

# Vazamento de memória - NodeJS

Um memory leak em Node.js ocorre quando o programa aloca memória para uso, mas não libera essa memória quando não é mais necessária. Isso pode acontecer devido a referências retidas acidentalmente ou ciclos de referência que impedem o coletor de lixo de limpar a memória não utilizada. Com o tempo, isso pode levar ao esgotamento dos recursos do sistema, causando lentidão ou falhas na aplicação.

## Monitoramento Inicial

Utilize ferramentas de monitoramento de recursos disponíveis no Linux para observar o comportamento da aplicação.

`top` e `htop`

Essas ferramentas permitem monitorar o uso de CPU e memória em tempo real.

```
top
```

```
htop
```

Observe se o processo do Node.js está consumindo mais memória com o tempo.

`free` e `vmstat`

Esses comandos fornecem informações sobre o uso de memória no sistema.

```
free -m
```

```
vmstat 1
```

Verifique se há um aumento constante na memória usada e se a swap está sendo utilizada.

## Identificação de Processos

Encontre o PID (Process ID) da aplicação Node.js.

```
ps aux | grep node
```

## Uso de `pmap`

O comando `pmap` exibe o mapa de memória do processo. Útil para ver quanto de memória virtual e física está sendo usada.

```
pmap -x <PID>
```

Substitua `<PID>` pelo PID encontrado no passo anterior.

## Ferramentas de Análise de Memória

### `smem`

O `smem` fornece uma visão detalhada do uso de memória por processo.

### Instalação:

```
sudo apt-get install smem
```

### Uso:

```
smem -p -c "pid user command swap uss pss rss"
```

### `valgrind`

O `valgrind` é uma ferramenta poderosa para detectar vazamentos de memória, mas pode impactar o desempenho da aplicação.

### Instalação:

```
sudo apt-get install valgrind
```

### Uso:

```
valgrind --tool=massif node app.js
```

Após a execução, o `valgrind` gerará um arquivo de perfil de memória que pode ser analisado para identificar vazamentos.

### Utilização de `node-heapdump`

Se houver a possibilidade de fazer alterações temporárias na aplicação, você pode utilizar o `heapdump` para capturar snapshots de heap.

### Instalação:

```
npm install heapdump
```

### Uso:

Adicione o seguinte código à aplicação para gerar um heapdump quando necessário:

```
const heapdump = require('heapdump');  
heapdump.writeSnapshot('/path/to/snapshot.heapsnapshot');
```

### Análise de Logs

Verifique os logs da aplicação e do sistema para quaisquer mensagens de erro ou advertência que possam indicar problemas de memória.

```
tail -f /var/log/syslog  
tail -f /var/log/messages
```