

Risoluzione dei problemi relativi all'utilizzo elevato di CPU e memoria sui dispositivi Kubernetes

Sommario

[Introduzione](#)

[1. Avviso di problema: CPU/memoria elevata sul pod](#)

[1.1. Avviso per la CPU](#)

[1.2. Avviso per la memoria](#)

[2. Profilatura per processo Kubernetes](#)

[2.1. Profilatura CPU \(/debug/prof/profile\)](#)

[2.2. Creazione profilo memoria \(/debug/prof/heap\)](#)

[2.3. Creazione di profili di routine \(/debug/prof/goroutine\)](#)

[2.4. Trova porta prof su un pod Kubernetes](#)

[3. Dati da raccogliere dal sistema](#)

[4. Informazioni sugli output del log prof raccolto](#)

[4.1. Lettura dell'output dalla profilatura della memoria \(/debug/prof/heap\)](#)

[5. Grafana](#)

[5.1. Query sulla CPU](#)

[5.2. Query su memoria](#)

Introduzione

In questo documento viene descritto come risolvere i problemi relativi alla CPU o alla memoria sulla piattaforma Cloud Native Deployment Platform (CNDP) utilizzata come SMF (Session Management Function) o PCF (Policy Control Function).

1. Avviso di problema: CPU/memoria elevata sul pod

La comprensione dell'avviso è importante per una corretta risoluzione del problema. In [questo collegamento](#) viene fornita una spiegazione di tutti gli avvisi predefiniti preconfigurati.

1.1. Avviso per la CPU

In questo caso, è presente un avviso predefinito attivo denominato `k8s-pod-cpu-usage-high`.

Si vede che è correlato a un baccello denominato: `smf-udp-proxy-0` ed è un contenitore: `k8s_smf-udp-proxy_smf-udp-proxy-0_smf`

Il contenitore è incluso nello spazio dei nomi: `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" ]
```

Sul dispositivo master Kubernetes, individuare il pod interessato immettendo questo comando:

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

1.2. Avviso per la memoria

In questo caso, è presente un avviso predefinito attivo denominato `container-memory-usage-high`.

Si può vedere che è correlato a un baccello denominato: `grafana-dashboard-sgw-765664b864-zwxct` ed è un contenitore: `k8s_istio-proxy_grafana-dashboard-sgw-765664b864-zwxct_smf_389290ee-77d1-4ff3-981d-58ea1c8eabdb_0`

Contenitore nello spazio dei nomi: `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-zwxct"
labels [ "alertname: container-memory-usage-high" "beta_kubernetes_io_arch: amd64" "beta_kubernetes_io_c"
annotations [ "summary: Pod grafana-dashboard-sgw-765664b864-zwxct/k8s_istio-proxy_grafana-dashboard-sgw-
```

Sul dispositivo master Kubernetes, individuare il pod interessato immettendo questo comando:

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

2. Profilatura per processo di Kubernetes

2.1. Profilatura CPU (`/debug/prof/profile`)

Il profiling della CPU serve come tecnica per acquisire e analizzare l'utilizzo della CPU di un programma Go in esecuzione.

Il programma campiona periodicamente lo stack di chiamate e registra le informazioni, consentendo di analizzare il punto in cui il programma impiega la maggior parte del tempo.

2.2. Creazione profilo memoria (`/debug/prof/heap`)

La profilatura della memoria fornisce informazioni dettagliate sull'allocazione della memoria e sui modelli di utilizzo nell'applicazione Go.

Consente di identificare le perdite di memoria e di ottimizzare l'utilizzo della memoria.

2.3. Creazione di profili di routine (/debug/prof/goroutine)

L'analisi della linea di base fornisce informazioni dettagliate sul comportamento di tutte le linee correnti visualizzandone le tracce dello stack. Questa analisi consente di identificare le Goroutine bloccate o che perdono e che possono influire sulle prestazioni del programma.

2.4. Trova porta prof su un pod Kubernetes

Comando:

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

Output di esempio:

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

3. Dati da raccogliere dal sistema

Durante il periodo di emissione del problema e di attivazione dell'avviso in Common Execution Environment (CEE), raccogliere i dati relativi al tempo prima, durante e dopo il problema:

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
```

Nodo master CNDP:

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. Informazioni sugli output del log prof raccolto

4.1. Lettura dell'output dalla profilatura della memoria (/debug/prof/heap)

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`.

Ogni sezione che inizia con un numero (ad esempio, 200) rappresenta una traccia dello stack di una routine.

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

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

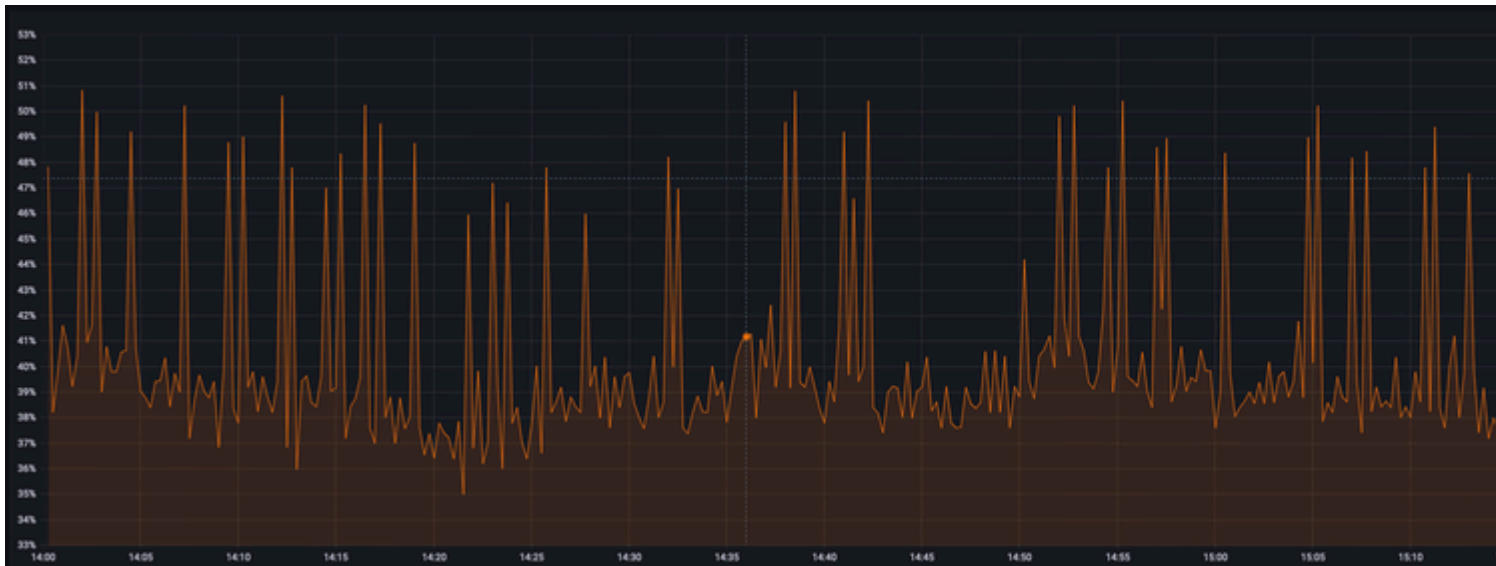
92 @ 0x4416c0 0x45121b 0x873ee2 0x874803 0x897b2b 0x46f5e1
# 0x873ee1 google.golang.org/grpc/internal/transport.(*controlBuffer).get+0x121 /opt/workspace/g
# 0x874802 google.golang.org/grpc/internal/transport.(*loopyWriter).run+0x1e2 /opt/workspace/g
# 0x897b2a google.golang.org/grpc/internal/transport.newHTTP2Server.func2+0xca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

5. Grafana

5.1. Query sulla CPU

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

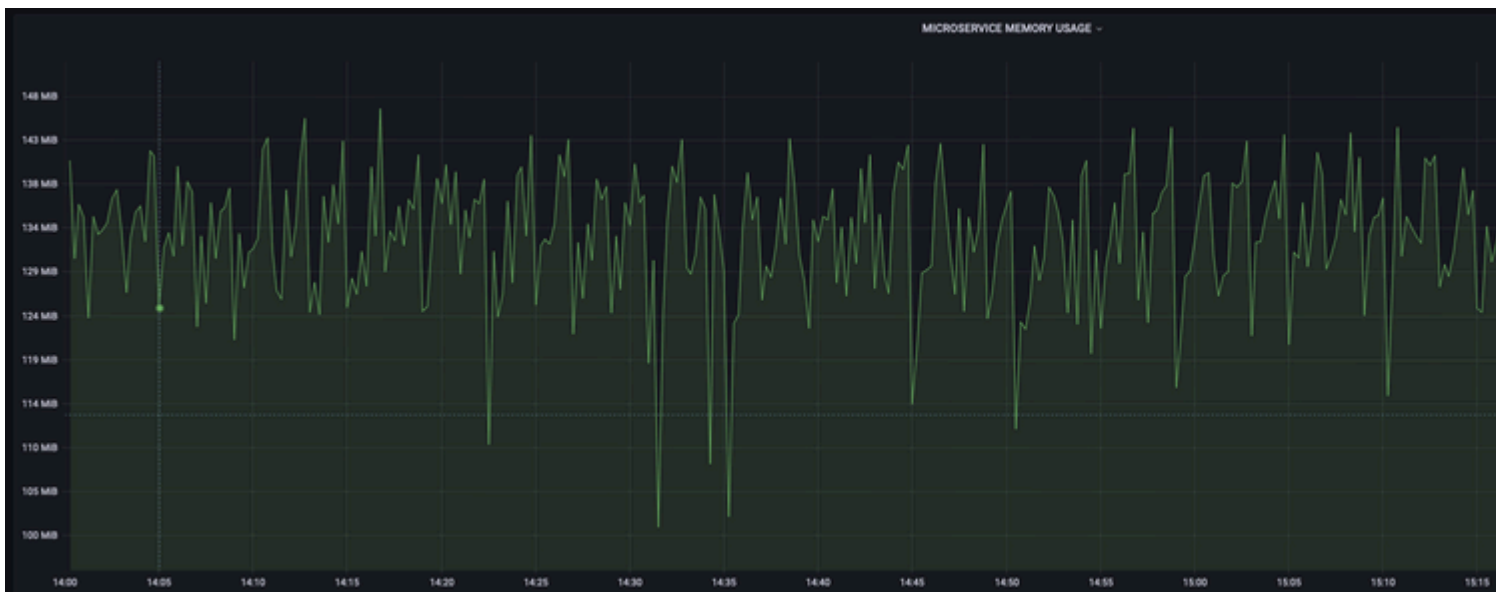
Esempio:



5.2. Query su memoria

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

Esempio:



Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).