Solving 'Out Of Memory' Errors

Richard Warburton

@richardwarburto www.monotonic.co.uk

```
java.lang.OutOfMemoryError: java heap space
```

Our Problem

Your application has run out of memory

Module Outline

What causes Out Of Memory Errors?

How can I solve them?

Can I proactively prevent out of memory errors?

Running out of Memory

Causes

Memory Leaks

If you leak memory for long enough you'll run out of memory

Memory Overconsumption

Using too much memory to perform a given task

Memory Overconsumption

Memory has a finite limit

Eg: -Xmx

Application needs more is available

Inefficiency or lack of RAM

Growing in relation to current load

Can go down as well as up

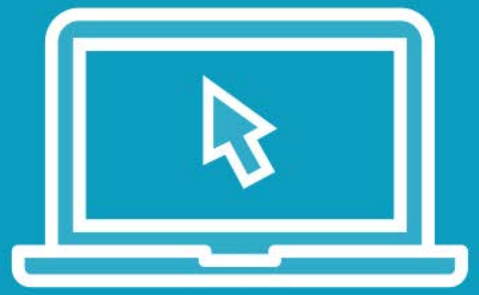
How do you know which is the cause?

Different Causes

Memory Leaks
Grows with activity over time
Don't free what's allocated

Memory Overconsumption
Grows with currently active work
Simply using too much memory

Demo



Simple stateless service that calculates stats from uploaded CSV file.

Example of Memory Overconsumption

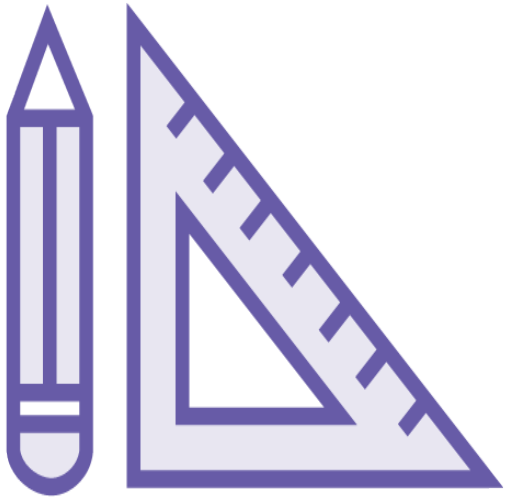
Reducing Memory Usage

Process

Identify what's using your memory

Reduce memory consumption

Allocate less, don't reference as much

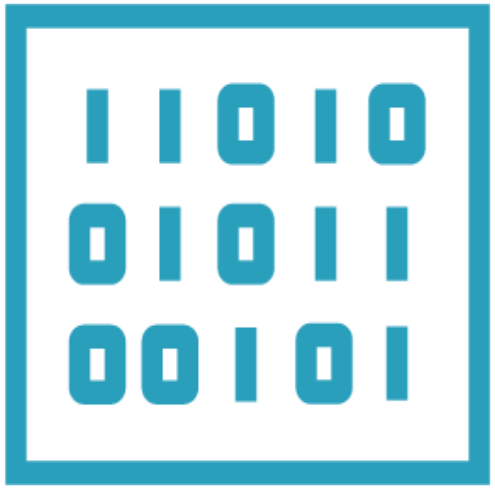


Measure again to validate

Simple stateless service that calculates stats from uploaded CSV file

Measuring the Memory Overconsumption

Tactic: Use Primitive Numeric Types



Primitives
Eg: int
Store a numeric value
4 bytes

Boxed numeric types
Eg: Integer
Object wraps value + pointers
16-24 bytes + 4-8 bytes per reference

Tactic: Recalculate Instead of Storing

Caches take memory

Sometimes caches can be harmful

Think Small

Not just memcache, also using fields, small collections

Tactic: Simplify Domain Model

Abstraction = Overhead
Every layer uses memory

Complexity = Overhead
Do you have overly complex domain model?

Pragmatism
Refactor when it helps your code, not hinders it

Tactic: Increase Available Memory

Configured Max Heap

-Xmx16G

Available Hardware

Have enough RAM for the JVM

Simple stateless service that calculates stats from uploaded CSV file

Fixing the Memory Overconsumption

Proactive Measures

"We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil."

Donald Knuth

Monitoring

Memory Consumption

GC Logs/JVisualVM

Application Performance Monitoring

Performance Testing

Drive system with stress test

Find and fix problems

20% Time

Often used for research or pet projects

Periodically use it for performance analysis/improvements

Speculatively pick at low hanging fruit

Only things that are likely to cause a production problem soon

Conclusions

Summary

Two main causes

- Leaks
- Overconsumption

Diagnose and measure before optimizing

Tactics for reducing memory use

Prevention better than cure, taking care to avoid premature optimization