

Kubernetes Pod의 높은 CPU/메모리 사용률 문제 해결

목차

[소개](#)

[1. Pod에서 CPU/메모리가 높음 문제 알림](#)

[1.1. CPU에 대한 경고](#)

[1.2. 메모리에 대한 경고](#)

[2. Kubernetes 프로세스별 프로파일링](#)

[2.1. CPU 프로파일링\(/debug/pprof/profile\)](#)

[2.2. 메모리 프로파일링\(/debug/pprof/heap\)](#)

[2.3. Goroutine 프로파일링\(/debug/pprof/goroutine\)](#)

[2.4. Kubernetes 포드에서 포트 찾기](#)

[3. 시스템에서 수집할 자료](#)

[4. 수집된 프로파일 로그 출력 이해](#)

[4.1. 메모리 프로파일링\(/debug/pprof/heap\)에서 출력을 읽는 중](#)

[5. 그라파나](#)

[5.1. CPU 쿼리](#)

[5.2. 메모리 쿼리](#)

소개

이 문서에서는 SMF(Session Management Function) 또는 PCF(Policy Control Function)로 활용되는 CNDP(Cloud Native Deployment Platform) 플랫폼에서 CPU 또는 메모리 문제를 해결하는 방법에 대해 설명합니다.

1. Pod에서 CPU/메모리가 높음 문제 알림

이 문제 해결을 제대로 시작하려면 경고를 이해하는 것이 중요합니다. 사전 구성된 모든 기본 경고에 대한 설명은 [이 링크에 있습니다](#).

1.1. CPU에 대한 경고

여기서 트리거되는 활성 기본 경고는 다음과 같습니다 `k8s-pod-cpu-usage-high` .

이 Pod는 다음과 같은 Pod와 관련이 있습니다. `smf-udp-proxy-0` 컨테이너가 있습니다. `k8s_smf-udp-proxy_smf-udp-proxy-0_smf`

이 컨테이너가 네임스페이스에 있음을 알 수 있습니다. `smf`

```
alerts active detail k8s-pod-cpu-usage-high 36fbd5e0bbce
severity major
type "Processing Error Alarm"
startsAt 2024-02-23T12:45:44.558Z
source smf-udp-proxy-0
summary "Container: k8s_smf-udp-proxy_smf-udp-proxy-0_smf of pod: smf-udp-proxy-0 in namespace: smf has
Labels [ "name: k8s_smf-udp-proxy_smf-udp-proxy-0_smf" "namespace: smf" "pod: smf-udp-proxy-0" ]
```

Kubernetes 마스터에서 다음 명령을 입력하여 영향을 받은 Pod를 찾습니다.

```
master $ kubectl get pods smf-udp-proxy-0 -n smf
```

1.2. 메모리에 대한 경고

여기서 트리거되는 활성 기본 경고는 다음과 같습니다 container-memory-usage-high .

Pod와 관련된 Pod를 확인할 수 있습니다. grafana-dashboard-sgw-765664b864-zwxct 컨테이너가 있습니다 . k8s_istio-proxy_grafana-dashboard-sgw-765664b864-zwxct_smf_389290ee-77d1-4ff3-981d-58ea1c8eabdb_0

이 컨테이너는 네임스페이스에 있습니다.smf

```
alerts active detail container-memory-usage-high 9065cb8256ba
severity critical
type "Processing Error Alarm"
startsAt 2024-04-25T10:17:38.196Z
source grafana-dashboard-sgw-765664b864-zwxct
summary "Pod grafana-dashboard-sgw-765664b864-zwxct/k8s_istio-proxy_grafana-dashboard-sgw-765664b864-zw
Labels [ "alertname: container-memory-usage-high" "beta_kubernetes_io_arch: amd64" "beta_kubernetes_io_
annotations [ "summary: Pod grafana-dashboard-sgw-765664b864-zwxct/k8s_istio-proxy_grafana-dashboard-sgw
```

Kubernetes 마스터에서 다음 명령을 입력하여 영향을 받은 Pod를 찾습니다.

```
master $ kubectl get pods grafana-dashboard-sgw-765664b864-zwxct -n smf
```

2. Kubernetes 프로세스별 프로파일링

2.1. CPU 프로파일링(/debug/pprof/profile)

CPU 프로파일링은 실행 중인 Go 프로그램의 CPU 사용량을 캡처하고 분석하는 기술입니다.

이 기능은 호출 스택을 주기적으로 샘플링하고 정보를 기록하므로 프로그램에서 대부분의 시간을

보내는 위치를 분석할 수 있습니다.

2.2. 메모리 프로파일링(/debug/pprof/heap)

메모리 프로파일링은 Go 애플리케이션의 메모리 할당 및 사용 패턴에 대한 통찰력을 제공합니다. 메모리 누수를 식별하고 메모리 사용을 최적화하는 데 도움이 될 수 있습니다.

2.3. Goroutine 프로파일링(/debug/pprof/goroutine)

Goroutine 프로파일링은 스택 추적을 표시하여 모든 현재 Goroutines의 동작에 대한 통찰력을 제공합니다. 이 분석은 프로그램의 성능에 영향을 줄 수 있는 고착되거나 유출된 Goroutines를 식별하는 데 도움이 됩니다.

2.4. Kubernetes 포드에서 포트 찾기

명령을 사용합니다:

```
master:~$ kubectl describe pod <POD NAME> -n <NAMESPACE> | grep -i pprof
```

출력 예:

```
master:~$ kubectl describe pod udp-proxy-0 -n smf-rcdn | grep -i pprof
PPROF_EP_PORT: 8851
master:~$
```

3. 시스템에서 수집할 자료

문제 발생 시간 및 CEE(Common Execution Environment)에 대한 활성 경고 발생 시, 문제 발생 전후의 시간에 해당하는 데이터를 수집해 주십시오.

CEE:

```
cee# show alerts active detail
cee# show alerts history detail
cee# tac-debug-pkg create from yyyy-mm-dd_hh:mm:ss to yyyy-mm-dd_hh:mm:ss
```

CNDP 마스터 노드:

General information:

```
master-1:~$ kubectl get pods <POD> -n <NAMESPACE>
master-1:~$ kubectl pods describe <POD> -n <NAMESPACE>
master-1:~$ kubectl logs <POD> -n <NAMESPACE> -c <CONTAINER>
```

Login to impacted pod and check top tool:
master-1:~\$ kubectl exec -it <POD> -n <NAMESPACE> bash
root@protocol-n0-0:/opt/workspace# top

If pprof socket is enabled on pod:
master-1:~\$ kubectl describe pod <POD NAME> -n <NAMESPACE> | grep -i pprof
master-1:~\$ curl http://<POD IP>:<PPROF PORT>/debug/pprof/goroutine?debug=1
master-1:~\$ curl http://<POD IP>:<PPROF PORT>/debug/pprof/heap
master-1:~\$ curl http://<POD IP>:<PPROF PORT>/debug/pprof/profile?seconds=30

4. 수집된 프로파일 로그 출력 이해

4.1. 메모리 프로파일링(/debug/pprof/heap)에서 출력을 읽는 중

This line indicates that a total of 1549 goroutines were captured in the profile. The top frame (0x9207a9) shows that the function `google.golang.org/grpc.(*addrConn).resetTransport` is being executed, and the line number in the source code is `clientconn.go:1164`.

숫자(예: 200)로 시작하는 각 섹션은 Goroutine의 스택 추적을 나타냅니다.

```
goroutine profile: total 1549
200 @ 0x4416c0 0x415d68 0x415d3e 0x415a2b 0x9207aa 0x46f5e1
# 0x9207a9 google.golang.org/grpc.(*addrConn).resetTransport+0x6e9 /opt/workspace/gtpc-ep/pkg/
```

The first line in each section shows the number of goroutines with the same stack trace. For example, there are 200 goroutines with the same stack trace represented by memory addresses (0x4416c0, 0x415d68, and more.). The lines that start with # represent the individual frames of the stack trace. Each frame shows the memory address, function name, and the source code location (file path and line number) where the function is defined.

```
200 @ 0x4416c0 0x45121b 0x873ee2 0x874803 0x89674b 0x46f5e1
# 0x873ee1 google.golang.org/grpc/internal/transport.(*controlBuffer).get+0x121 /opt/workspace/g
# 0x874802 google.golang.org/grpc/internal/transport.(*loopyWriter).run+0x1e2 /opt/workspace/g
# 0x89674a google.golang.org/grpc/internal/transport.newHTTP2Client.func3+0x7a /opt/workspace/g

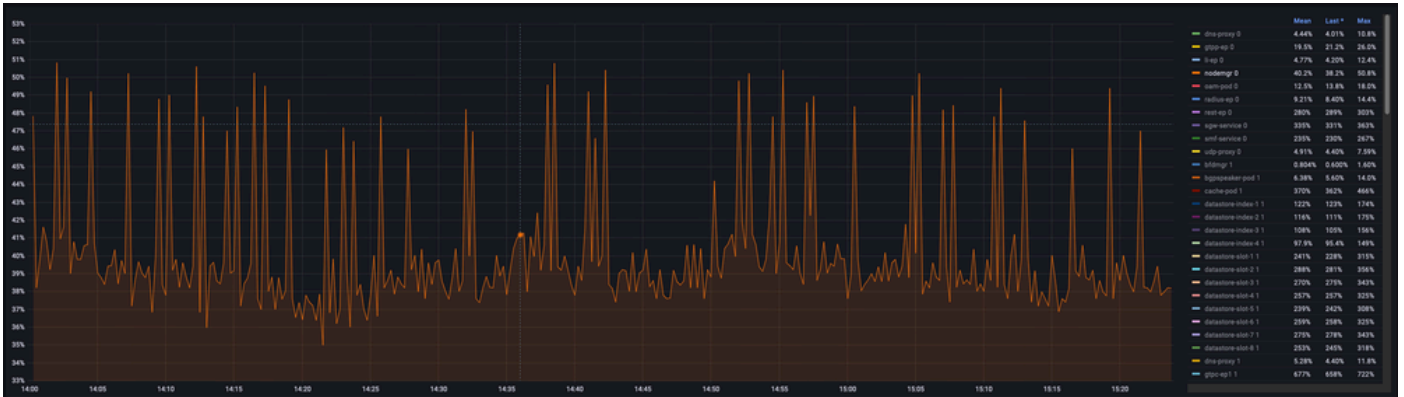
92 @ 0x4416c0 0x45121b 0x873ee2 0x874803 0x897b2b 0x46f5e1
# 0x873ee1 google.golang.org/grpc/internal/transport.(*controlBuffer).get+0x121 /opt/workspace/g
# 0x874802 google.golang.org/grpc/internal/transport.(*loopyWriter).run+0x1e2 /opt/workspace/g
# 0x897b2a google.golang.org/grpc/internal/transport.newHTTP2Server.func2+0xca /opt/workspace/g
```

5. 그래파나

5.1. CPU 쿼리

```
sum(cpu_percent{service_name=~"[[microservice]]"}) by (service_name,instance_id)
```

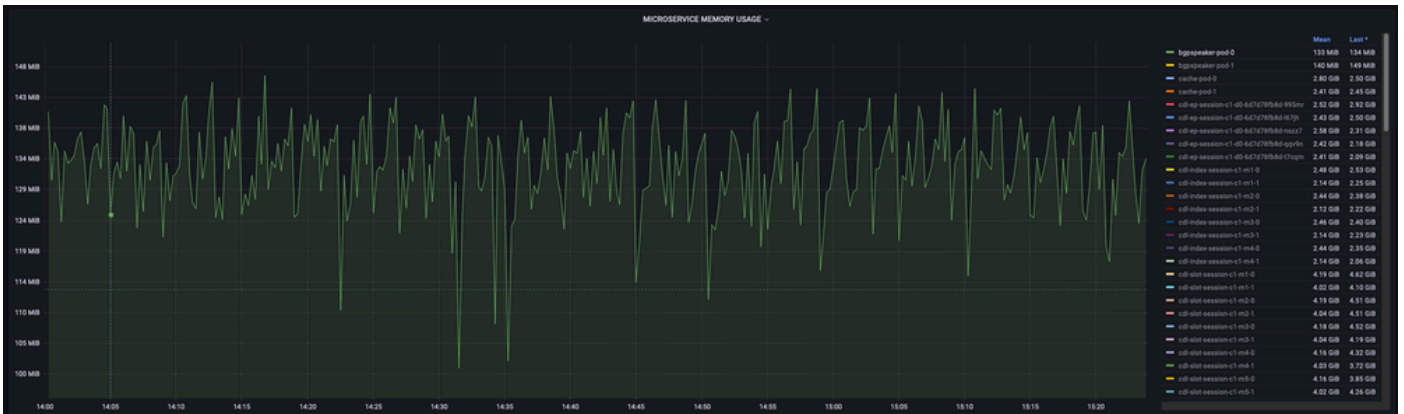
예:



5.2. 메모리 쿼리

```
sum(increase(mem_usage_kb{service_name=~"[[microservice]]"}[15m])) by (service_name,instance_id)
```

예:



이 번역에 관하여

Cisco는 전 세계 사용자에게 다양한 언어로 지원 콘텐츠를 제공하기 위해 기계 번역 기술과 수작업 번역을 병행하여 이 문서를 번역했습니다. 아무리 품질이 높은 기계 번역이라도 전문 번역가의 번역 결과물만큼 정확하지는 않습니다. Cisco Systems, Inc.는 이 같은 번역에 대해 어떠한 책임도 지지 않으며 항상 원본 영문 문서(링크 제공됨)를 참조할 것을 권장합니다.