MSPlus: Monitoramento de Algoritmos Distribuídos com Alta Granularidade

Elizeu Elieber Fachini[1] **, Gustavo M. D. Vieira**[1]

[1]DComp – CCGT – UFSCar
Sorocaba, São Paulo, Brasil

elizeuelieber@gmail.com, gdvieira@ufscar.br

Abstract. *In this paper we describe MSPlus, a monitoring tool focused on the observation of resources of a computational cluster during the experimental execution of a distributed algorithm. The monitoring data is then used to characterize the algorithm performance with respect to the usage of these resources. This work main contribution is the first monitoring tool capable of recording data with a 1 s granularity, during relatively long periods of time and with a negligible use of system resources.*

Resumo. *Neste artigo apresentamos MSPlus, uma ferramenta de monitoramento voltada para observação dos recursos de um aglomerado computacional durante a execução experimental de um algoritmo distribuído, com objetivo de caracterizar o desempenho desse algoritmo em relação ao uso desses recursos. A principal contribuição deste trabalho é a primeira ferramenta de monitoramento que permite registrar dados com granularidade de 1 s, durante períodos relativamente longos e com consumo desprezível de recursos do sistema.*

1. Introdução

O *monitoramento* é o ato de coletar informações referentes às características e estado dos recursos de interesse [Zanikolas and Sakellariou 2005]. Em qualquer sistema computacional, seja ele composto por um único computador até grandes aglomerados ou grades computacionais, é necessário obter dados sobre o estado de variados recursos de interesse para verificar e acompanhar o seu comportamento. O monitoramento pode ser utilizado para gerência e alocação de recursos, para detecção e correção de falhas e também para avaliação de parâmetros de desempenho [Zanikolas and Sakellariou 2005]. Isto é feito através de um conjunto de ferramentas de monitoramento, as quais captam informações desses recursos, as processa e as entrega aos usuários.

O conjunto de ferramentas de monitoramento é também chamado de sistema de monitoramento. Em um aglomerado ou grade computacional um sistema de monitoramento deve obter, transferir e agregar de forma coerente as informações de um conjunto de máquinas distintas. Esses serviços são usualmente realizados por um ou mais componentes de *middleware* [Massie et al. 2004, Sottile and Minnich 2002, Tierney et al. 1998], sendo o trabalho de integração dos mesmos nem sempre trivial. Necessidades específicas de monitoramento, como por exemplo a análise *post-mortem* de computações de longa duração, saem do escopo dessas ferramentas.

Neste artigo apresentamos MSPlus, um sistema de monitoramento voltada para observação dos recursos de um aglomerado computacional durante a execução experimental de um algoritmo distribuído, com objetivo de caracterizar o desempenho desse

algoritmo em relação ao uso desses recursos. Para atender a esse objetivo é necessário coletar dados com uma alta granularidade sem no entanto competir com o algoritmo distribuído sendo monitorado por recursos do sistema. A principal contribuição deste trabalho é o primeiro sistema de monitoramento que permite registrar dados com granularidade de 1 s, durante períodos relativamente longos e com consumo desprezível de recursos do sistema.

2. Monitoramento de Sistemas Distribuídos

O sistema de monitoramento em um sistema distribuído é o responsável por capturar, processar e fornecer dados referentes ao que está acontecendo em cada nó pertencente ao sistema. Esses dados podem ser: quantidade de memória usada, taxa de transferência de dados do disco rígido, porcentagem da capacidade de processamento utilizado, entre outros. O sistema de monitoramento comporta-se de diversas maneiras quanto a disponibilidade dos dados: *online*, *semi-online* e *post-mortem*. Em sistemas *online* e *semi-online* os dados são atualizados frequentemente. No caso de *post-mortem*, os dados são apresentados ao término do processo observado ou posteriormente a um período pré-definido para a realização do monitoramento [Tesser 2011]. A transmissão de dados entre componentes de um sistema de monitoramento, é feita através de dois modelos: *pull* e *push*. O modelo *pull* ocorre quando a transmissão de dados é solicitada pelo receptor, ou seja, sob demanda. No modelo *push*, as transmissões de dados são feitas sem a necessidade de solicitação por parte do receptor.

2.1. Trabalhos Relacionados

De forma geral, sistemas de monitoramento de aglomerados/grades estão preocupados com o monitoramento operacional de vastos sistemas computacionais, com objetivo primordial de detectar anomalias em seu funcionamento. A grande variedade desses sistemas pode ser observada no contexto da taxonomia de sistemas de monitoramento de grade proposta por [Zanikolas and Sakellariou 2005]. Entre alguns dos sistemas mais conhecidos, podemos citar: Ganglia [Massie et al. 2004], Netlogger [Tierney et al. 1998], Supermon [Sottile and Minnich 2002], Munin[1], entre outros.

Infelizmente, praticamente todos os sistemas compartilham as mesmas decisões de projeto no que se refere a granularidade de coleta, sumarização e arquitetura de distribuição dos dados de monitorização. Todos esses sistemas coletam dados com pouca regularidade, fazendo amostragens da ordem de 5 a 15 minutos. Essas amostragens pouco regulares são suficientes para o propósito desses sistemas e ajudam a reduzir o impacto total do monitoramento sobre o sistema sendo monitorado. Devido ao tempo que ficam executando, esses sistemas resumem os dados monitorados tendo apenas dados com a granularidade mínima referentes a algumas horas ou minutos no passado. Dados mais antigos são resumidos em unidades de amostragem maiores de horas a até dias. Por fim, sistemas de monitoramento de grade distribuem as informações através de arquitetura complexas, de alto custo computacional. Entre estes custos podemos listas o uso de CPU dos nós sendo monitorados, o disco desses nós e a rede usada para distribuir esses dados dentro do aglomerado.

[1] `http://munin-monitoring.org/`

3. MSPlus

3.1. Visão Geral

Para contornar os problemas encontrados nos sistemas convencionais de monitoramento de aglomerados/grades ao serem empregados para monitorar execuções de algoritmos distribuídos criamos um novo sistema chamado MSPlus. O projeto desse sistema está atrelado ao seu caso de uso principal, ser uma ferramenta de auxílio ao projetista e implementador de cenários de teste de sistemas distribuídos. Dessa forma, esse é um sistema simples, autocontido, de fácil compreensão e de baixo custo de execução. Dentro desse caso de uso, o MSPlus além de fazer monitoramento permite gerar gráficos e exportar os dados em formato de texto.

O MSPlus tem como característica principal a possibilidade de monitorar cada máquina do aglomerado com granularidade de até 1 s sem causar grande impacto nos recursos do sistema sendo monitorado. Dependendo das necessidades do projetista do experimento, período de amostragem maiores que 1 s podem ser usados, com a garantia que todas as amostras colhidas podem ser acessadas posteriormente. Ou seja, o MSPlus não faz nenhum tipo de resumo dos dados, nunca descartando nenhuma amostra feita. Adicionalmente, o sistema não escreve no disco nem tampouco trafega dados pela rede durante o monitoramento. Não usar esses recursos é importante nesse contexto, pois mesmo que o disco e a rede em um sistema de aglomerados sejam de alta velocidade, qualquer utilização dos mesmos poderá comprometer o resultado real da medida de desempenho destes dispositivos. Desta forma, todas as etapas de processamento, resumo e análise dos dados são feitas *post-mortem*. O modelo de distribuição de dados é *pull*, com o cuidado adicional de que a própria distribuição da dados assim como o processamento também ocorre *post-mortem*.

O MSPlus é desenvolvido em Python para sistemas Linux e tem como principal intuito capturar dados referentes aos estados dos dispositivos de um sistema computacional e os armazenar. Para garantir um baixo consumo de recursos da máquina o MSPlus é muito simples, sendo apenas um programa Python praticamente autocontido em um único arquivo. A única dependência externa do MSPlus é a biblioteca RRDtool[2] usada para armazenar os dados no disco.

3.2. Arquitetura de Software

Considerando as características de um sistema de monitoramento, a ferramenta é composta por um consumidor e *plugins* de dispositivos integrados ao sistema principal. A Figura 1 mostra o sistema de forma simplificada, mostrando que o consumidor consulta os *plugins* para obter as métricas sendo monitoradas e armazena os dados obtidos usado o banco de dados do RRDtool.

Uma propriedade interessante do MSPlus é que ele monitora sistemas distribuídos sem a distribuição *online* dos dados. O sistema de monitoramento poderá ser iniciado em todos os nós monitorados por um *script* independente (como `mpirun`) ou integrado ao mesmo mecanismo utilizado para iniciar o sistema distribuído que está sendo monitorado. Essa integração é particularmente direta devido a simplicidade da ferramenta. Posteriormente os dados são coletados em cada nó, novamente usando os mecanismos de coleta de

[2] `http://oss.oetiker.ch/rrdtool/`

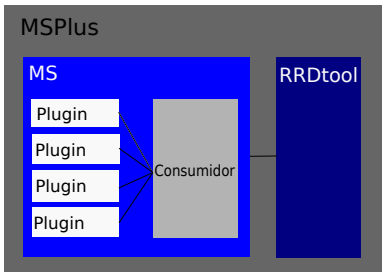

Figura 1. MSPlus

dados experimentais usados pelo sistema distribuído em teste, e processados *post-mortem* de forma centralizada. O sistema de monitoramento possui um suporte simples para a geração de gráficos que pode ser usado imediatamente após uma execução e também permite a exportação dos dados, permitindo uma análise integrada ao processamento dos dados do experimento sendo monitorado. Os dados exportados estão em arquivos texto, uma representação menos eficiente mas que pode ser facilmente usada por várias ferramentas de análise como R, GnuPlot e Excel.

O MSPlus captura os dados por um tempo previamente determinado, com frequência de um ou mais segundos, e os envia provisoriamente para a memória. Após a captura de todas as amostras é feito um processamento prévio dos dados e os mesmos são armazenados no RRDtool. O processamento prévio é feito apenas em alguns casos, pois nem todas as métricas necessitam de tratamento pós captura. O resultado da monitorização do experimento fica registrado em vários bancos de dados do RRDtool registrando as várias métricas. O RRDtool é uma biblioteca voltada para o resumo de grandes volumes de dados, mas no MSPlus ele é configurado para preservar com fidelidade todas as amostras coletadas.

Internamente ao MSPlus o monitoramento em si é feito por *plugins* de dispositivos, responsáveis por fazer a captura dos estados dos dispositivos e os entregar ao consumidor. Esses *plugins* foram desenvolvidos buscando o menor custo possível, sendo então desenvolvidos usando a mesma linguagem da aplicação principal e sendo executados dentro do processo principal do sistema de monitoramento. Por mais simples que pareça, essa é um ponto onde o MSPlus se destaca em relação a sistemas de monitoramento existentes. Os *plugins* atualmente implementados na ferramenta são os seguintes:

Dispositivo de armazenamento: Monitora o comportamento dos dispositivos de disco do sistema, computando métricas como: vazão em operações por segundo e bytes por segundo, latência e utilização.

CPU: Monitora a utilização de CPU, dividida em métricas como consumo de usuário, sistema, espera E/S, etc.

Memória: Monitora a utilização de memória, com medidas detalhadas de quantidade de memória usada pelo núcleo, aplicativos, cache, buffers e swap.

Rede: Monitora a vazão de rede em bytes pode segundo recebidos e enviados.

4. Disponibilidade e Demonstração de Uso

O MSPlus pode ser obtido em código fonte, sem necessidade de instalação, em sua página de projeto: `https://bitbucket.org/elizeuelieber/msplus`. A documentação do programa está contida no arquivo `Readme` acessível na página de entrada

do projeto (`https://bitbucket.org/elizeuelieber/msplus`) e pela documentação *online* acessível executando-se `python msplus.py -help`.

Para o salão de ferramentas do SBRC planejamos dois tipos de demonstrações, de acordo com o tempo disponível dos interessados. Em menos de 5 minutos podemos demonstrar a coleta de dados de monitoramento do computador portátil usado na demonstração. Como coletamos dados segundo a segundo, é possível observar comportamentos interessantes neste espaço curto de tempo. Em 30 minutos é possível montar uma execução experimental no aglomerado da UFSCar, coletar os dados e correlacionar as métricas de execução com as métricas monitoradas. Em ambas as situações não é necessário nenhum equipamento ou espaço especial para a demonstração, somente conexão à Internet. Adicionalmente, como as execuções a serem monitoradas podem demorar um tempo potencialmente longo, a demonstração pode ser feita combinando-se o início da execução do sistema de monitoramento, mas com a análise de dados coletados em outro momento. Esse seria uma opção indicada para demonstração a grandes grupos e não apenas a indivíduos.

5. Dados Experimentais

Nesta seção apresentamos alguns resultados experimentais que mostram a funcionalidade e as vantagens do uso do MSPlus. Para esses testes focamos no comportamento de uma única máquina para enfatizar as características mais marcantes da ferramenta: a alta granularidade e o baixo consumo de recursos do sistema.

Todos os testes foram executados em uma máquina x86_64, com processador Intel Core i5-3320M, 8GB RAM e 1TB HD. Essa é uma máquina menos robusta do que a usualmente encontrada em aglomerados, o que é interessante para mostrar que o MSPlus exige poucos recursos do sistema. Nesta máquina rodava o SO Ubuntu Linux 14.04.2 LTS, Python 2.7.4 e RRDtool 1.47.

5.1. Métricas e Granularidade

As figuras a seguir mostram um exemplo das métricas registradas pelo MSPlus. Todas essas figuras registram a mesma execução de aproximadamente 3 h (11.000 s) da máquina de testes enquanto a mesma era usada normalmente. Na Figura 2 podemos observar o uso de CPU da máquina sendo testada.

A medida de CPU da máquina de teste ilustra como a granularidade de medida da métrica pode ser útil. Na Figura 2 (a) observamos uma anomalia por volta de 5000 s. Como possuímos todos os dados disponíveis com granularidade de 1 s, foi possível analisar com mais detalhes os valores da métrica ao redor desse instante de tempo como ilustrado nas Figuras 2 (b), (c) e (d). Assim, podemos observar que algum processo começou a ocupar 100% de CPU por volta de 4808 s e podemos usar esse valor para correlacionar o comportamento de outras métricas.

Como exemplo de outras métricas captadas pelo MSPlus, a Figura 3 (a) mostra o consumo de memória por vários componentes do sistema e Figura 3 (b) mostra o volume de dados transmitidos pela rede. A Figura 4 mostra a vazão do dispositivo de E/S (disco), onde a Figura 4 (a) mostra essa vazão em operações de E/S e a Figura 4 (b) em megabytes.

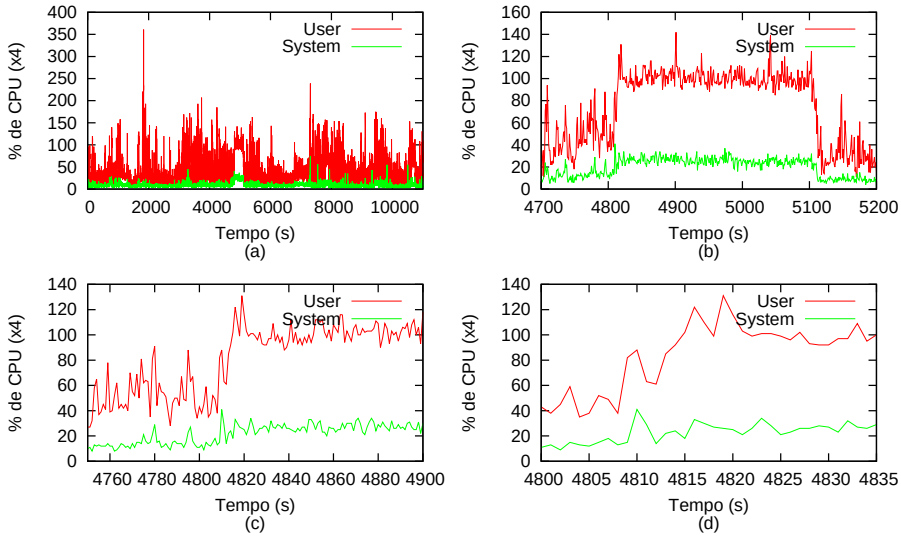

Figura 2. CPU

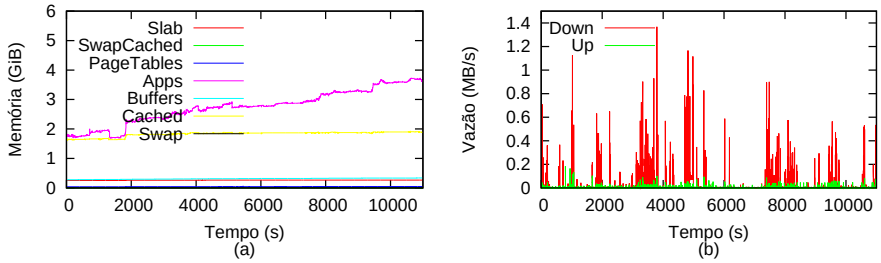

Figura 3. Memória e Rede

5.2. Baixo Consumo de Recursos

Para demonstrar o baixo consumo de recursos do MSPlus, construímos um experimento de longa duração que o compara a ferramenta Munin. Nesse experimento tanto o MSPlus quanto o Munin estão com suas configurações padrão a exceção do tempo de amostragem que foi configurado para 1 s. Os dois sistemas de monitoramento foram executados então por aproximadamente 6 h e 23 m (23.000 s) na máquina de testes sendo que a mesma ficou completamente ociosa durante todo o tempo do experimento. Os dados comparativos das duas ferramentas estão nas Figuras 5, 7 e 6 que mostram as métricas de CPU, memória e vazão de E/S, respectivamente. Observando essas figuras em comparação às figuras da

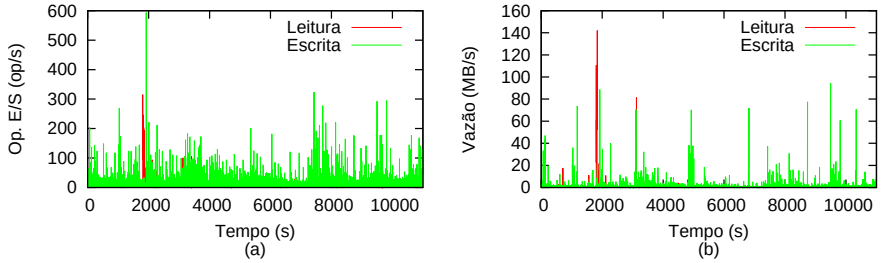

Figura 4. Vazão de E/S

Seção 5.1 é possível observar que a máquina encontrava-se realmente ociosa.

Em relação ao consumo de CPU o MSPlus usa em média 3,41% de uma das CPUs do sistema (*system + user*) com desvio padrão de 1,41, enquanto o Munin usa 29,1% com desvio padrão de 1,9. Usando o teste-t existe 0% de probabilidade que, ao longo desta execução de 23.000 s essa diferença se deva ao acaso. A vantagem do MSPlus nesse caso se deve ao fato que ele não implementa *plugins* como processos separados e que não faz uso de *shell scripts* para acessar as métricas disponibilizadas pelo núcleo do Linux. Disparar um sub processo e um interpretador de comandos tem um alto custo em termos de ciclos de CPU.

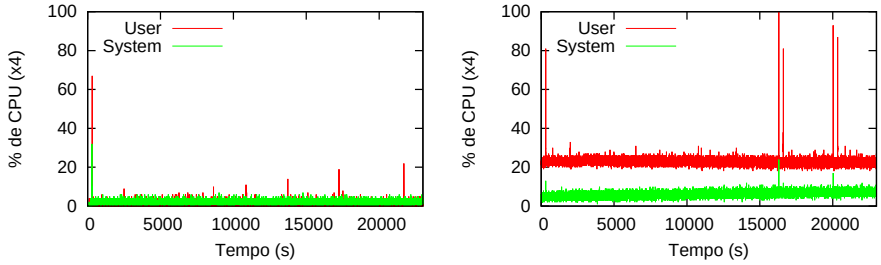

Figura 5. CPU

A vazão de E/S do MSPlus durante o teste teve média de 0,0031 MB/s com desvio padrão de 0,0295, enquanto o Munin registrou o valor de 2,01 MB/s com desvio padrão de 0,17. O teste-t mostra que essa diferença é estatisticamente significativa com 0% de probabilidade de ser um variação ao acaso. O valor alto do Munin se deve ao fato que como cada execução do mesmo é isolada das demais, ele deve usar a memória persistente para armazenar variáveis usadas para computar as métrica. O MSPlus opera inteiramente em memória, tendo impacto mínimo nas operações de E/S.

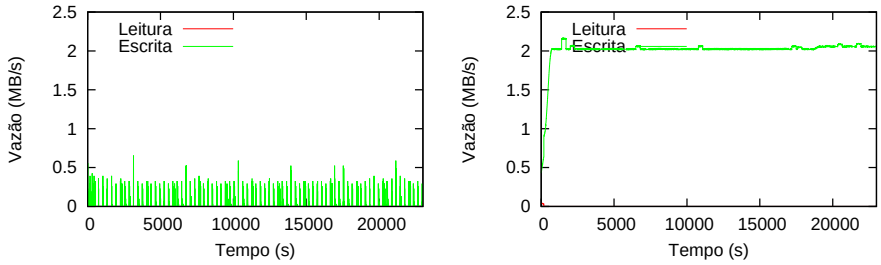

Figura 6. Vazão de E/S

O uso de memória do MSPlus e do Munin pode ser avaliado indiretamente usando os dados globais de consumo de memória obtidos. No início da execução do MSPlus o sistema utilizava 0,61 GB de memória com aplicativos, enquanto que no fim da execução esse valor ficou em 0,98 GB. Logo o MSPlus usou 0,37 GB durante as 6 h e 23 m da execução. O Munin começou sua execução com 0,66 GB de memória usados por aplicativos e terminou com 0,83 GB utilizados. O uso total do Munin foi de 0,17 GB durante toda a execução. Neste critério o MSPlus usou mais recursos do sistema que o Munin, o que era esperado pois o MSPlus usa memória como recurso para economizar disco e rede. Na verdade, o consumo do Munin foi de certa forma surpreendente pois ele não guarda

nenhuma informação na memória principal, o que indica que o consumo observado se deve a algum vazamento de memória ou outro erro de programação.

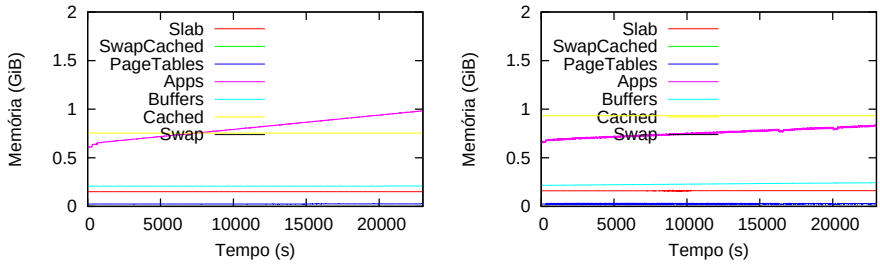

Figura 7. Memória

Em resumo, o MSPlus apresentou um consumo consideravelmente inferior ao Munin nas métricas de uso de CPU e de vazão de E/S, porém apresentou um consumo maior de memória. Esse consumo de memória é esperado devido às decisões de projeto do MSPlus, mas pode ser mitigado com escritas eventuais ao disco. Essas escritas poderiam ser controladas por parâmetros de linha de comando de tal forma que o projetista do experimento tenha controle sobre como a ferramenta utiliza os recursos disponíveis. Essa funcionalidade, no entanto, ainda não foi implementada.

6. Conclusão

Neste artigo apresentamos MSPlus, o primeiro sistema de monitoramento que permite registrar dados com granularidade de 1 s, durante períodos relativamente longos e com consumo desprezível de recursos do sistema. Demonstramos experimentalmente que a ferramenta é capaz de apurar dados com alta granularidade e mostramos um uso consideravelmente menor de recursos do sistema em comparação a uma ferramenta convencional configurada para o mesmo propósito. O MSPlus é bem simples, podendo ser facilmente integrado em ambientes de execução de algoritmos distribuídos, permitindo a investigação de características de desempenho do algoritmo sendo executado e correlacioná-las a métricas de desempenho das máquinas onde executa.

Referências

Massie, M. L., Chun, B. N., and Culler, D. E. (2004). The Ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30:817–840.

Sottile, M. J. and Minnich, R. G. (2002). Supermon: A high-speed cluster monitoring system. In *Cluster Computing, 2002. Proceedings. 2002 IEEE International Conference on*, pages 39–46. IEEE.

Tesser, R. K. (2011). Monitoramento on-line em sistemas distribuídos: Mecanismo hierárquico para coleta de dados.

Tierney, B., Johnston, W., Crowley, B., Hoo, G., Brooks, C., and Gunter, D. (1998). The netlogger methodology for high performance distributed systems performance analysis. In *High Performance Distributed Computing, 1998. Proceedings. The Seventh International Symposium on*, pages 260–267.

Zanikolas, S. and Sakellariou, R. (2005). A taxonomy of grid monitoring systems. *Future Generation Computer Systems*, 21(1):163–188.