

Relatório de Monitoramento e Análise de Desempenho da Instância

Introdução

Este relatório apresenta uma análise do desempenho de uma instância virtualizada (VM) ao longo de uma hora. O objetivo é monitorar o uso de recursos (CPU, memória, disco, rede) e aplicar técnicas de inteligência artificial (IA) para identificar padrões de uso, prever o uso da CPU e detectar anomalias.

Os dados de uso de recursos foram coletados a cada segundo durante uma hora e armazenados em um arquivo CSV. Em seguida, foram aplicadas técnicas de IA, como regressão linear, detecção de anomalias (Isolation Forest) e clustering (K-Means) para analisar os dados.

Preparação do Ambiente

Antes de iniciar a coleta e análise dos dados, foi realizada a preparação do ambiente Python com os seguintes passos

Instalação das Dependências:

```
vboxuser@ubuntu:~$ python3 --version
Python 3.12.3
vboxuser@ubuntu:~$ pip3 install psutil pandas scikit-learn matplotlib seaborn
```

Criação e Ativação do Ambiente Virtual:

```
vboxuser@ubuntu:~$ cd monitoramento_vm
vboxuser@ubuntu:~/monitoramento_vm$ source env/bin/activate
(env) vboxuser@ubuntu:~/monitoramento_vm$ nano monitora_vm.py
(env) vboxuser@ubuntu:~/monitoramento_vm$ python monitora_vm.py
Iniciando a coleta de dados... Isso pode levar até 1 hora!
Coletados dados por 0 minuto(s)...
```

Criação do Script:

```
GNU nano 7.2          monitora_vm.py
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', ls='solid')
plt.title("Previsão de Uso de CPU vs. Valores Reais")
plt.xlabel("Valores Reais")
plt.ylabel("Previsões")
plt.grid(True)
plt.savefig('regressao_linear_cpu_1h.png')
plt.close()
print("Gráfico de Regressão Linear salvo como 'regressao_linear_cpu_1h.png'.")

# --- Detecção de Anomalias (Isolation Forest) ---
print("Iniciando a análise de Detecção de Anomalias...")
modelo_anomalias = IsolationForest(contamination=0.1) # 10% dos dados podem ser anomalias

# Treinando o modelo com os dados (excluindo o target 'CPU' por enquanto)
dados_analisados = df[['Memória', 'Disco', 'Bytes Enviados', 'Bytes Recebidos']]
modelo_anomalias.fit(dados_analisados)

# Prevendo as anomalias
anomalias = modelo_anomalias.predict(dados_analisados)
```

Código completo no arquivo `monitora_vm.py`

Coleta de Dados

A coleta foi realizada utilizando a biblioteca `psutil`, que permite monitorar recursos do sistema operacional. Foram capturadas, a cada segundo durante 1 hora (totalizando **3.600 amostras**), as seguintes métricas:

- **Uso de CPU (%)**
- **Uso de Memória RAM (%)**
- **Uso de Disco (%)**
- **Bytes Enviados pela Rede**
- **Bytes Recebidos pela Rede**

Esses dados foram organizados em um DataFrame do **Pandas** e salvos em um arquivo CSV: `dados_monitoramento_1h.csv`.

Regressão Linear: Previsão do Uso de CPU

Prever o uso de CPU com base nos outros recursos do sistema (Memória, Disco, Rede).

Técnica Utilizada:

- Algoritmo: LinearRegression da biblioteca **Scikit-Learn**.
- Variáveis independentes: Memória, Disco, Bytes Enviados, Bytes Recebidos.

- Variável dependente (target): CPU.

Procedimento:

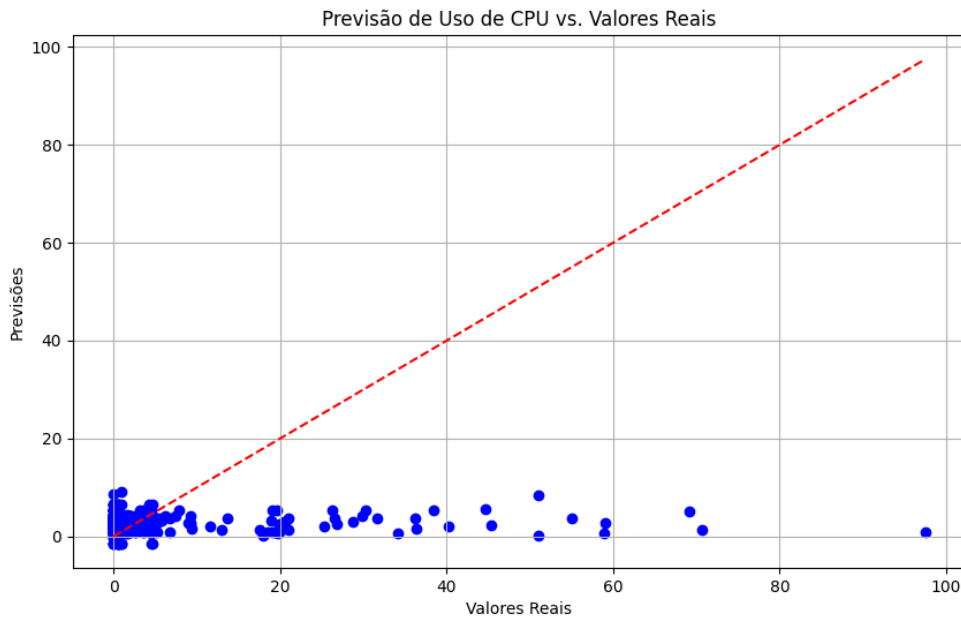
Divisão dos dados: 80% treino / 20% teste.

Treinamento do modelo de regressão.

- **R²**: Indica o quão bem as variáveis explicam o uso de CPU.
- **MSE**: Reflete o erro médio das previsões.

```
Coletados dados por 42 minuto(s)...
Coletados dados por 43 minuto(s)...
Coletados dados por 44 minuto(s)...
Coletados dados por 45 minuto(s)...
Coletados dados por 46 minuto(s)...
Coletados dados por 47 minuto(s)...
Coletados dados por 48 minuto(s)...
Coletados dados por 49 minuto(s)...
Coletados dados por 50 minuto(s)...
Coletados dados por 51 minuto(s)...
Coletados dados por 52 minuto(s)...
Coletados dados por 53 minuto(s)...
Coletados dados por 54 minuto(s)...
Coletados dados por 55 minuto(s)...
Coletados dados por 56 minuto(s)...
Coletados dados por 57 minuto(s)...
Coletados dados por 58 minuto(s)...
Coletados dados por 59 minuto(s)...
Coleta de dados concluída. Criando o DataFrame...
Dados salvos em 'dados_monitoramento_1h.csv'.
Iniciando a análise de Regressão Linear...
R²: -0.0155
Erro Médio Quadrático (MSE): 78.0813
Gráfico de Regressão Linear salvo como 'regressao_linear_cpu_1h.png'.
Iniciando a análise de Detecção de Anomalias...
Gráfico de Detecção de Anomalias salvo como 'deteccao_anomalias_1h.png'.
Iniciando a análise de Clustering com K-Means...
Gráfico de Clustering salvo como 'clustering_kmeans_1h.png'.
Gerando o gráfico de Histórico de Uso de Recursos...
Gráfico de Histórico de Uso de Recursos salvo como 'historico_uso_recursos_1h.png'.
Análise concluída! Todos os gráficos foram salvos com sucesso.
```

Saída Gráfica:



Ajuste Ruim: Conforme o relatório anterior, o R^2 próximo de zero e o MSE elevado indicam que o modelo de regressão linear não é capaz de capturar a relação entre as métricas de uso de recursos (memória, disco, rede) e o uso da CPU.

Visualização: O gráfico mostra uma dispersão aleatória dos pontos, sem uma relação clara entre os valores reais e os previstos. A linha vermelha (que representa a previsão perfeita) não se ajusta aos dados.

Deteção de Anomalias: Isolation Forest

Detectar padrões incomuns no comportamento dos recursos, especialmente para identificar possíveis falhas, picos inesperados ou comportamentos anômalos.

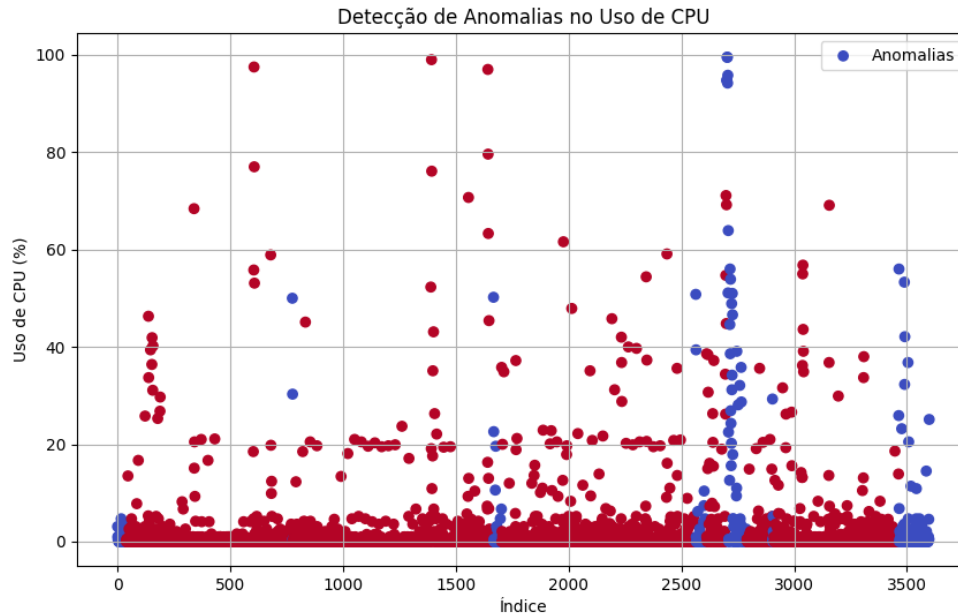
Técnica Utilizada:

Algoritmo: IsolationForest, ideal para detecção de outliers em grandes volumes de dados.

Procedimento:

- Treinamento do modelo com todas as variáveis exceto o uso de CPU.
- Previsão dos pontos anômalos (label -1).
- Associação da detecção ao DataFrame.

Saída Gráfica:



Identificação de Picos: O gráfico mostra uma série de pontos vermelhos (anomalias) espalhados ao longo do tempo, especialmente em momentos de alta utilização da CPU.

Picos de Carga: Muitas das anomalias parecem coincidir com picos de uso da CPU. Isso sugere que o Isolation Forest está identificando esses picos como desvios do comportamento normal.

Comportamentos Incomuns: Algumas anomalias ocorrem em momentos de baixa utilização da CPU. Estes podem indicar problemas diferentes, como processos em segundo plano que estão consumindo recursos inesperadamente ou padrões anormais de acesso ao disco ou à rede.

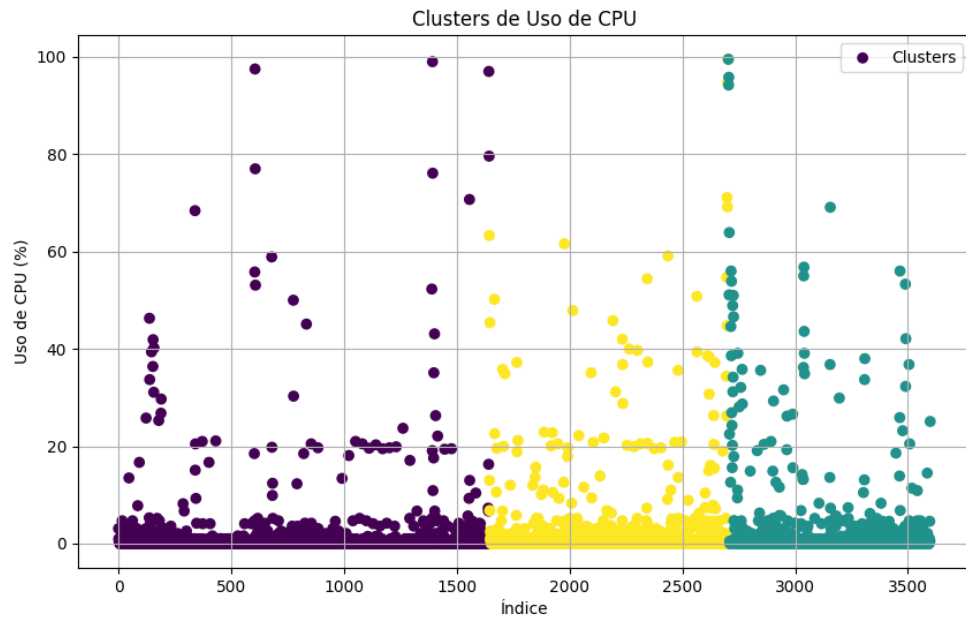
Clustering: Agrupamento com K-Means

Agrupar padrões semelhantes de uso de recursos, com base na similaridade entre os dados.

Procedimento:

- Aplicação de K-Means nas variáveis de entrada (Memória, Disco, Bytes Enviados, Bytes Recebidos).
- Classificação de cada amostra em um dos três clusters.
- Visualização da variação de uso de CPU conforme os agrupamentos.

Saída Gráfica:



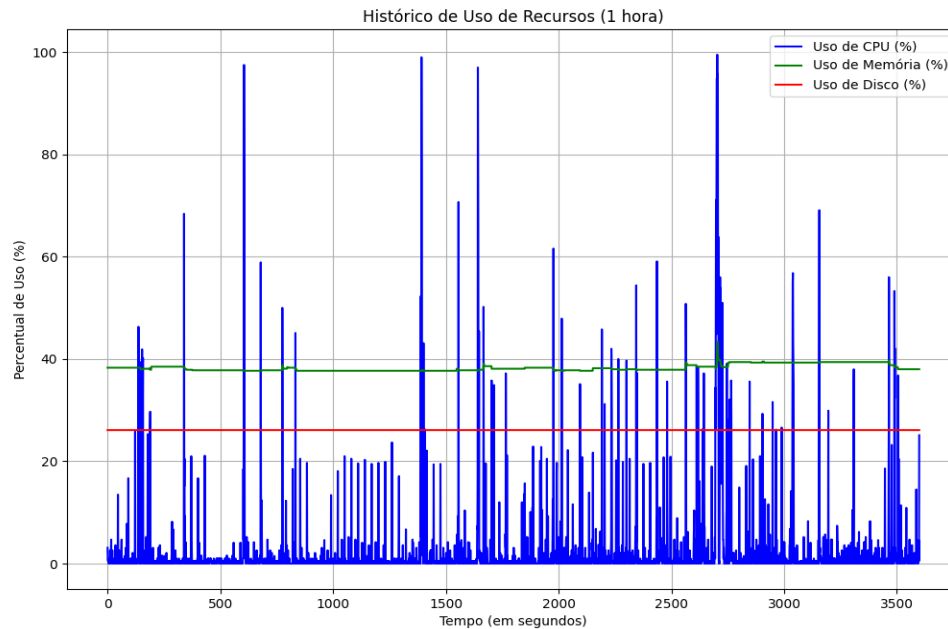
Segmentação do Uso da CPU: O K-Means agrupou os dados em três clusters distintos, que podem representar diferentes perfis de carga de trabalho:

- Cluster Roxo: Representa períodos de baixa utilização da CPU.
- Cluster Amarelo: Representa períodos de utilização moderada da CPU.
- Cluster Verde: Representa períodos de alta utilização da CPU.

Histórico de Uso de Recursos

Exibir a evolução temporal dos recursos durante o período monitorado.

Saída Gráfica:



Variação da CPU: O gráfico mostra que o uso da CPU varia significativamente ao longo do tempo, com picos frequentes.

Estabilidade da Memória e do Disco: O uso da memória e do disco permanece relativamente estável, o que sugere que a CPU é o gargalo de desempenho mais significativo na VM.

Correlação: Não parece haver uma correlação clara entre o uso da CPU e o uso da memória ou do disco. Picos de CPU nem sempre estão associados a mudanças no uso da memória ou do disco.

Conclusão

A análise realizada, combinando diferentes técnicas de Inteligência Artificial, forneceu uma visão detalhada sobre o desempenho da VM monitorada. A **regressão linear**, que tentava prever o uso de CPU com base em variáveis como memória e tráfego de rede, não foi eficaz. Isso sugere que o modelo precisa de ajustes, como a inclusão de novas variáveis ou o uso de modelos mais complexos, como redes neurais ou árvores de decisão. Por outro lado, a **detecção de anomalias** conseguiu identificar momentos em que o sistema estava fora do padrão, o que pode indicar falhas ou picos inesperados. Ao combinar isso com a técnica de **clustering (K-Means)**, vimos que as anomalias estavam frequentemente associadas a comportamentos atípicos, fora dos padrões definidos pelos clusters.

Em resumo, enquanto a combinação de técnicas proporcionou uma boa visão geral do comportamento do sistema, ficou claro que a previsão do uso de CPU precisa de modelos mais robustos. A detecção de anomalias e o clustering foram úteis para identificar padrões e momentos críticos, e essas áreas devem ser exploradas mais a fundo para melhorar a análise e otimizar o desempenho da VM.