos na linguagem *shell script*. Porém é possível utilizá-la para executar aplicações em qualquer outra linguagem.

A criação de uma nova rotina de diagnóstico ocorre de forma que, para cada objeto de análise, o usuário deve criar dois *scripts*. Um será executado no *host*, recuperando os dados desejados e retornando-os com um identificador, que é passado para o segundo *script* que fará a manipulação e armazenamento das informações obtidas. Para o *script* executado no servidor, a API fornece variáveis que identificam o *host* da informação sendo processada, e ajudam a definir o formato de exibição dos dados na interface web.

A organização dos *scripts* é feita de maneira que o administrador possa habilitá-los e desabilitá-los conforme necessidade. Para isso o SDI dispõe de dois diretórios: *commands-available* e *commands-enabled*. No primeiro encontram-se todos os *scripts* desenvolvidos pelo usuário ou fornecidos pelo sistema, eles podem ser habilitados e desabilitados utilizando referências (*links* simbólicos) à eles no segundo diretório. O mesmo acontece com estados e sumários, que serão abordados adiante.

A periodicidade das execuções depende da natureza da informação requerida. Informações mais dinâmicas como a carga da máquina pode ser interessante obter a cada quinze minutos, enquanto que problemas na versão de aplicações podem ser checados apenas uma vez por dia. É possível controlar a periodicidade através da estruturação de diretórios do SDI ou através da programação do *script* criado.

Para o primeiro caso existe o diretório *commands-enabled*, que possui subdiretórios que representam períodos de tempo (toda hora, todo dia, etc). Basta habilitar o *script* em um desses diretórios para assumir a periodicidade representada por ele.

É possível determinar também a ordem que os *scripts* serão executados nos *hosts* utilizando um prefixo numérico no *link* que caracteriza sua ativação. Com isso é possível definir uma biblioteca de funções que será o primeiro *script* a ser executado (000-lib, por exemplo), e será utilizada pelos demais *scripts*, pois ficará definida na memória dos *hosts*.

### 3.2. Tratamento dos Dados Recebidos

Manter conexões com muitas máquinas ao mesmo tempo pode significar uma grande quantidade de informações chegando a cada momento. Foi criada uma função dedicada a gerenciar e tratar estas informações, com desempenho satisfatório e de maneira a consumir a menor quantidade de recursos possível.

Essa função é executada em paralelo em todos os *hosts*. O número de execuções simultâneas é controlada por um semáforo, que visa amenizar a carga no servidor. Suas tarefas incluem o recebimento dos dados, passagem para o tratamento ao *script* responsável (criado pelo usuário ou fornecido pelo SDI), verificação da corretude das respostas e organização das informações geradas, salvando-as nos logs, estados e sumários.

### 3.3. Sumarização

Para uma melhor organização das informações dentro do SDI, foi criado o conceito de classes, que são grupos de máquinas a serem monitoradas. Por exemplo, no caso do Projeto Paraná Digital (seção 4), as classes foram utilizadas para separar as máquinas de acordo com suas posições geográficas, ou seja, em Núcleos Regionais de Ensino (NREs).

Além da localização, o usuário pode organizar suas máquinas pela capacidade de processamento, função, ou qualquer outra característica. A separação dessa forma facilita a busca por máquinas e permite uma administração mais personalizada.

Outro conceito importante é o de estados, responsáveis por informar a situação da rede (ou de máquinas isoladas) em determinado momento, perante alguma característica sendo observada. Situações que podem representar estados são:

- online / offline: máquinas em dois grupos, as que estão ligadas e desligadas.
- versão de software: informações sobre versões de software instalados, importante em parques de máquinas onde a homogeneidade do sistema é um requisito.
- serviços: alerta sobre indisponibilidade de algum serviço.

Sumários são conjuntos de estados relacionados. A quantidade e nome dos sumários é personalizável, assim como a associação entre eles e os estados que representam.

### 3.4. Interface Web

O SDI foi desenvolvido pensando na coleta de dados em tempo real. Este objetivo foi alcançado, porém a maneira de analisar estes dados era muito precária, como verificar arquivos de texto. Um novo conceito de como os dados devem ser apresentados foi criado. Uma interface web provê as informações de maneira sumarizada e intuitiva.

A interface web é baseada em duas ferramentas: *Server Side Includes*[SSI] e XML. SSI faz parte dos servidores de páginas web, e permite utilizar diretivas *include* para unir arquivos uns nos outros. XML é uma forma de enviar dados a um navegador.

Dois diretórios armazenam dados obtidos dos *hosts*: um responsável por armazenar o histórico, e outro por armazenar a última informação coletada. A construção das páginas utiliza como base este último diretório, logo o dado mostrado representa a informação mais recente sobre tal *host*. O SSI recupera estas informações em formato XML e envia para o *browser* do cliente, onde o mesmo é processado e os dados exibidos.

Cada máquina tem seu arquivo XML geral, o qual inclui outros arquivos correspondentes às informações coletadas, um para cada script habilitado. O SDI utiliza outro arquivo XML para classes que incluem os arquivos de *hosts* citados anteriormente. A mesma idéia é utilizada para gerar os sumários.

## 4. Caso de Uso - Projeto Paraná Digital

O Projeto Paraná Digital (PRD) [Castilho et al. 2006] conta com mais de 2100 laboratórios de informática instalados nas escolas públicas do estado do Paraná. Cada laboratório possui ao menos uma servidora e cinco multiterminais [Oliveira et al. 2006] (em *boot* remoto) com quatro terminais cada, totalizando 44 mil pontos de trabalho, utilizando apenas software livre.

```

1 se modo web está habilitado então
2   Gere os arquivos web de estados
3   para cada classe encontrada faça
4     Gere os arquivos web da classe
5   Gere os arquivos web de sumários
6 Crie um arquivo de comandos geral (1)
7 para cada classe encontrada faça
8   para todo host que pertence a classe faça
9     Crie um arquivo de comandos para este host (2)
10    ( enquanto o SDI está aberto faça
11      Marque este host como offline
12      Conecte-se com o host criando um túnel ssh
13      Envie os scripts habilitados em onconnect.d para o túnel
14      Monitore e envie modificações ocorridas nos arquivos (1) e (2)
15      Ao receber mensagens do host envie para a função parse
16      Quando a conexão cair, marque este host como offline
17      Aguarde de 1 a 4 minutos
18    ) em um novo processo

```

**Algoritmo 1:** Algoritmo que descreve o funcionamento básico do SDI

O gerenciamento da rede é realizado através de um núcleo, que comunica-se com todas as escolas através de uma rede interna fornecida pela COPEL<sup>3</sup>, utilizando fibra ótica ou conexão via satélite. Isto permite que toda a administração da rede seja centralizada.

Neste cenário o SDI foi utilizado para automatizar tarefas onde o problema era recorrente e a solução conhecida, e ainda proporcionar diagnóstico prévio de situações onde um auxílio técnico é necessário. As ações tomadas pelo SDI tiveram reflexos principalmente no número de escolas com arquivos corrompidos e nas escolas com problemas em discos rígidos e RAID, o que melhorou consideravelmente o estado da rede.

A interface web gerada para o PRD<sup>4</sup> é um grande exemplo de como o SDI é capaz de gerenciar um grande número de informações, além de organizá-las de maneira sumarizada para tomada de decisões. A figura 1 mostra o número de escolas conectadas no momento, juntamente com outras informações específicas de interesse da rede.

## 5. Conclusão

Ter conhecimento de tudo o que acontece em uma rede pode ser a diferença entre ela estar ou não funcionando corretamente. O SDI foi criado com esta idéia e funciona tanto em redes de grande ou pequeno porte bem como em redes locais ou distribuídas pela internet.

Foram estudados vários softwares existentes para tal propósito, porém nenhum deles conseguiu suprir todas as necessidades da equipe, principalmente a respeito da facilidade de uso e personalização, além da necessidade de visualizar os dados coletados de

<sup>3</sup>Companhia Paranaense de Energia

<sup>4</sup><http://yoda.c3sl.ufpr.br>

SDI	
Selecione uma página: <a href="#">SUMÁRIO</a>	
1899 hosts online	
0 escolas em ajuda remota	
1 escolas sem o arquivo prd.zone	
0 escolas ausentes do prd.zone	
0 escolas em PrD 1.1	
1 escolas sem o arquivo de versão	
<b>Ajuda Remota</b>	
<b>Sem arquivo prd.zone</b>	
ID	Status
escola1460	OFFLINE

**Figura 1. Demonstração do sumário principal do SDI na rede do Paraná Digital**

uma maneira simples e amigável.

O sistema se mostrou efetivo e de grande valia durante seu uso na rede do Projeto Paraná Digital, tornando a gerência da rede um processo mais simples, menos trabalhoso e mais ágil na resolução de problemas. O número de máquinas na rede do projeto ajudou no desenvolvimento de uma solução robusta e confiável para qualquer outro ambiente.

## Referências

- API. Api do Sdi. [http://wiki.c3sl.ufpr.br/sdi/index.php/Criando\\_Scripts](http://wiki.c3sl.ufpr.br/sdi/index.php/Criando_Scripts). Acesso em: Abril, 2009.
- Bona, L. C. E., Castilho, M. A., Weingaertner, D., Silva, F., Sunyé, M. S., Lourenço, L. H. A., and Ribas, B. C. (2008). *Managing a Grid of Computer Laboratories for Educational Purposes*. In: LAGrid08 - Latin American Grid, Campo Grande. Proceedings of the Second Latin American Grind.
- Castilho, M. A., Sunye, M. S., Weingaerter, D., Bona, L. C. E., Silva, F., Direne, A. I., Garcia, L. S., and Guedes, A. L. P. (2006). *Making government policies for education possible by means of Open Source technology: a successful case*, volume 1. Hershey - USA: Idea Group Publishing, 1st edition.
- ClusterIt. Clusterit. <http://clusterit.sourceforge.net/>. Acesso em: Março, 2009.
- ClusterSSH. Clusterssh. <http://clusterssh.wiki.sourceforge.net/>. Acesso em: Março, 2009.
- Oliveira, A., Vignatti, T., Weingaertner, D., Silva, F., Castilho, M., and Sunyé, M. (2006). Um modelo de computacao multiusuário baseado em computadores pessoais. *VII Workshop de Software Livre*.
- Pssh. Parallel Ssh. <http://www.theether.org/pssh/>. Acesso em: Março, 2009.
- Ssh. Secure Shell. <http://www.openssh.org/>. Acesso em: Abril, 2009.
- SSI. Server Side Includes. [http://httpd.apache.org/docs/2.2/mod/mod\\_include.html](http://httpd.apache.org/docs/2.2/mod/mod_include.html). Acesso em: Abril, 2009.