



Go GC Settings

Bryan Boreham (@bboreham)



Hi, I'm Bryan Boreham

At Weaveworks, I work on Open Source DevOps tools doing system visualisation, monitoring and CI/CD

I also contribute to Container Network Interface, Kubernetes, Prometheus

Program optimisation is my video-game.

 @bboreham

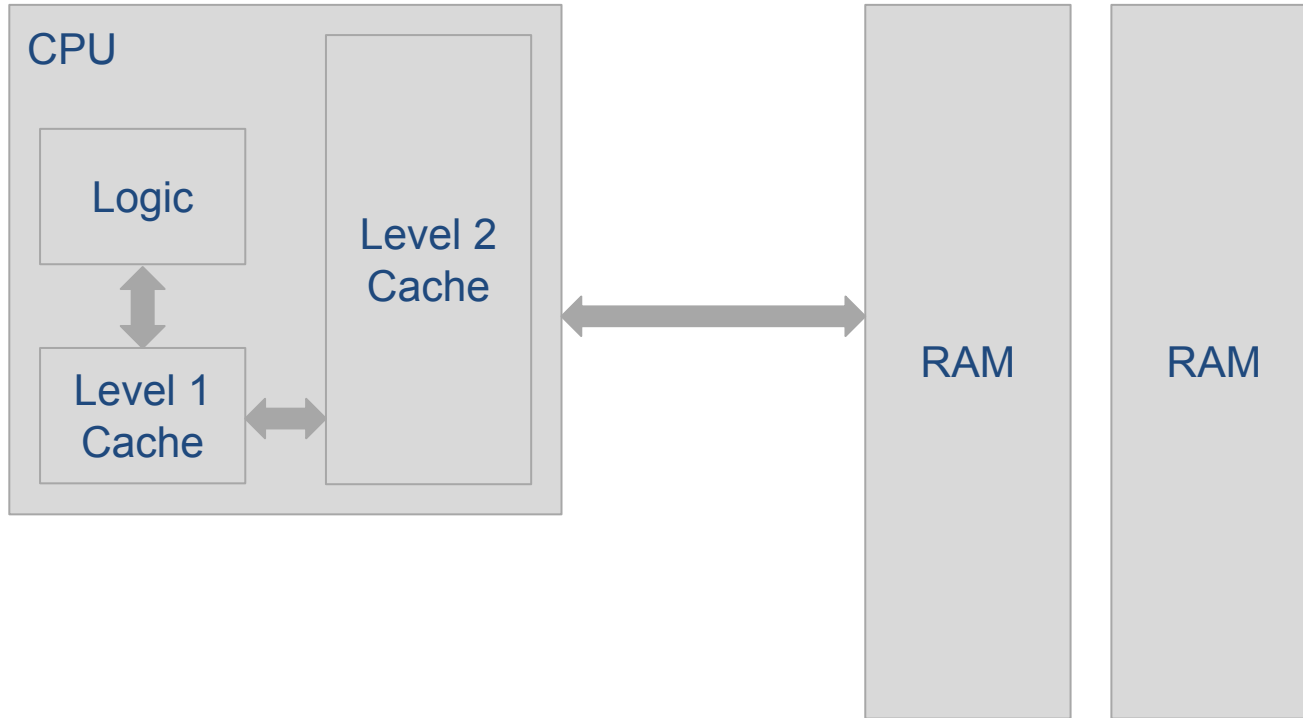
Why care about Garbage?

(this is the point where Bryan moans about never having enough RAM)

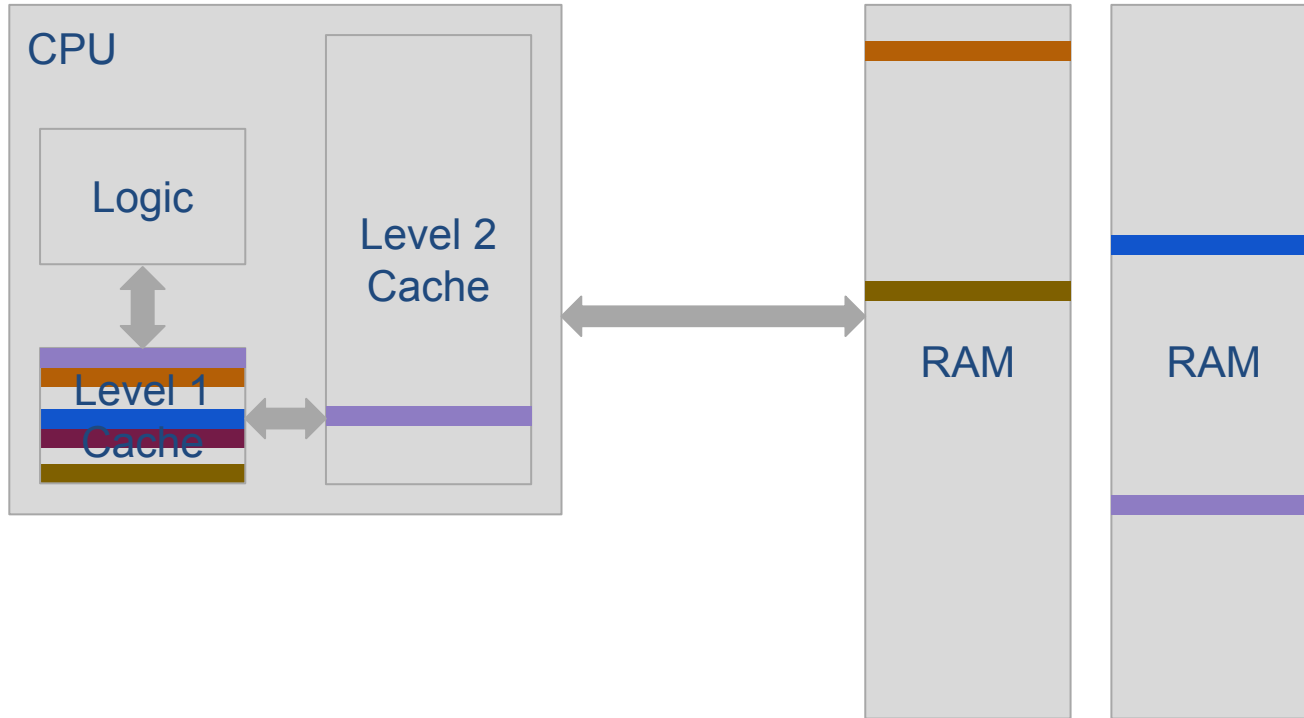


<https://www.redbubble.com/people/clgart/>

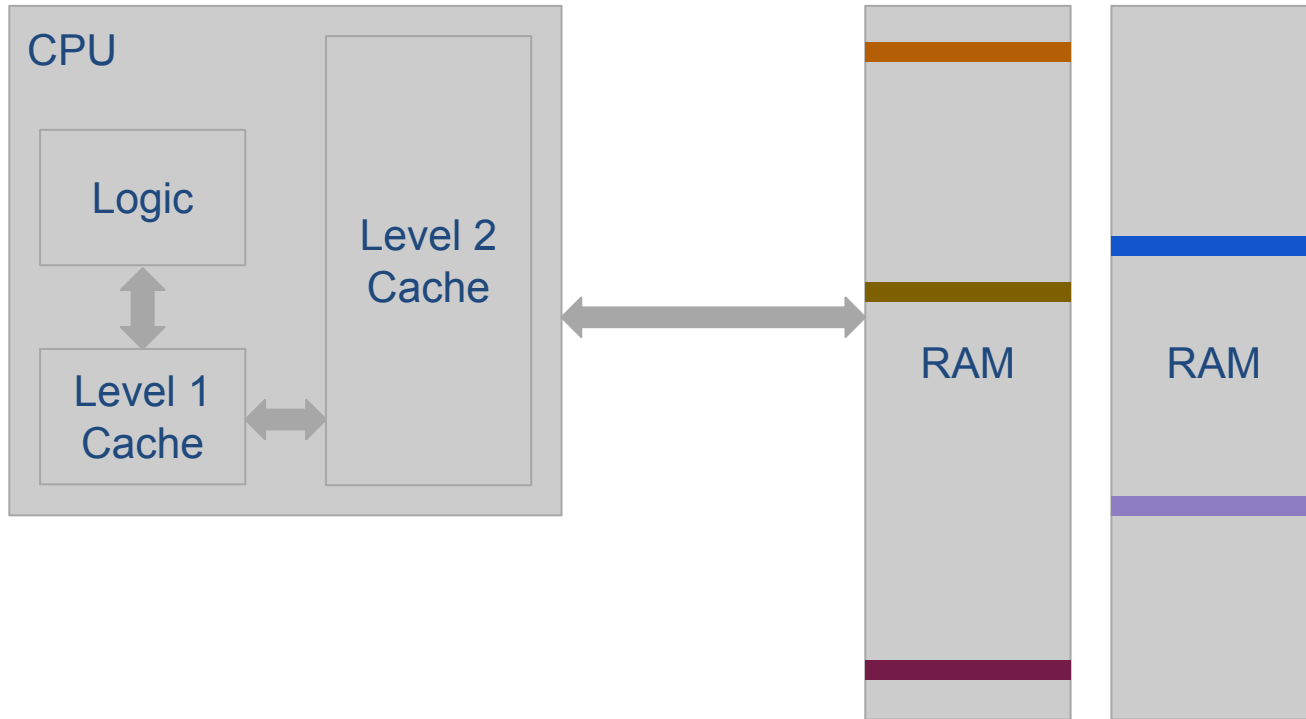
CPU memory architecture



Caches in use



After GC has run



Stack vs Heap



http://www.clipartpanda.com/clipart_images/stack-files-max-39545601



Photo: JohnNyberg, rgbstock.com

GC Options

-Xmsn

-Xmxn

-XX:-UseConcMarkSweepGC

-XX:-UseParallelGC

-XX:-UseParallelOldGC

-XX:-UseSerialGC

-XX:+UseG1GC

-XX:+UseGCOverheadLimit

-XX:MaxGCPauseMillis=n

-XX:InitiatingHeapOccupancyPercent=n

-XX:MaxTenuringThreshold=n

-XX:+ScavengeBeforeFullGC

-XX:ParallelGCThreads=n

-XX:ConcGCThreads=n

-XX:G1ReservePercent=n

-XX:G1HeapRegionSize=n

-XX:MaxHeapFreeRatio=n

-XX:MaxNewSize=size

-XX:MaxPermSize=n

-XX:MinHeapFreeRatio=n

-XX:NewSize=n

-XX:NewRatio=n

-XX:SurvivorRatio=n

-XX:TargetSurvivorRatio=n

-XX:+HandlePromotionFailure

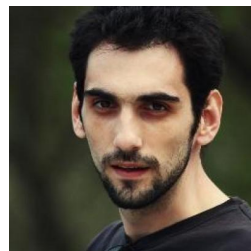
Go GC Options

GOGC

Test set-up

Server: github.com/stefanprodan/k8s-podinfo

- microservice skeleton



Client: github.com/rakyll/hey

- HTTP load generator

```
hey -z 2m -c 10 -q 20 ".../metrics"
```

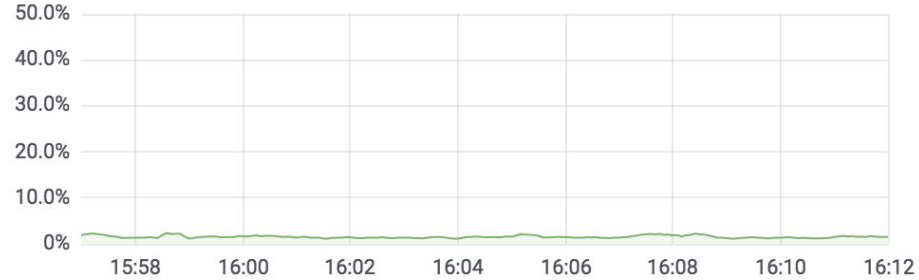


Situation: large stable data set

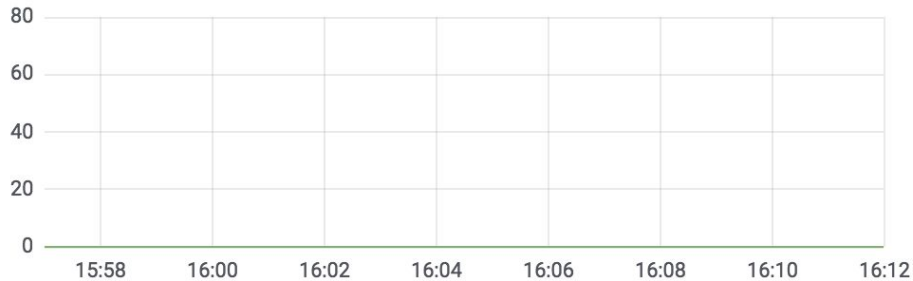
Heap size



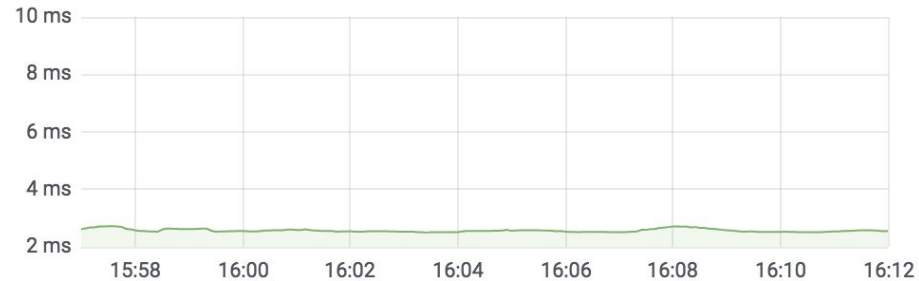
CPU usage



GC Rate

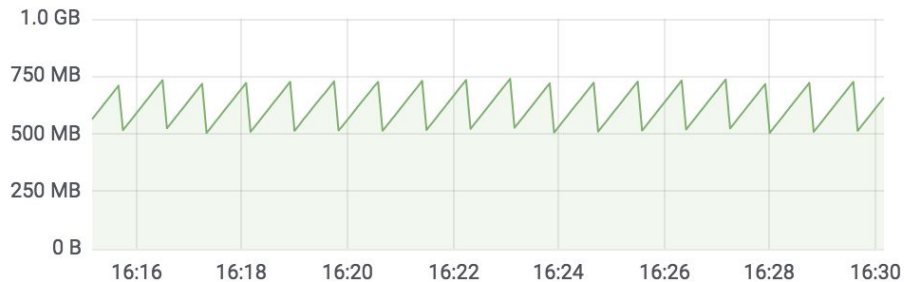


Latency



Lower GOGC to save RAM

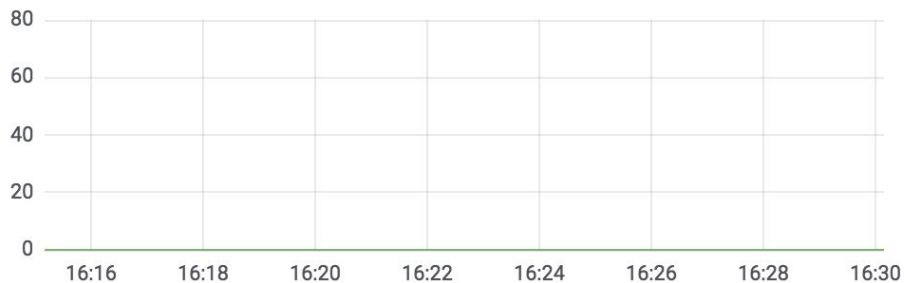
Heap size



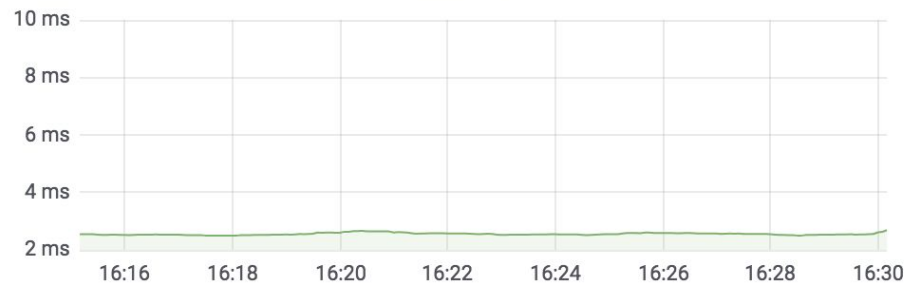
CPU usage



GC Rate

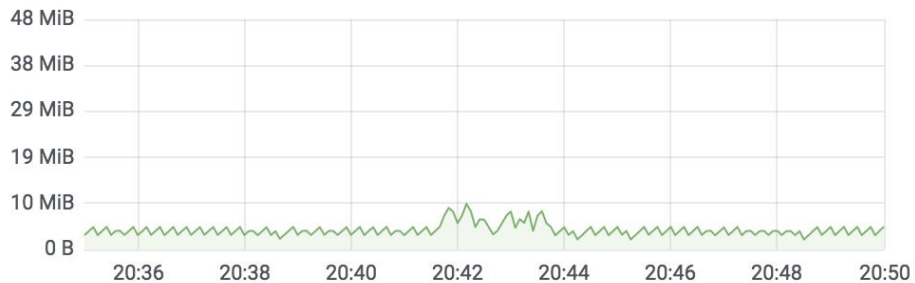


Latency

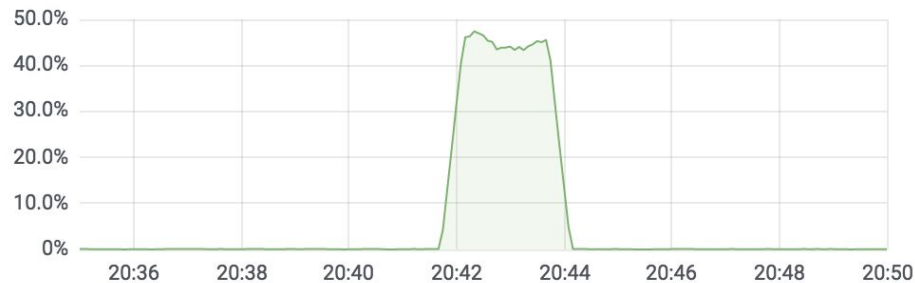


Situation: high GC rate, small heap

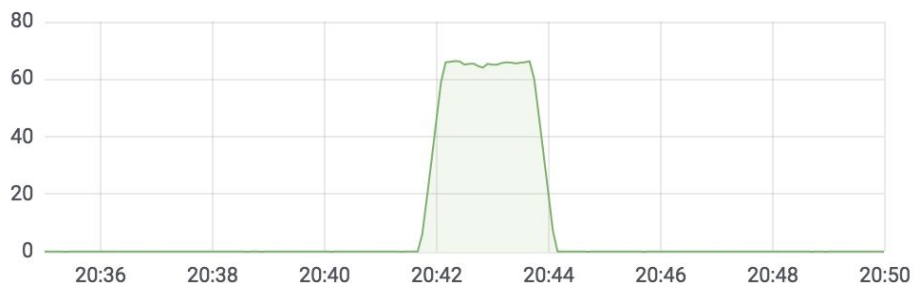
Heap size



CPU usage



GC Rate

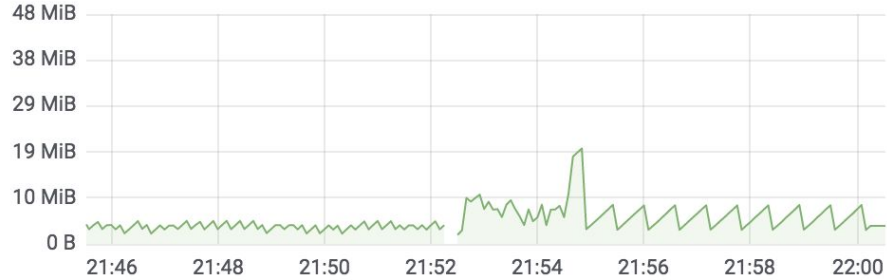


Latency



Raise GOGC to save CPU

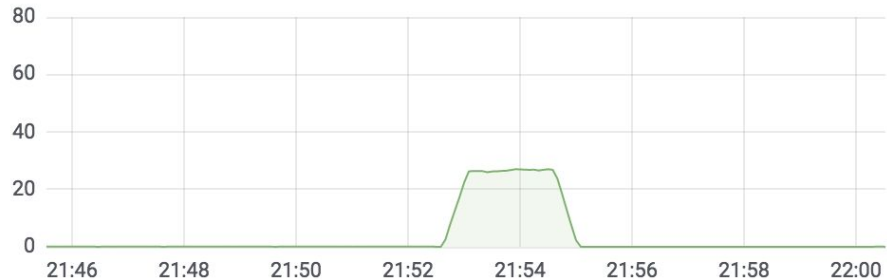
Heap size



CPU usage



GC Rate

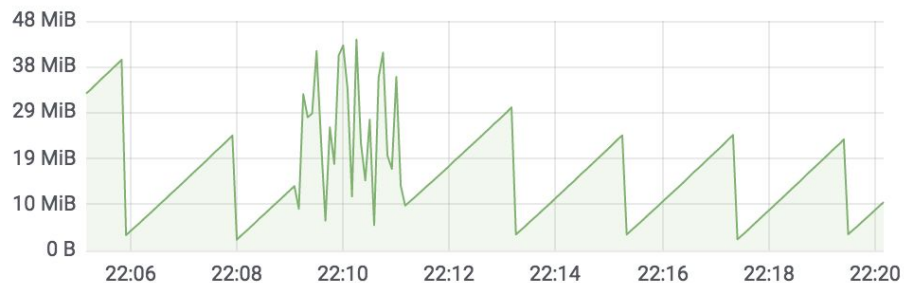


Latency



GOGC=1000

Heap size



CPU usage



GC Rate



Latency



What's happening here? (no load, GOGC=100)

Heap size



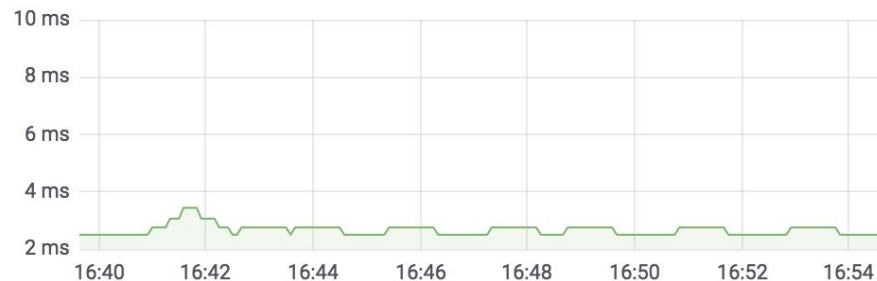
CPU usage



GC Rate



Latency



There *is* another setting!

```
/usr/local/go/src/runtime/proc.go line 4310(!):
```

```
// If we go this long without a garbage collection,  
// one is forced to run.
```

```
var forcegcperiod int64 = 2 * 60 * 1e9
```

What if we turn off GC completely?

GOGC=off

Building podinfo

```
$ go clean -cache
```

```
$ time go build
```

```
real    0m7.331s  
user    0m12.152s  
sys     0m0.660s
```

```
$ go clean -cache
```

```
$ export GOGC=off
```

```
$ time go build
```

```
real    0m5.412s  
user    0m8.460s  
sys     0m0.704s
```



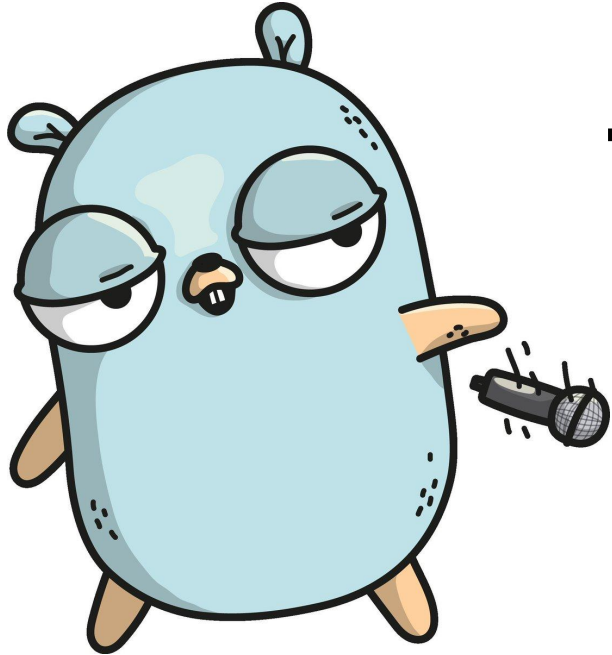
The future?

SetMaxHeap

GOGCMIN



<https://twitter.com/ashleymcnamara>



THANK YOU!

 @bboreham

More information

Good explanation of GOGC and other settings

<https://dave.cheney.net/tag/gogc>

Really detailed history

<https://blog.golang.org/ismmkeynote>