

# SISTEMA DISTRIBUÍDO PARA MONITORAR O USO DOS RECURSOS DE HARDWARE E SOFTWARE EM ESTAÇÕES DE TRABALHO GNU/LINUX

Jamiel Spezia

Orientador: Maglan Cristiano Diemer

Lajeado, dezembro de 2007

# Sumário

- Introdução
- Trabalhos relacionados
- Sistema de monitoração
- Implementação
- Avaliações
- Demonstração do sistema
- Conclusão
- Console

# Introdução

- Importância do monitoramento de Hardware e Software
- Controle manual torna-se complicado
- Soluções existentes
  - Proprietárias
  - SO Windows
  - SO GNU/Linux – Análise de rede, inventário e controle de configurações

# Introdução

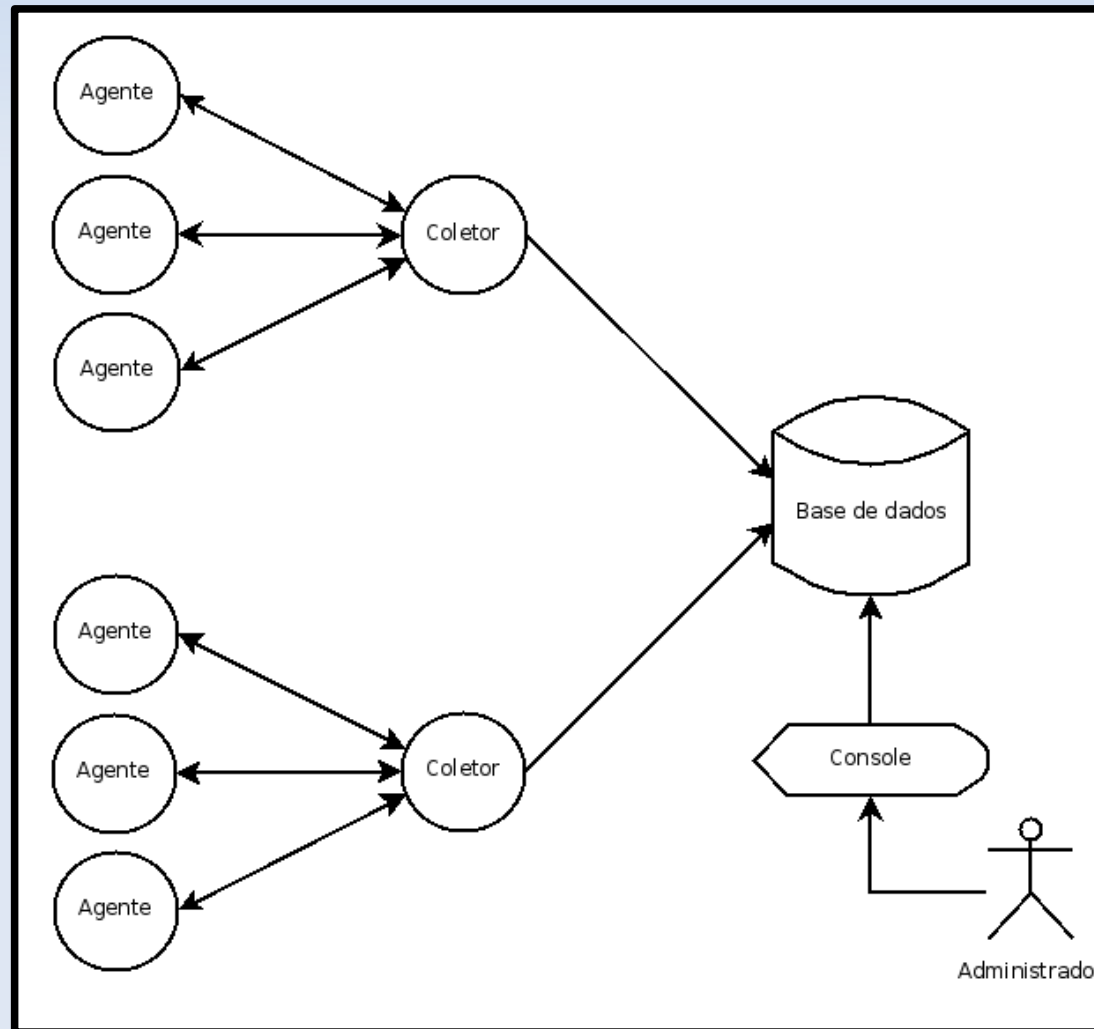
- Objetivo
  - Desenvolver um solução para o SO GNU/Linux
- Motivação
  - Contribuir com um sistema para tal propósito
  - Desafio pessoal

# Trabalhos relacionados

- NetEye
- TraumaZero
- Cacic
- Puppet
- Hyperic HQ
- Zenoss

# Sistema de monitoração

- Estrutura do sistema

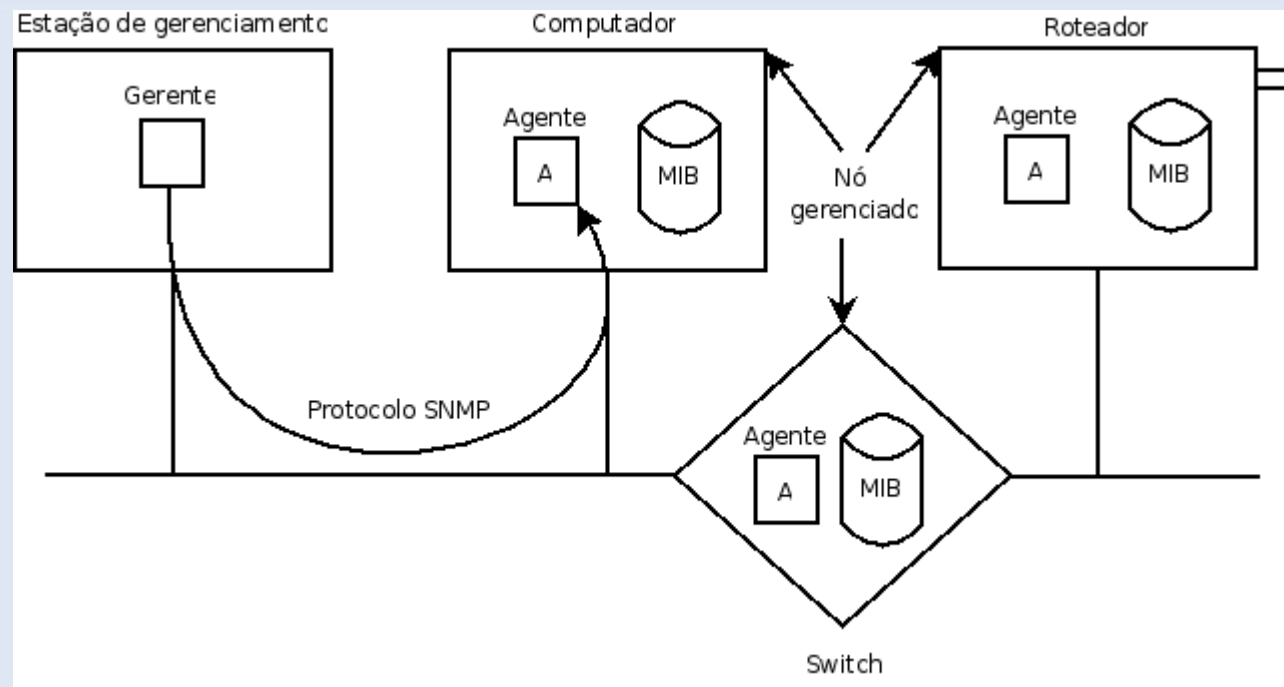


# Sistema de monitoração

- Processos
- Kernel
  - Gerenciamento de tempo
    - HZ
    - Jiffies
  - Gerenciamento de memória
    - Páginas

# Sistema de monitoração

- Comunicação
  - Protocolo SNMP



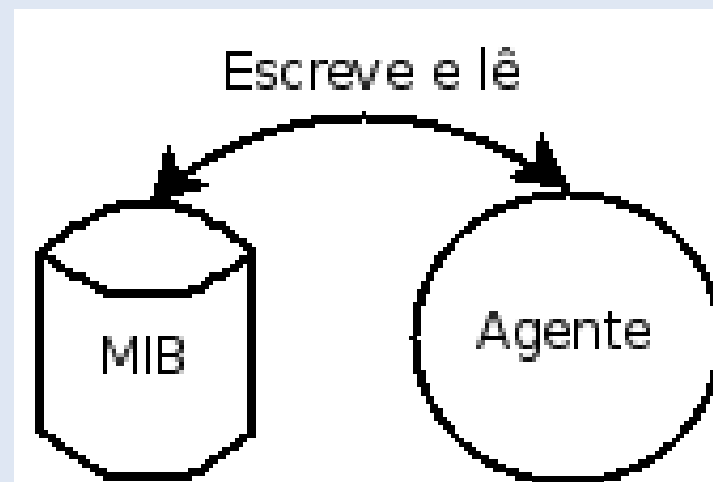


# Implementação

- SDMR foi desenvolvido em C
- 2 aplicações: Agente e Coletor
- 1 SGBD: postgresql
- Protocolo SNMP: netSnmp

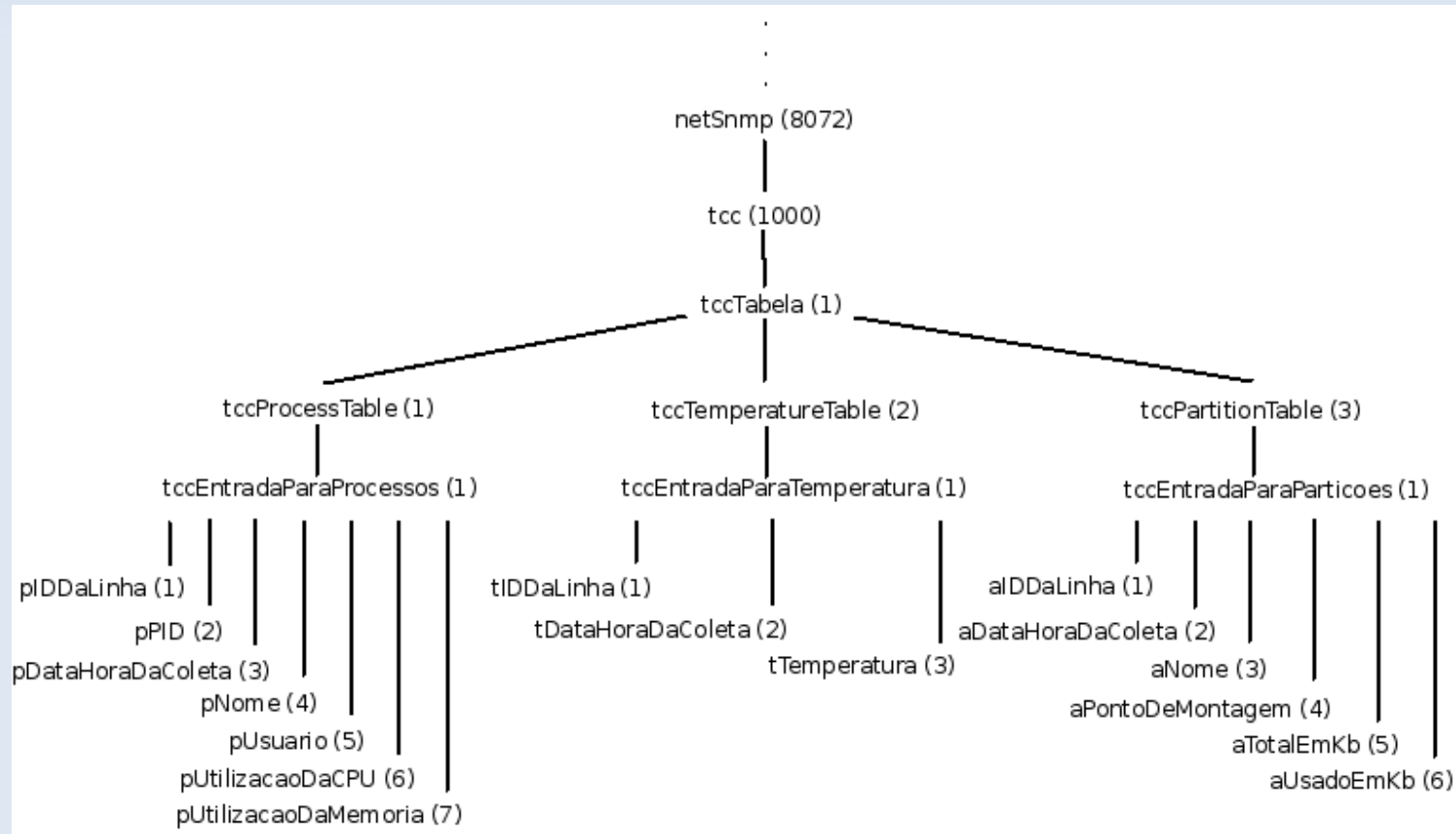
# Implementação

- Agente
  - Agente extensível do SNMP
    - Registra MIB
    - Gerenciamento das informações armazenadas
  - Captura as informações da estação de trabalho



# Implementação

- Agente
  - Estrutura da MIB



# Implementação

- Agente
  - Informação do processador
    - /proc/<pid>/stat

```
jamiel@eagle:~$ cat /proc/6072/stat
6072 (soffice.bin) S 5831 5747 5747 0 -1 4202496 24580 82 533 0 1258 63 0 0 15 0
6 0 17918 294838272 22340 4294967295 134512640 134856208 3219435648 3219434204
4294960144 0 0 4096 2076271871 4294967295 0 0 17 1 0 0 0
```

- 1ª informação: Identificador do processo
- 2ª informação: Nome do processo
- 14ª informação: *jiffies* em modo usuário
- 15ª informação: *jiffies* em modo Kernel

# Implementação

- Agente
  - Informação de memória
    - /proc/<pid>/statm

```
jamiel@eagle:~$ cat /proc/6072/statm  
75876 23437 16994 84 0 21746 0
```

- 2ª informação: número de páginas

# Implementação

- Agente
  - Informações do disco rígido
    - *getmntent()*
    - *statfs()*
  - Temperatura do processador
    - */proc/acpi/thermal\_zone/TZ00/temperature*

# Implementação

- Agente

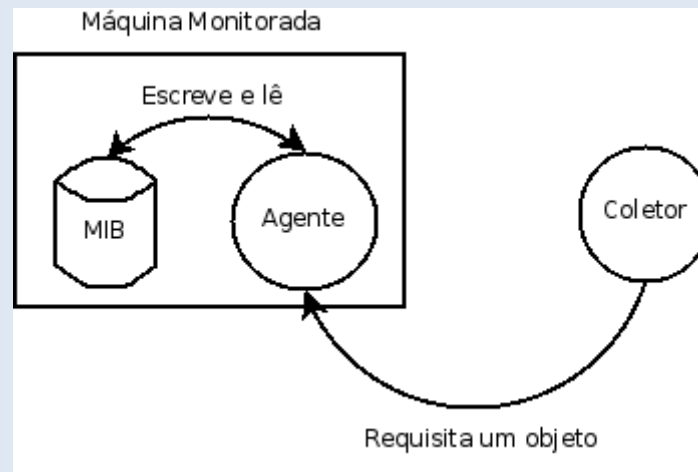
- Processador e memória

- Taxa de uso do processador  $p = \frac{(jiffies' - jiffies)}{(HZ * (t' - t))} * 100$

- Taxa de uso da memória  $p = \frac{memOcupada}{totalDeMemória} * 100$

# Implementação

- Coletor
  - Coleta informações dos agentes
  - Agrupa informações dos processos
  - Armazena na base de dados





# Implementação

- Coletor
  - Informações dos processos vindas do agente

dataHoraDaColeta	PID Nome	Dono	% CPU	% Memória	
20071116 140500	5874 Xorg	root	10	30	Tempo 1
20071116 140500	5876 kmail	jamiel	6	7	
20071116 140502	5874 Xorg	root	20	30	Tempo 2
20071116 140502	5876 kmail	jamiel	12	7	
20071116 140504	5874 Xorg	root	6	30	Tempo 3
20071116 140504	5987 kopete	jamiel	6	3	

- Agrupamento

dataHoraInicial	dataHoraFinal	PID Nome	Dono	% CPU	% Memória
20071116 140500	20071116 140504	5874 Xorg	root	12	30
20071116 140500	20071116 140504	5876 kmail	jamiel	6	7
20071116 140500	20071116 140504	5987 kopete	jamiel	2	3

# Implementação

- Base de dados
  - Postgresql
  - Tabelas

processos
#id: integer
+nomeDoHost: varchar(50)
+pid: integer
+dataHoraInicial: timestamp
+dataHoraFinal: timestamp
+nome: varchar(50)
+usuario: varchar(50)
+utilizacaoDaCPU: float
+utilizacaoDaMemoria: float

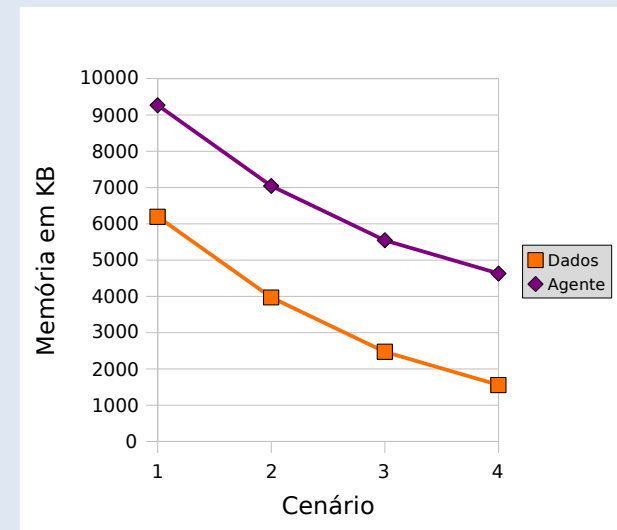
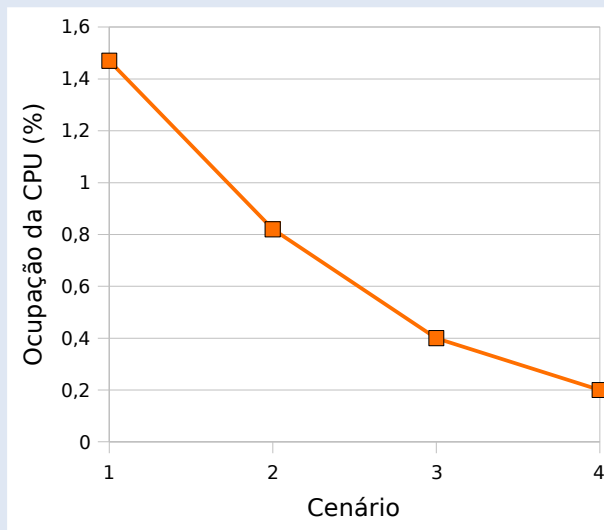
particoes
#id: integer
+nomeDoHost: varchar(50)
+dataHoraDaColeta: integer
+nome: varchar(50)
+pontoDeMontagem: varchar(50)
+totalEmKb: float
+usadoEmKb: float

temperaturas
#id: integer
+nomeDoHost: varchar(50)
+dataHoraDaColeta: integer
+temperatura: integer

# Avaliações

- Impactos causados pelo agente na estação de trabalho
  - 1 hora – Intel® Core™ 2 Duo de 1,66 Ghz

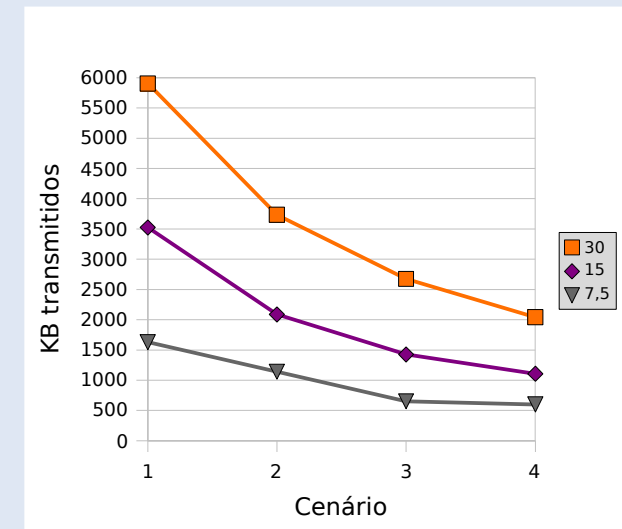
Cenário	Processos	Temperatura	Partições
1	2	60	360
2	4	120	720
3	8	240	1440
4	16	480	2880



# Avaliações

- Impacto causado na rede pela transmissão das informações entre o agente e o coletor
  - 1 agente para 1 coletor – Wireshark

Cenário	Processos	Temperatura	Partições
1	2	60	360
2	4	120	720
3	8	240	1440
4	16	480	2880



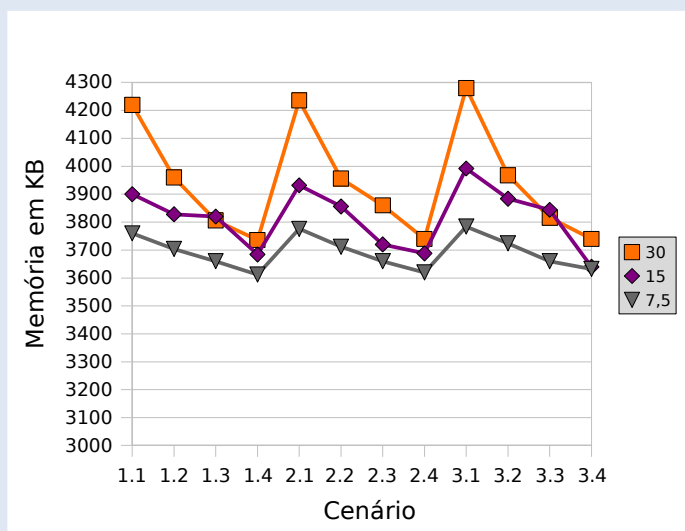
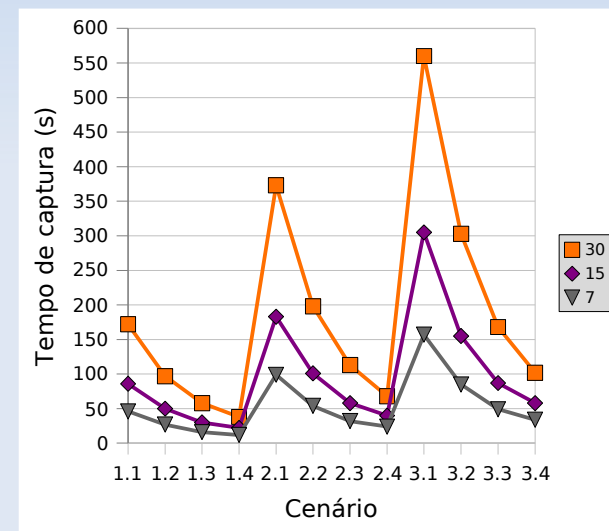
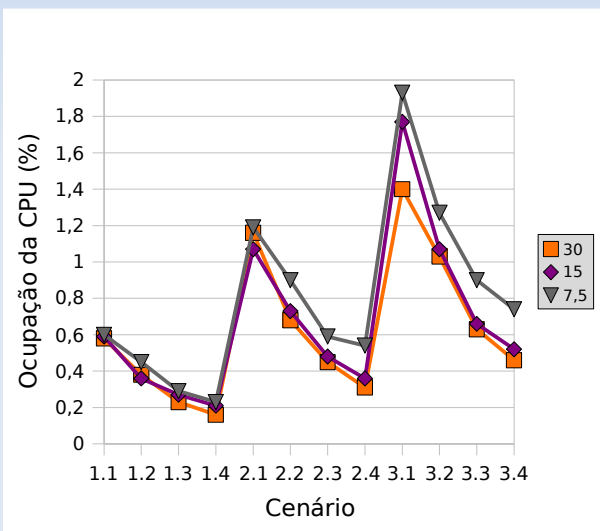
# Avaliações

- Estimativa do número de agentes por coletor

Cenário	Nº de agentes	Processos	Temperatura	Partições
1.1	10	2	60	360
1.2	10	4	120	720
1.3	10	8	240	1440
1.4	10	16	480	2880
2.1	20	2	60	360
2.2	20	4	120	720
2.3	20	8	240	1440
2.4	20	16	480	2880
3.1	30	2	60	360
3.2	30	4	120	720
3.3	30	8	240	1440
3.4	30	16	480	2880

# Avaliações

- Estimativa do número de agentes por coletor

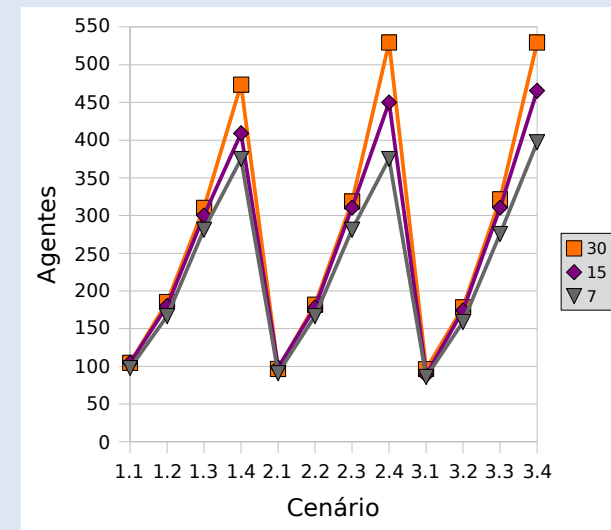


# Avaliações

- Estimativa do número de agentes por coletor

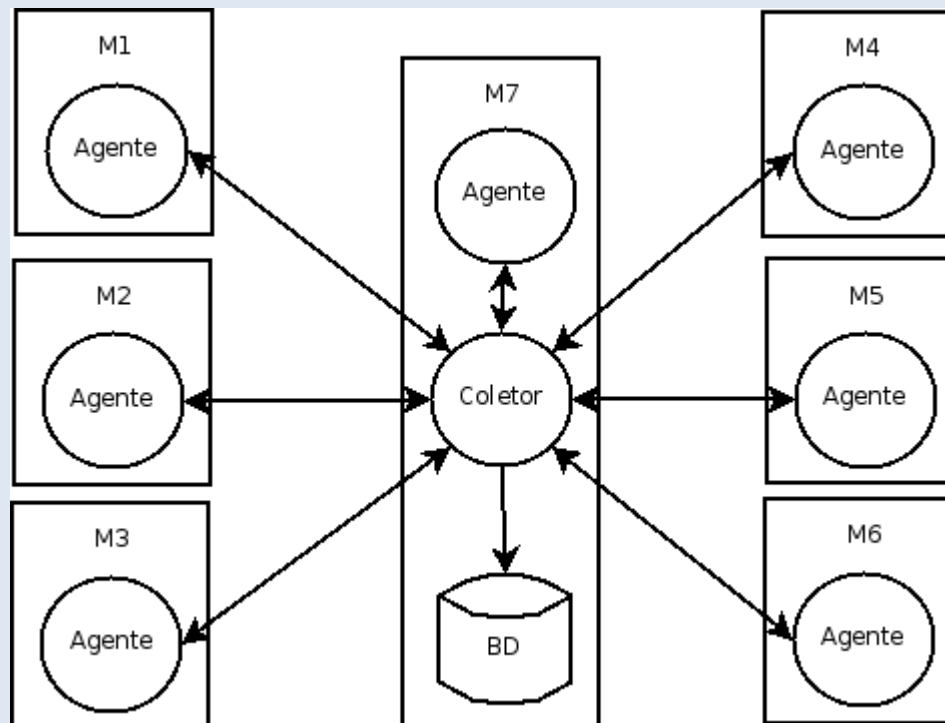
Cenário	% CPU	Tempo de coleta (s)	Nº máx.
1.1	0,6	46	97
1.2	0,45	27	166
1.3	0,29	16	281
1.4	0,23	12	375
2.1	1,19	99	90
2.2	0,9	54	166
2.3	0,59	32	281
2.4	0,54	24	375
3.1	1,93	157	85
3.2	1,27	85	158
3.3	0,9	49	275
3.4	0,74	34	397

$$e = \frac{(tdt * na)}{tc}$$



# Avaliações

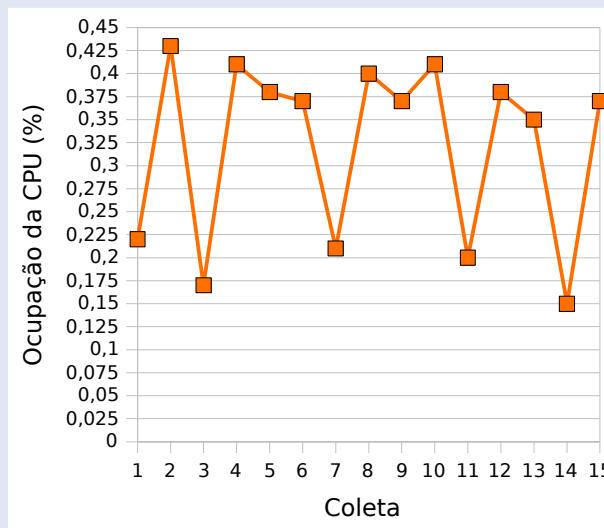
- Teste real
  - Cenário utilizado
    - Tempo de captura: 32 – 480 - 2880
    - Tempo de coleta: 10 minutos





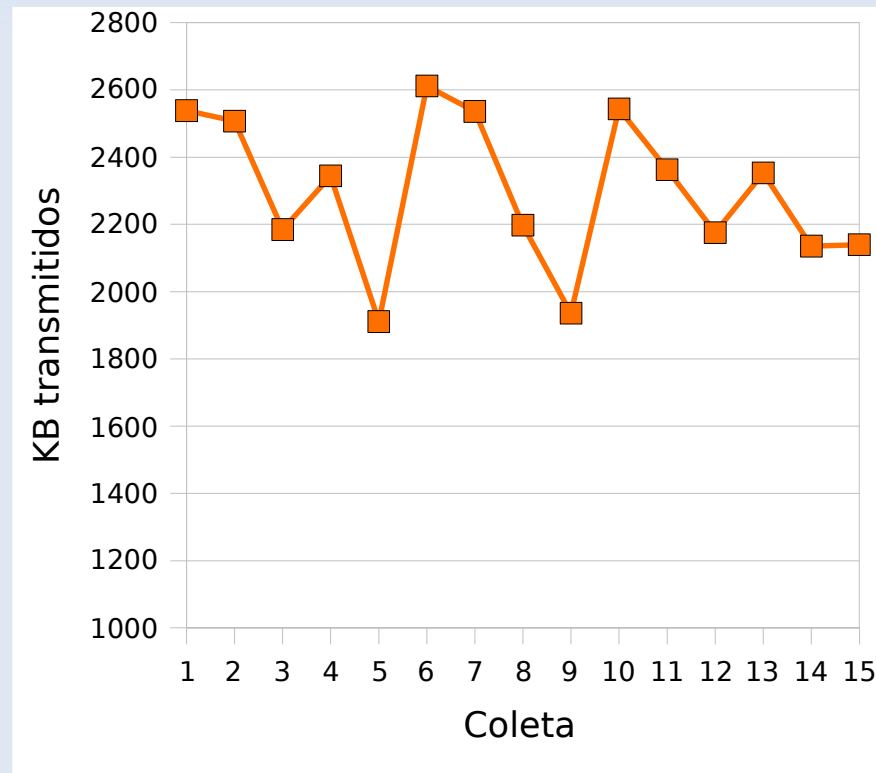
# Avaliações

- Teste real
  - Agentes
    - Processamento abaixo de 0,3%
    - Memória abaixo de 4 MB
  - Coletor
    - Memória abaixo de 4.1 MB
    - Processamento



# Avaliações

- Teste real
  - Comunicação



# Demonstração

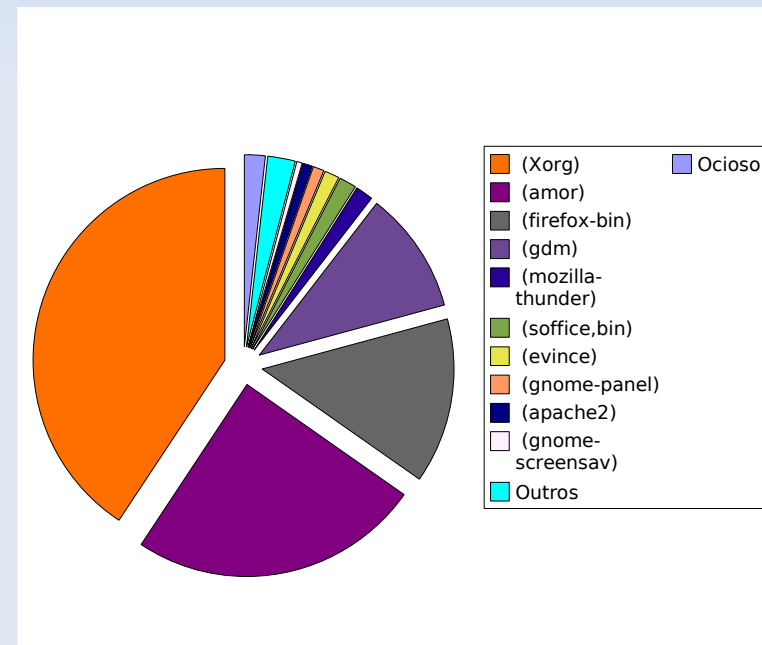
# Conclusão

- Uso do SDMR é viável
  - Dobrando o tempo de captura diminui-se o impacto causa na rede
- Contribuição
- Trabalhos futuros
  - Desenvolvimento do console
  - Implementar *threads* no coletor
  - Detecção automática de agentes
  - Captura de novas informações
  - Suporte a outros SOs

# Console

## ■ Utilização da CPU pelos processos

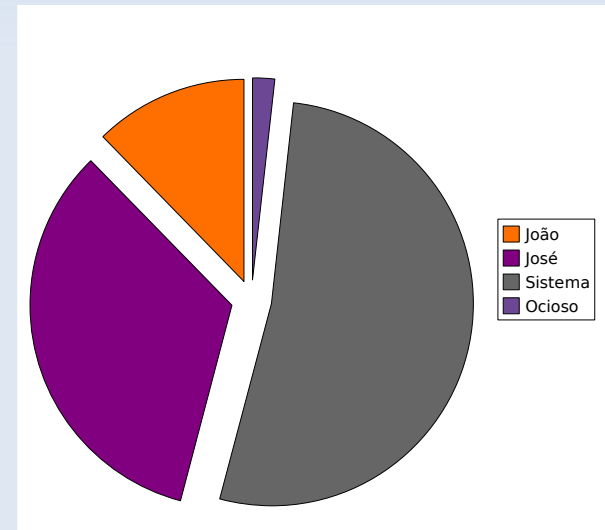
Processo	CPU
(Xorg)	40,78%
(amor)	24,57%
(firefox-bin)	13,75%
(gdm)	10,47%
(mozilla-thunder)	1,54%
(soffice,bin)	1,48%
(evince)	1,24%
(gnome-panel)	0,93%
(apache2)	0,83%
(gnome-screensav)	0,47%
Outros	2,27%
Ocioso	1,67%



# Console

- Utilização da CPU por usuário

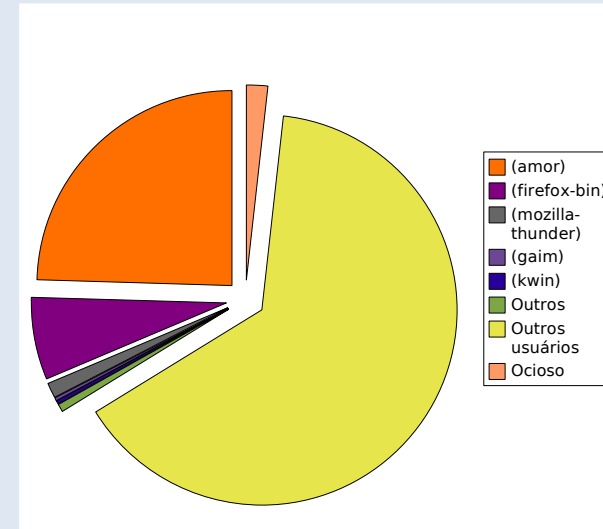
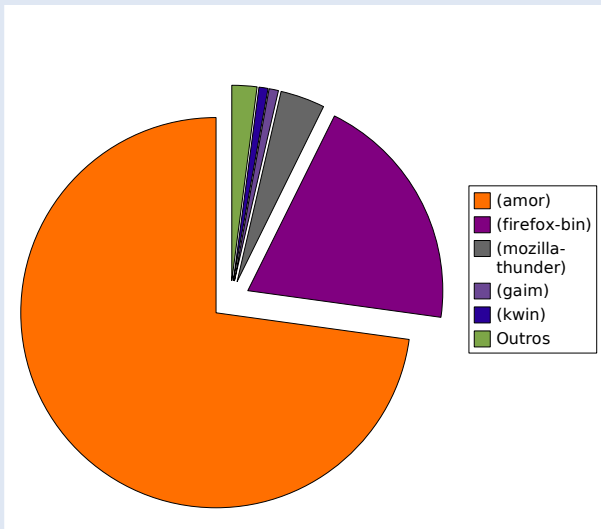
Usuário	CPU
João	12,31%
José	33,70%
Sistema	52,31%
Ocioso	1,67%



# Console

## ■ Utilização da CPU pelo usuário José

Processo	CPU
(amor)	24,57%
(firefox-bin)	6,70%
(mozilla-thunder)	1,25%
(gaim)	0,26%
(kwin)	0,25%
Outros	0,67%
Outros usuários	64,63%
Ocioso	1,67%



# Dúvidas?

SISTEMA DISTRIBUÍDO PARA MONITORAR O USO DOS RECURSOS  
DE HARDWARE E SOFTWARE EM ESTAÇÕES DE TRABALHO  
GNU/LINUX

Jamiel Spezia  
<jamiel@solis.coop.br>

Orientador: Maglan Cristiano Diemer

Lajeado, dezembro de 2007