

Utilização de Scripts para Monitoramento de Sistemas Linux: Abordagem para Criação de Relatórios e Gráficos com Ferramentas *open-source*

Matheus Torquato (IF-Sertão PE - Salgueiro)

Harley Mello (IF-Sertão PE - Salgueiro)

Lucas Torquato (IFAL - Reitoria)

Jean Araujo (UAG - UFRPE)

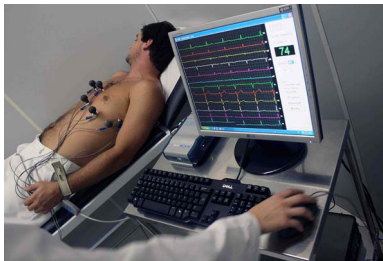
Erico Guedes (IFAL - Palmeira dos Índios)



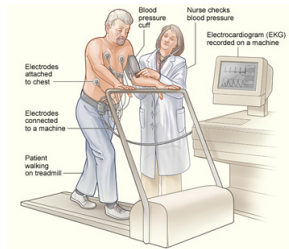
FreeBASE, ERBASE 2015, Salvador - BA

16 de abril de 2015

Por que monitorar sistemas?

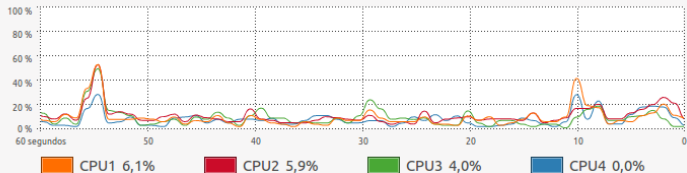


Checar status

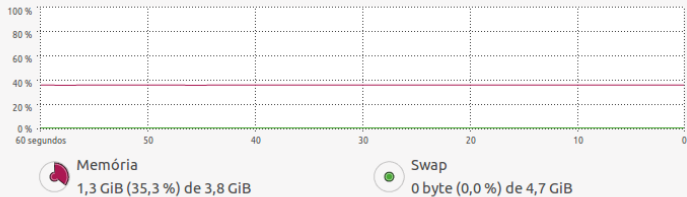


Planejamento de capacidade

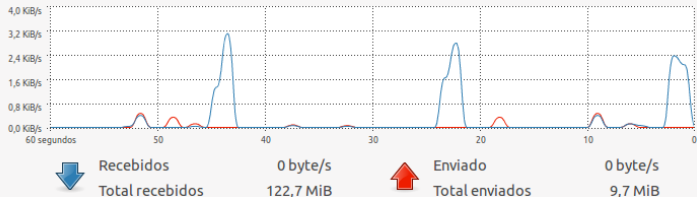
Histórico da CPU



Histórico da memória e swap



Histórico da rede



Grande ideia!



Grande ideia!

- Avaliar o consumo de recursos de um processo específico
- Monitoramento de recursos de diversas naturezas
 - ▶ Como a descarga de bateria pode afetar o desempenho de um processo específico? E seu consumo de recursos?
- Validar uma nova abordagem, uma nova ideia

Problemas

- Nem sempre as ferramentas possuem flexibilidade
 - ▶ Necessidade de adaptação
- Curva de aprendizagem em ferramentas complexas
- Criação de relatórios e gráficos específicos

Possibilidades

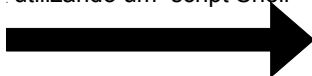
- Sistemas Linux fornecem diversas ferramentas para monitoramento
- Muitas rodam em CLI (*Command Line Interface*)
 - ▶ Geralmente, aprende-se mais rápido
 - ▶ Respostas imediatas
 - ▶ Shell Script com acesso direto às ferramentas
- Ferramentas flexíveis para geração de gráficos como Gnuplot

Roteiro

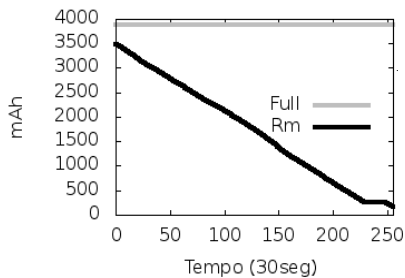
- 1 Problemática
- 2 Abordagem proposta
- 3 Algumas ferramentas de monitoramento
- 4 Estudo de caso
- 5 Conclusões



Monitoramento de recursos
utilizando um script Shell



Descarga de bateria



Geração de gráficos
utilizando o Gnuplot

Roteiro

- 1 Problemática
- 2 Abordagem proposta
- 3 Algumas ferramentas de monitoramento**
- 4 Estudo de caso
- 5 Conclusões

Algumas ferramentas de monitoramento [Andresen, R. (2004)]

- `top`
 - ▶ Tempo real
- `ps`
 - ▶ Comando estático
- `iostat`
 - ▶ *sysstat*
- `/proc/meminfo`
 - ▶ Pseudo-Sistema de arquivo `/proc`

Roteiro

- 1 Problemática
- 2 Abordagem proposta
- 3 Algumas ferramentas de monitoramento
- 4 Estudo de caso**
- 5 Conclusões

Monitoramento do consumo de recursos computacionais (CPU e RAM) de um processo

- Definir comando para obtenção das métricas de interesse.

- ▶ `ps -p $PID -o %cpu,%mem`

- Saída

%CPU %MEM

1.0 3.8

Monitoramento do consumo de recursos computacionais (CPU e RAM) de um processo

- Saída

%CPU %MEM

1.0 3.8

- É necessário filtrar a saída para recolher apenas os dados necessários

- ▶ `ps -p $PID -o %cpu,%mem | grep [0-9]`

- Nova Saída

1.0 3.8

Monitoramento do consumo de recursos computacionais (CPU e RAM) de um processo

- Elaboração do Shell Script para recolher resultados
 - ▶ Criar arquivo de relatório
 - ▶ Criar um cabeçalho para o arquivo
 - ▶ Recolher (incrementalmente) os resultados do monitoramento

Monitoramento do consumo de recursos computacionais (CPU e RAM) de um processo

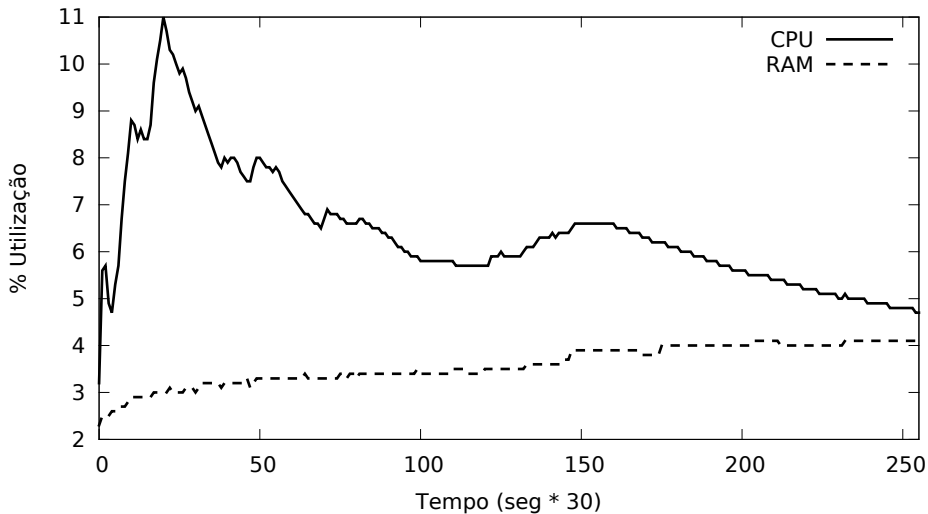
Algoritmo 1 Monitor de processos

```
1: # /bin/bash
2: echo "Contador %CPU %MEM"»"ChromeMonitor".txt
3: CONTADOR=0
4: while True do
5:     echo $CONTADOR $(ps -p 5382 -o %cpu,%mem | grep [0-9]) »"Ch-
      romeMonitor".txt
6:     sleep 30
7:     CONTADOR=$((CONTADOR+1))
8: end while
```

Arquivo de saída (Relatório em tsv)

Contador	%CPU	%MEM
0	3.2	2.3
1	5.6	2.5
2	5.7	2.5
3	4.9	2.5
4	4.7	2.6
...		

Monitoramento do Google Chrome Web Browser



Roteiro

- 1 Problemática
- 2 Abordagem proposta
- 3 Algumas ferramentas de monitoramento
- 4 Estudo de caso
- 5 Conclusões

Conclusões

- Abordagem para a flexibilização do monitoramento de Sistemas Linux
 - ▶ Criação de relatórios e gráficos personalizados
 - ▶ Scripts de exemplo: `LaptopBatteryMonitor.sh` e `TaskMonitor.sh`
 - ▶ Templates do gnuplot: `LaptopBatteryMonitor.tpl` e `TaskMonitorPlot.tpl`
- Facilitar a implementação de monitoramento personalizado.

Dúvidas?

Material:

<https://github.com/matheustor4/LinuxMonitoring>

- Matheus Torquato (matheustor4.professor@gmail.com)
- Harley Mello (harley.macedo@ifsertao-pe.edu.br)
- Lucas Torquato (lucas.torquato@ifal.edu.br)
- Jean Araujo (jean@uag.ufrpe.br)
- Erico Guedes (erico@ifalpalmeira.edu.br)

Obrigado!