Monitoramento de Memória e CPU utilizando séries temporais gerando assinatura de comportamento com médias móveis

Resumo:

1 Introdução

Não é incomum a existência de equipamentos que ficam determinados períodos ociosos, ou mesmo que não têm suas capacidades exploradas adequadamente. Isso pode ocorrer em datacenters, onde há equipamentos como servidores, mas também pode acontecer em estações de trabalho e computadores pessoais. Todo este recurso não utilizado tem seu valor e mensurá-lo pode trazer diversas vantagens seja para realocar demandas de processamento para equipamentos mais ociosos ou mesmo para nortear futuras aquisições, por exemplo auxiliando na compra de um novo servidor para que este tenha características adequadas ao que for demandado.

Neste projeto é proposto a criação de um sistema de monitoramento de equipamentos capturando informações do consumo de memória e CPU, onde a captura ocorre inicialmente localmente para posteriormente ser realizado o envio de uma assinatura comportamental deste equipamento utilizando a média móvel exponencial. O envio desta forma traz menos *overhead* para a rede, já que o envio ocorre em períodos maiores, além disso outra vantagem é que a visualização da média móvel exponencial mostra uma o comportamento do equipamento em determinado período e não somente um recorte do status atual da memória e CPU.

….

2 Funcionamento do monitor

Para cada equipamento monitorado foram desenvolvidos dois módulos chamados de monitor e transmissor. Além disso, há um servidor web, conforme a Figura 1, onde fica uma Rest (*Representational State Transfer*) API (*Application Programming Interface*) que é o canal de comunicação entre o módulo transmissor e o servidor de banco de dados que grava os dados dos comportamentos de todos os equipamentos monitorados. Ainda neste servidor web, se encontram disponíveis as apresentações dos dados coletados de todos os equipamentos tanto em forma gráfica como em forma de listas, sendo possível a aplicação de filtros e ordenações, tornando possível por exemplo visualizar os equipamentos monitorados com mais recursos disponíveis.

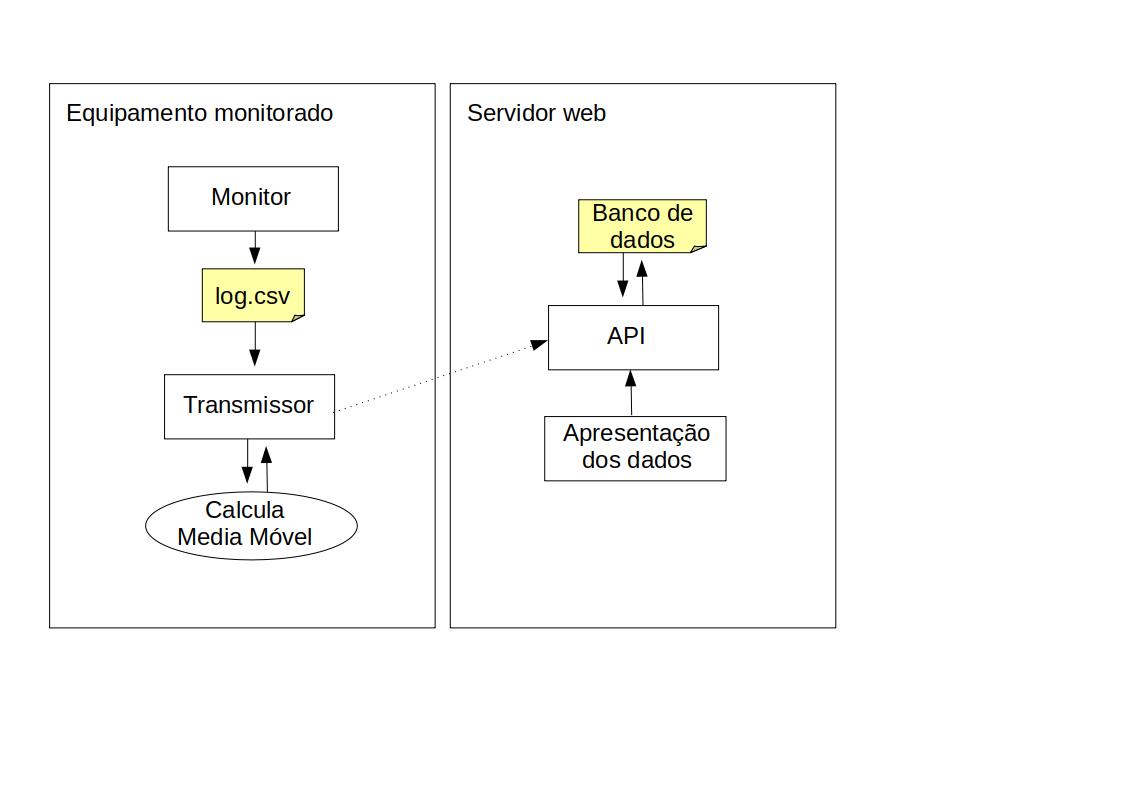


Figura 1: Módulos do sistema de monitoramento no equipamento local e no servidor.

Resumidamente, o monitor coleta a cada 30 segundos o percentual de consumo de memória e CPU e grava estes dados localmente em um arquivo CSV (*Comma-separated values*), sendo que neste arquivo também consta a data e hora em que foi realizada a coleta. Já o módulo transmissor utilizando o arquivo CSV gerado, faz o cálculo da média móvel exponencial, enviando o resultado a cada 1170 segundos para a API, que centralizará todos os dados de monitoramento. O transmissor também envia o nome do equipamento que está sendo monitorado, a data e hora atual. A seguir tem-se o detalhamento de cada um destes módulos.

Os módulos monitor e transmissor foram desenvolvidos com a linguagem Python. Além disso foram utilizadas diversas bibliotecas desta linguagem para auxiliar tarefas, como o Pandas para realizar o cálculo da média móvel exponencial, a biblioteca Json para serializar informações de envio à API, Socket para verificar o nome do host que envia as informações e Requests que fornece funções para trabalhar com requisições HTTP (*Hypertext Transfer Protocol*).

No servidor também foi utilizada a linguagem Python em conjunto com o Framework Django, que provê o arcabouço de gravação no banco de dados PostgreSql, além de também fornecer funções de recebimento e serialização das requisições do Rest API.

2.1 Monitor

O módulo monitor é quem realiza a primeira etapa onde é gravado em um arquivo CSV as informações capturadas em tempo real do equipamento em um intervalo de 30 segundos. Ou seja, a cada 30 segundos as informações de percentual de consumo de memória e CPU são capturadas e gravadas em arquivo. É importante ressaltar que esse módulo é executado localmente, não havendo nenhuma interação com o servidor. O formato do arquivo gerado é simples, sendo que neste constam três colunas: data/hora, consumo de memória e consumo de CPU, conforme a Figura 2:

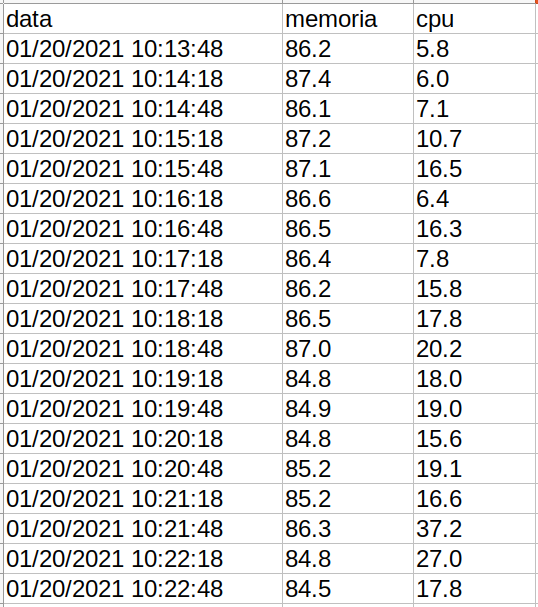


Figura 2: Trecho de um arquivo CSV gerado pelo monitor.

2.2 Transmissor

O transmissor, a cada intervalo de 39 períodos de captura do monitor, ou seja, 1170 segundos, faz a leitura do arquivo CSV gerado e calcula a média móvel exponencial. Posteriormente esta informação é transmitida para a Rest API do servidor web, que recebe estes dados e grava os em banco de dados. Neste momento além do resultado do cálculo, é enviado o nome do equipamento, pois somente desta forma será possível mais tarde distinguir a quem pertence as informações.

2.3 Rest API

Uma API, é um conjunto de rotinas e padrões estabelecidos por uma aplicação A, para que outras aplicações consigam utilizar as funcionalidades da aplicação A, sem precisar conhecer detalhes da implementação do software.

Ou seja, as APIs permitem uma interoperabilidade entre aplicações. Em outras palavras, a comunicação entre aplicações e entre os usuários.

Rest é um conjunto de restrições utilizadas para que as requisições HTTP atendam as diretrizes definidas na arquitetura. Basicamente, as restrições determinadas pela arquitetura Rest são:

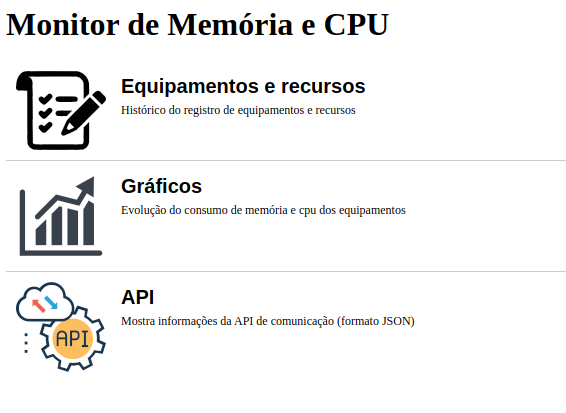
* cliente-servidor: as aplicações existentes no servidor e no cliente devem ser separadas;
* sem estado: as requisições são feitas de forma independente, ou seja, cada uma executa apenas uma determinada ação;
* cache: a API deve utilizar o cache para evitar chamadas recorrentes ao servidor;
* interface uniforme: agrupa outros quatro conceitos em que determina que os recursos devem ser identificados, a manipulação dos recursos deve ser por meio de representação, com mensagens autodescritivas e utilizar links para navegar pelo aplicativo.

Portanto, quando se fala em Rest API, significa utilizar uma API para acessar aplicações back-end, de modo que essa comunicação seja feita com os padrões definidos pelo estilo de arquitetura Rest.

As requisições HTTP realizadas em uma Rest API normalmente retornam dados no formato JSON. Existem outros formatos possíveis de retorno, como o XML, entretanto, o JSON (JavaScript Object Notation) é o mais utilizado, sendo este o formato utilizado neste projeto.

2.4 Apresentação dos dados

No módulo web, onde ocorre a apresentação dos dados, inicialmente são mostradas três opções, como pode se ver na Figura 3.

Figura 3: Menu inicial do módulo de apresentação dos dados.

Na primeira opção é mostrada uma listagem dos dados enviados para o servidor. Nesta interface é possível realizar filtros por equipamento e organizar a listagem por CPU e memória, conforme a Figura 4. Dessa forma é possível visualizar quais equipamentos estão consumindo mais recursos.

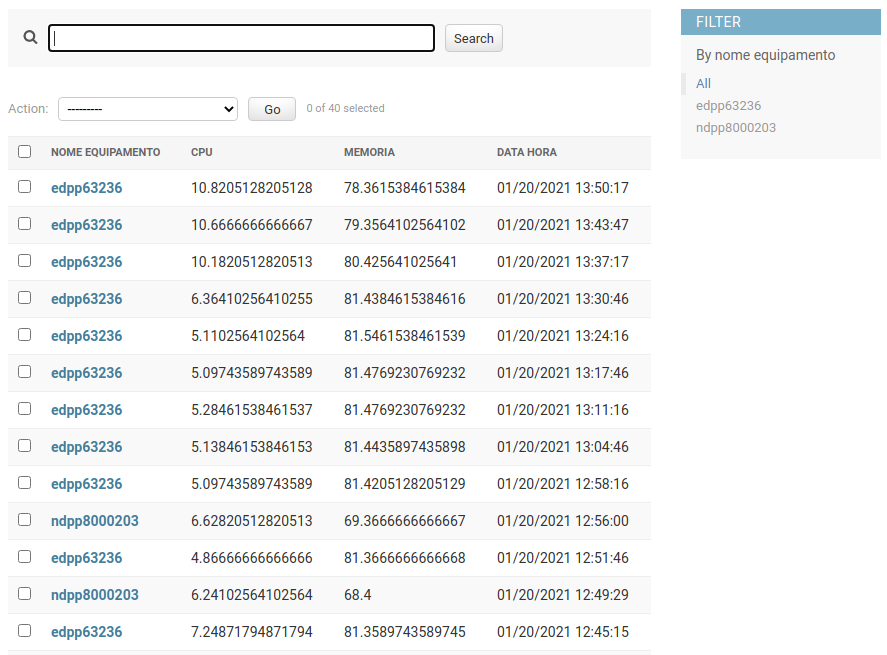


Figura 4: Listagem dos dados enviados para o servidor por todos os equipamentos.

Na próxima opção é possível verificar a evolução de determinado equipamento através da projeção de um gráfico que demonstra o comportamento dos recursos de CPU e memória de cada equipamento cadastrado, como demonstrado na Figura 5. Há uma tela prévia a esta, onde é necessário selecionar em uma lista qual equipamento se deseja visualizar o gráfico comportamental.

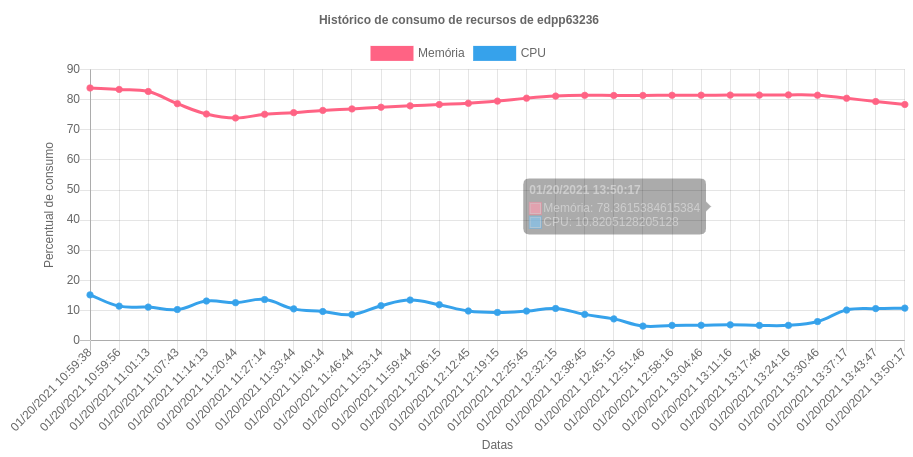


Figura 5: Evolução dos dados enviados por cada um dos equipamentos.

Na última opção é possível visualizar os equipamentos cadastrados na API de forma bruta, onde não houve qualquer processamento. Conforme a Figura 6, nesta opção a lista é mostrada no formato JSON.

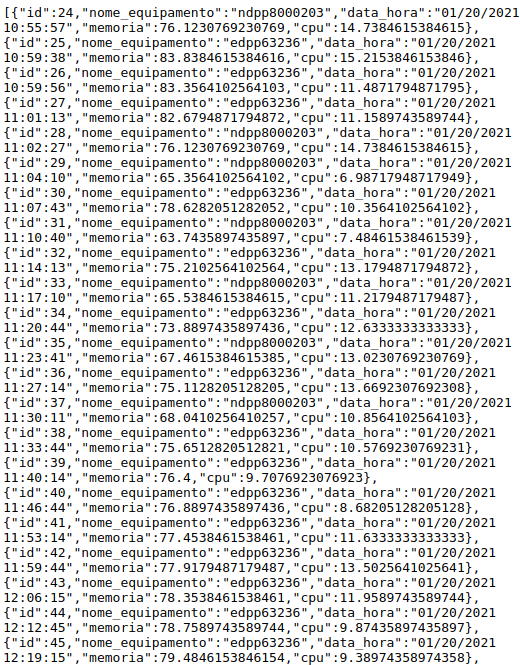


Figura 6: Informações dos recursos no formato JSON.

3 Método utilizado

Utilizamos a coleta de informações em um intervalo de 30 segundos porque….

Utilizamos a média móvel exponencial de 39 períodos devido a…

A carga adicional na rede e no equipamento monitorado é baixa, pois o envio ocorre a cada 19 minutos e 30 segundos, que é o intervalo de tempo necessário para que seja possível gerar a informação comportamental de uma média exponencial de 39 períodos.

5 Resultados

6 Conclusão