

Fluid infrastructure

Economies of “scale” using containers

Tom Wells
Chief Disruption Officer
Synthesis Labs



scale

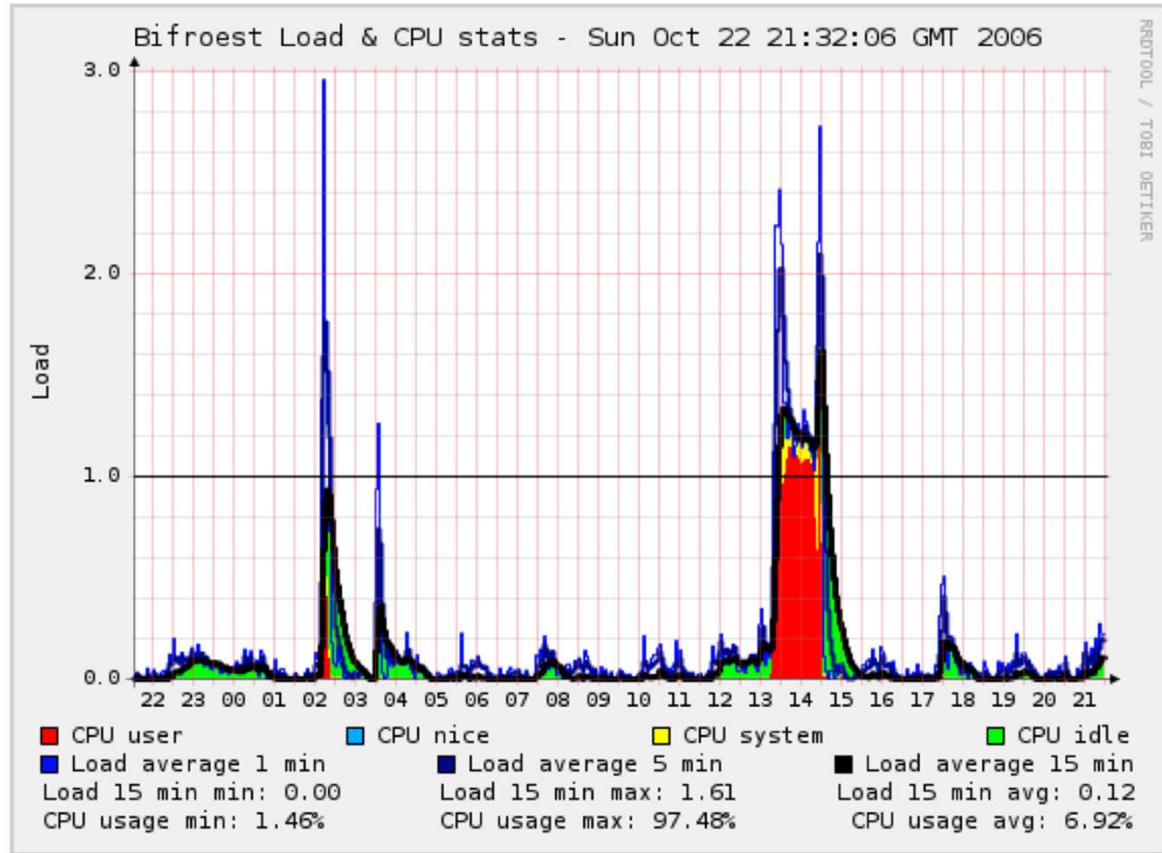
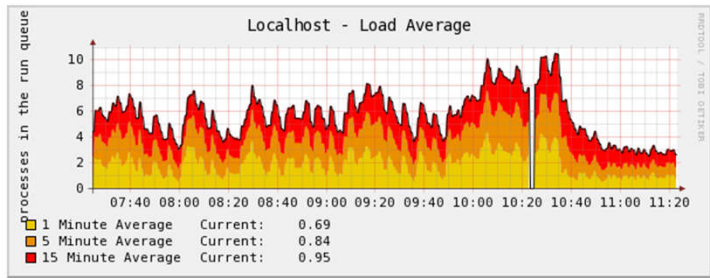
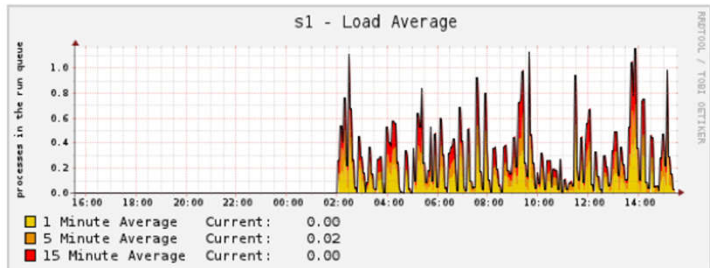
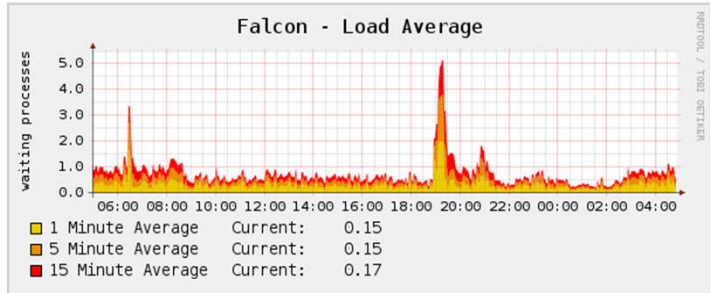
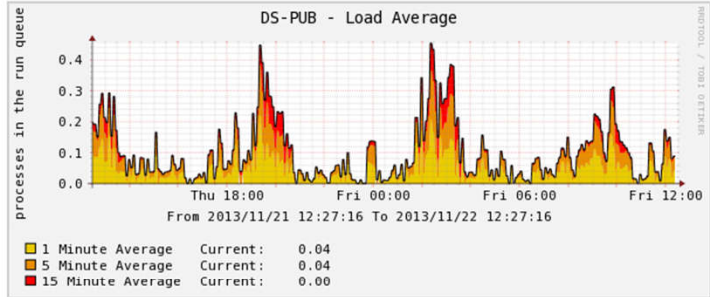


scale

aka.

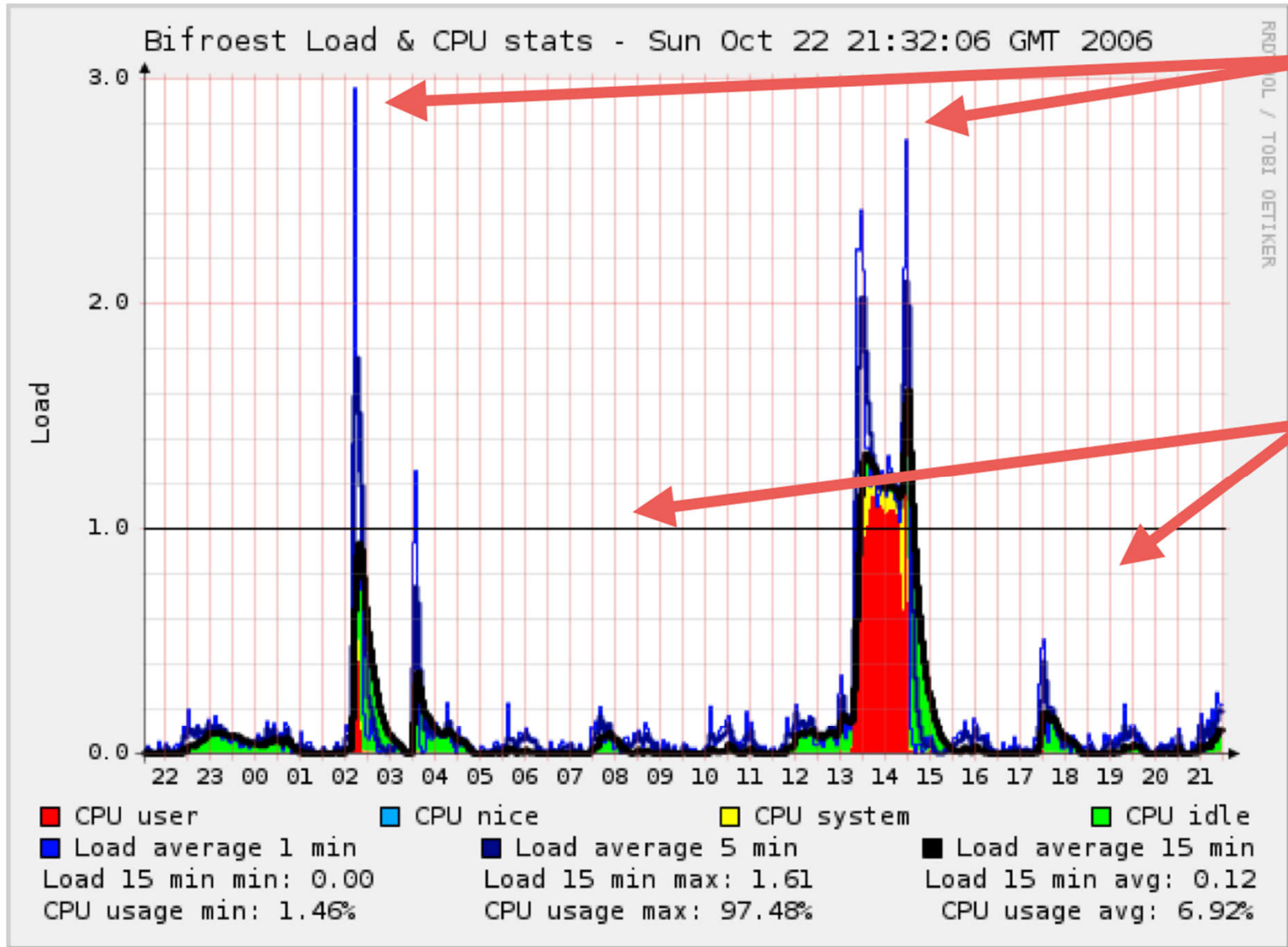
supply \sim demand





typical load average
graphs



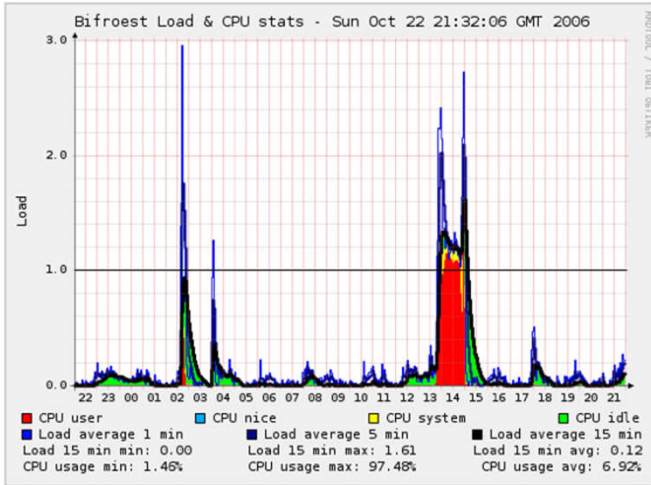


sizing for peaks

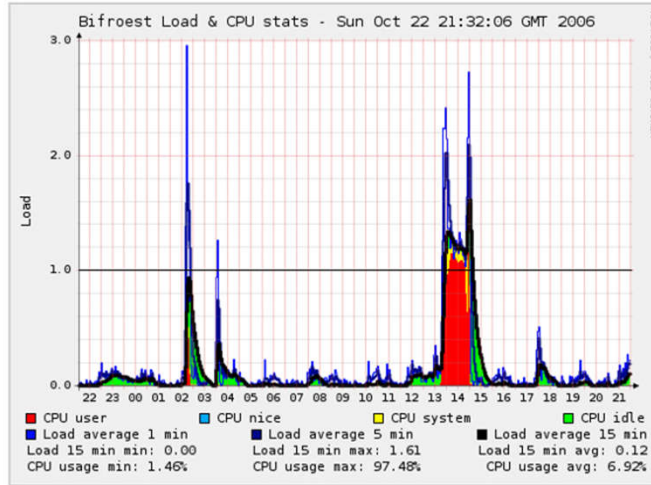
vs.

under-utilisation
(waste)

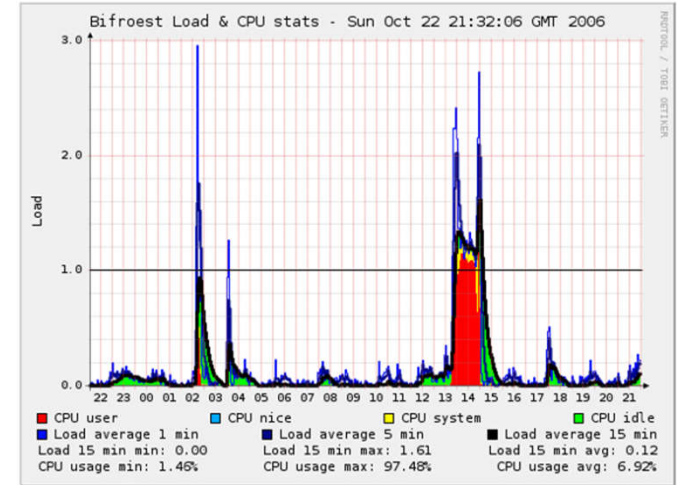




Production 1



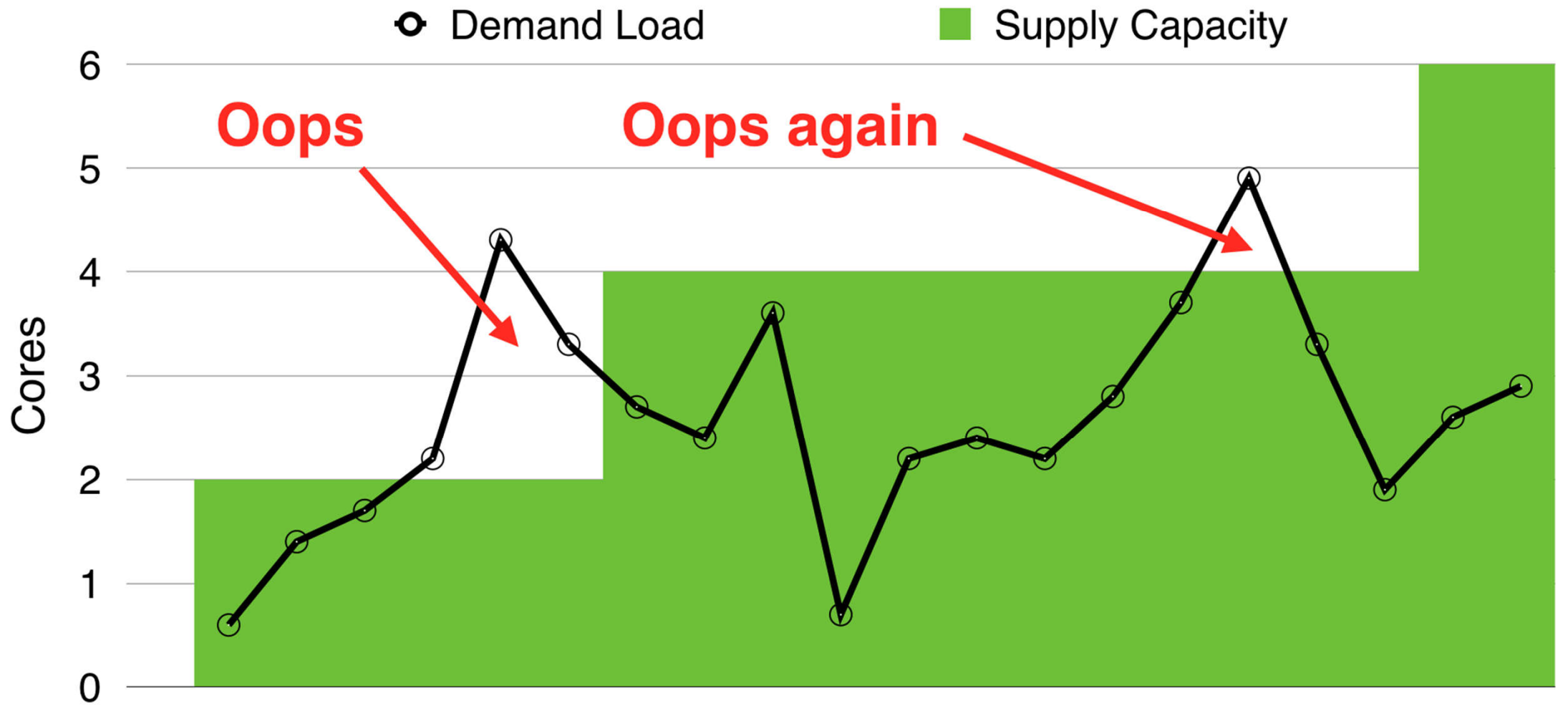
Production 2



Production 3

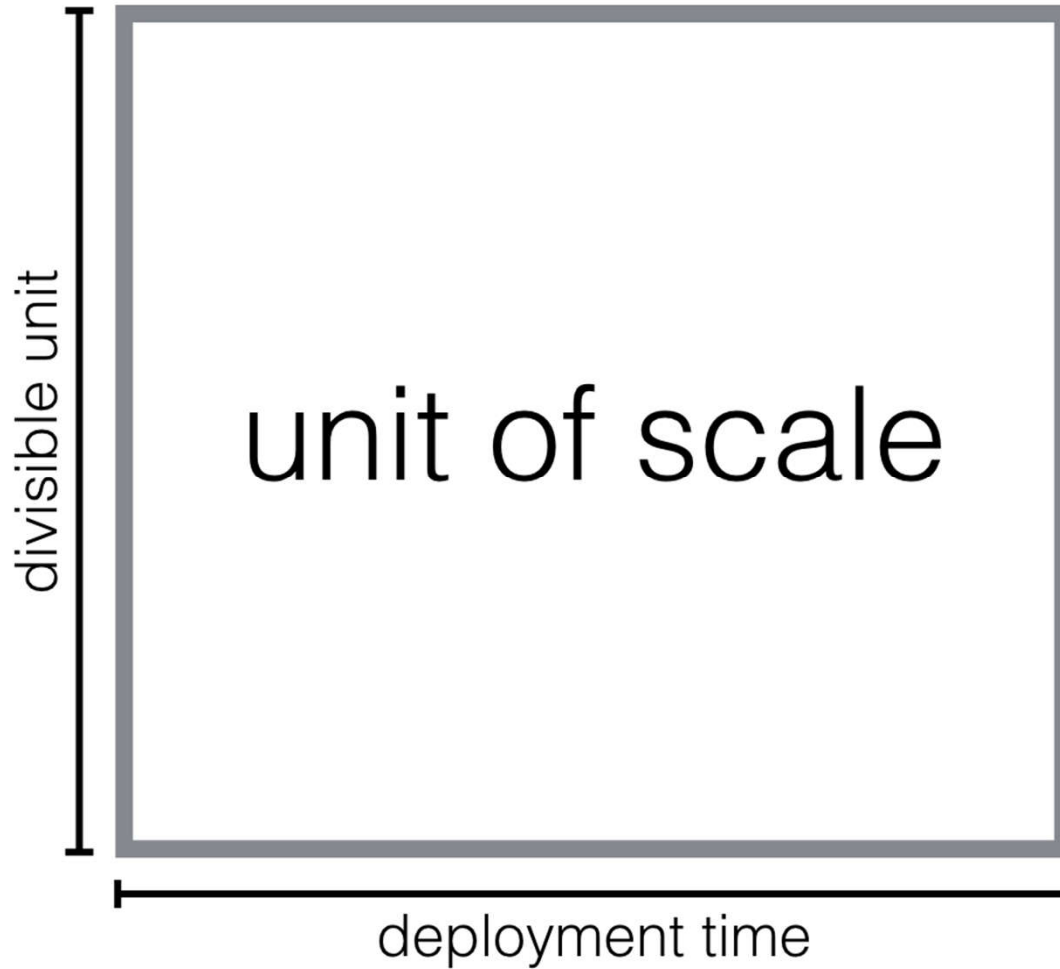
redundancy amplifies waste





typical scaling up





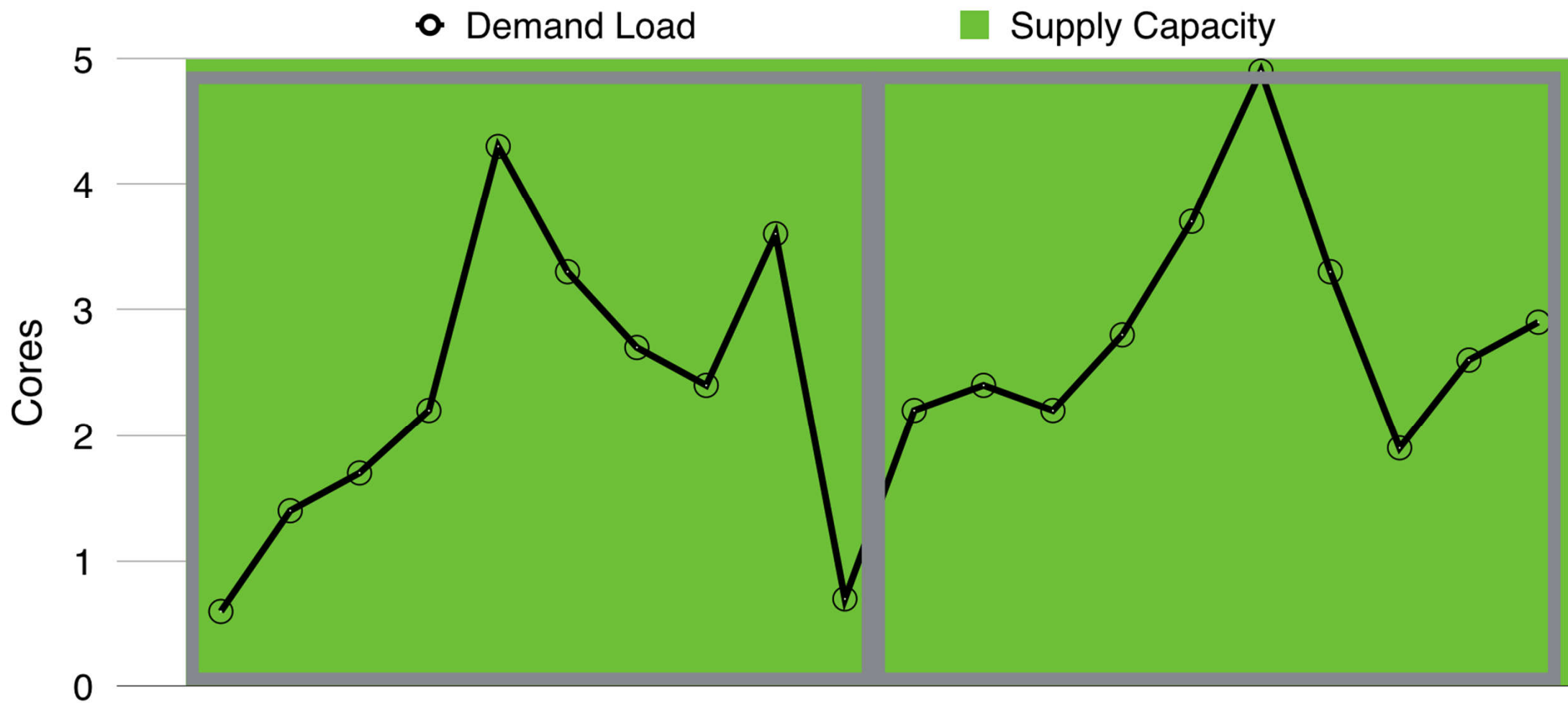
Objective #1

Reduce deployment time
(*weeks, days, hours, minutes, seconds*)

Objective #2

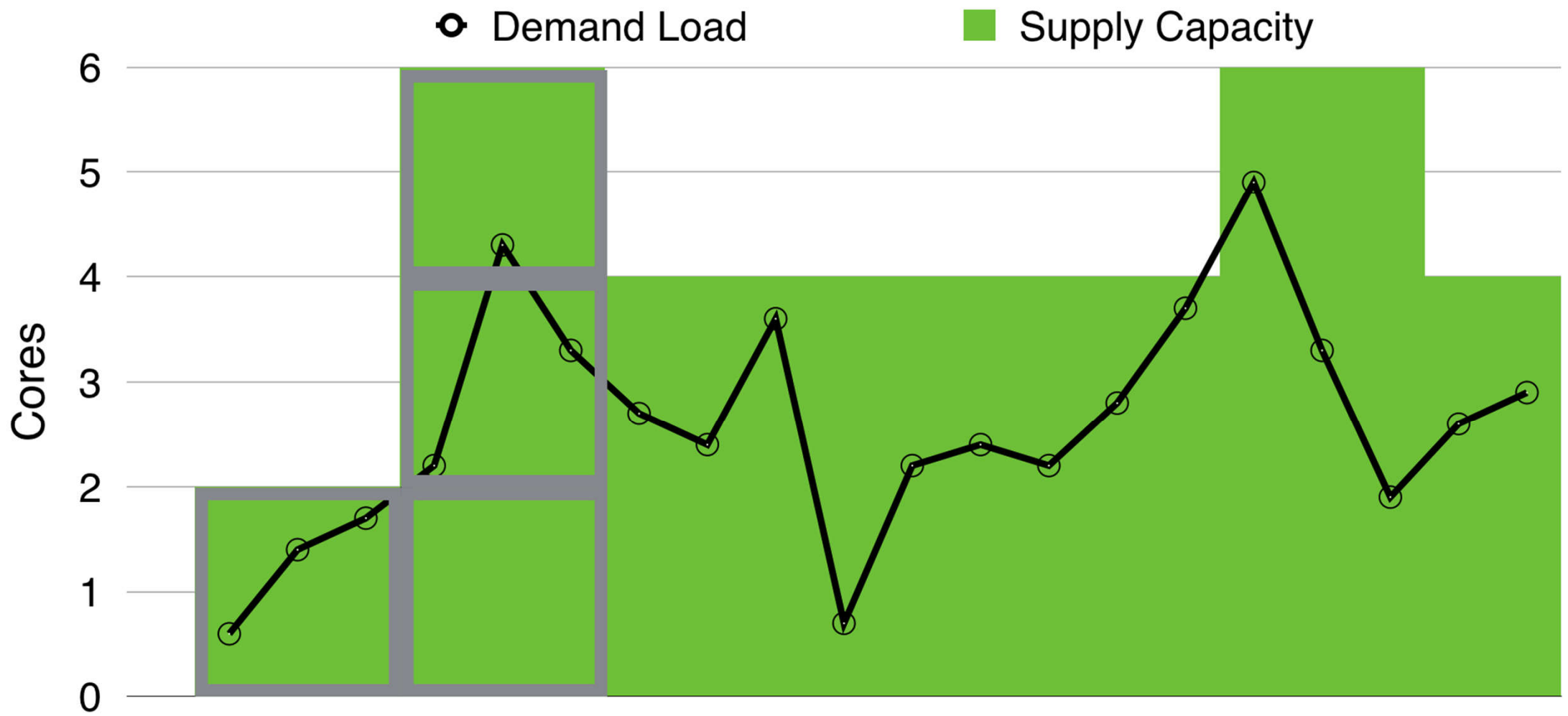
Reduce size of divisible unit
(*large, medium, micro hosts, containers*)





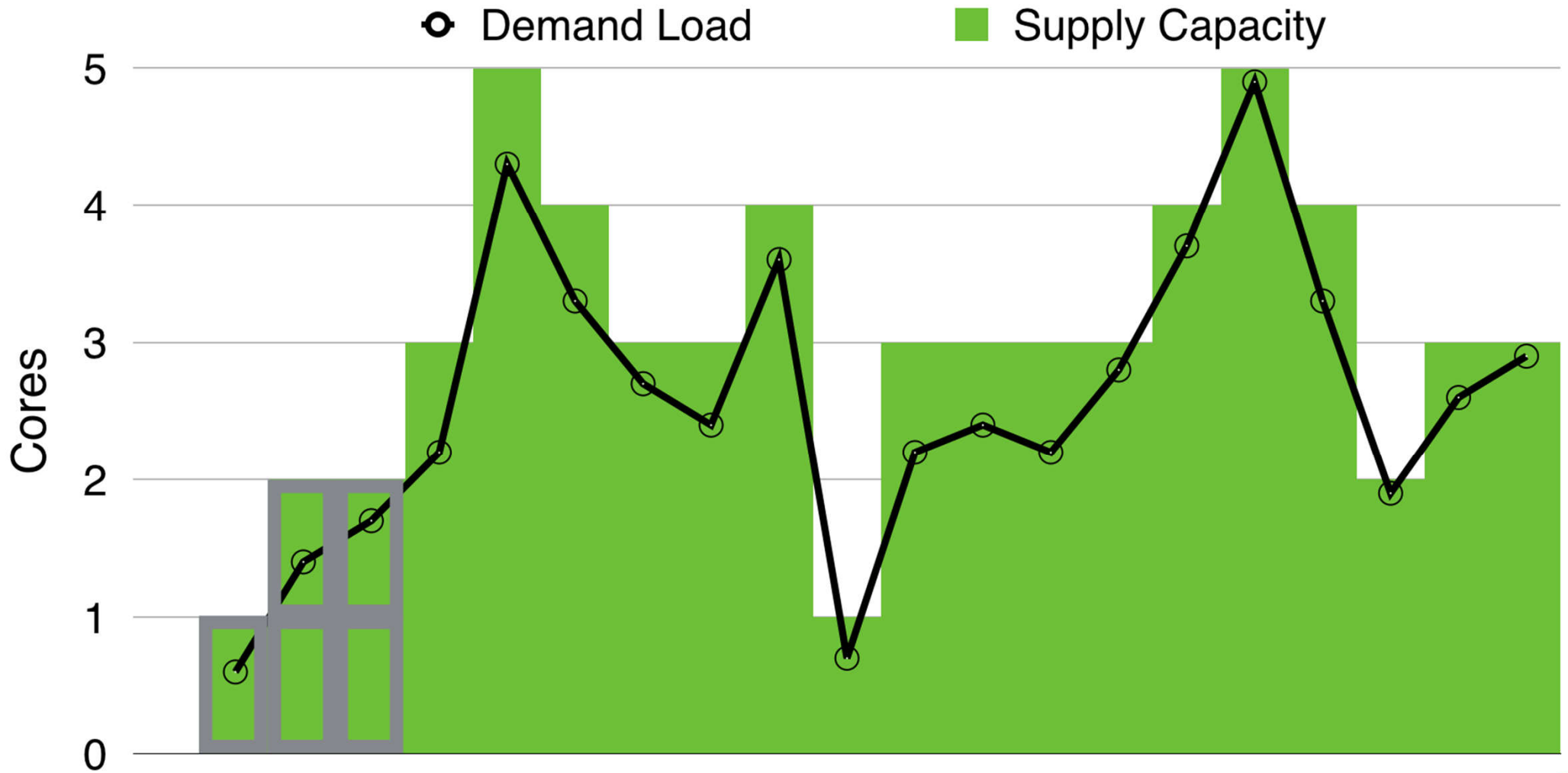
lets break it down





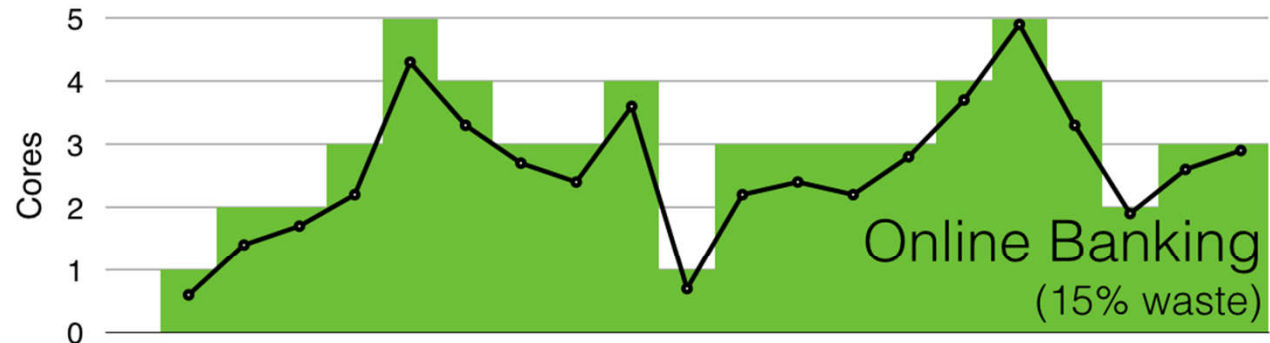
lets break it down



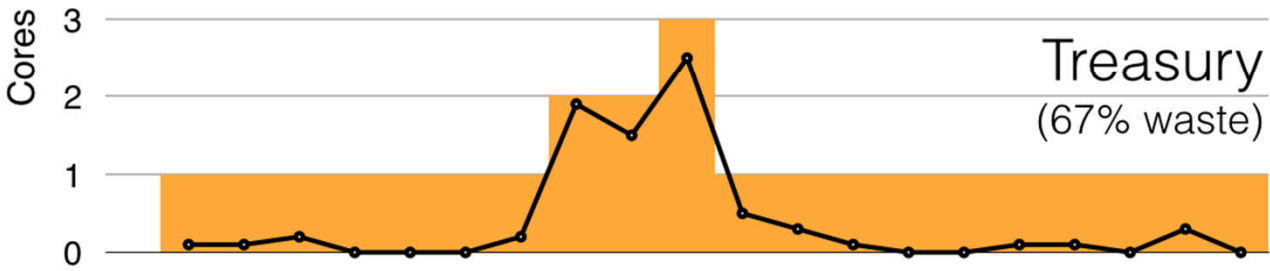


lets break it down

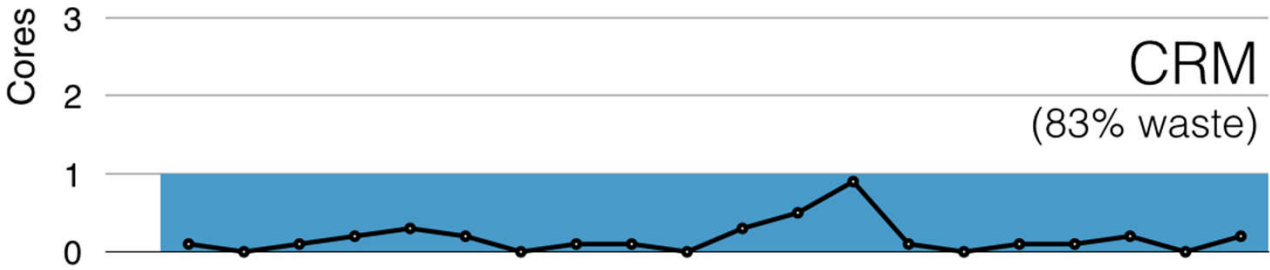




multiple apps
scaling within
a silo

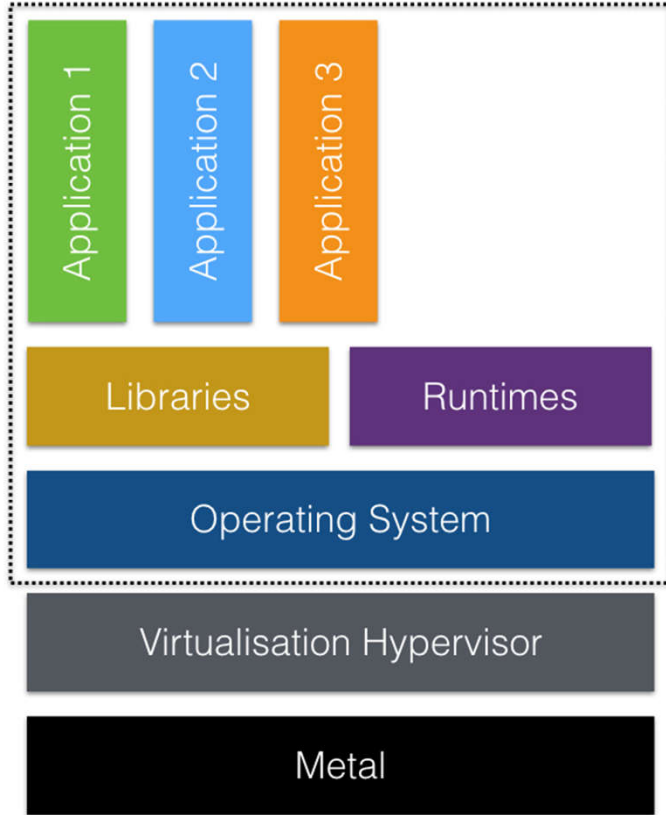


Objective #3
Encapsulate application
dependencies

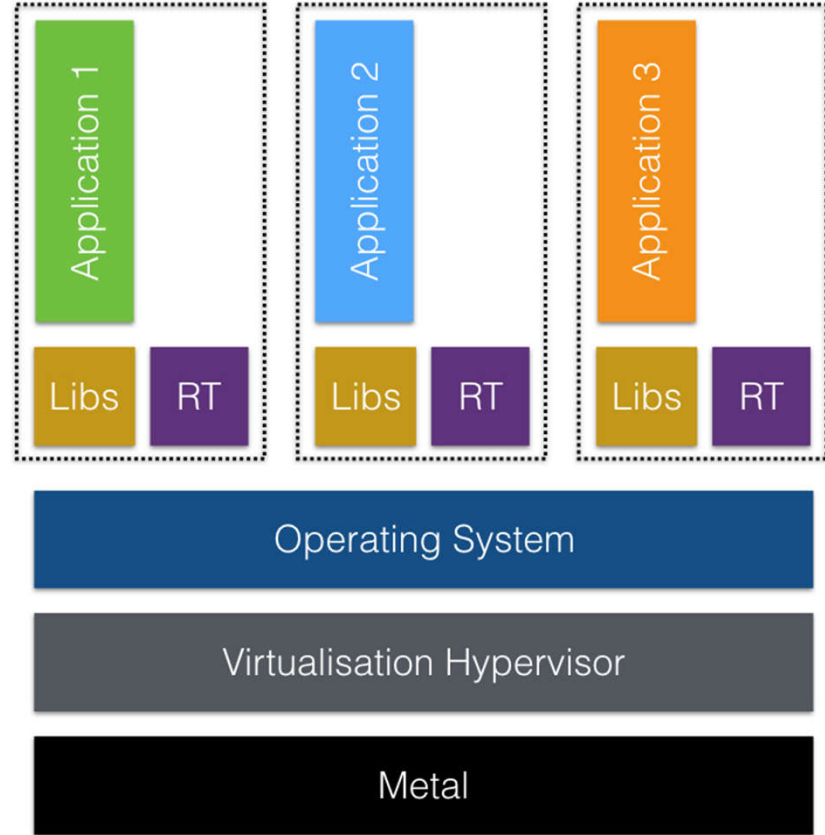


Objective #4
Break down application
infrastructure silos



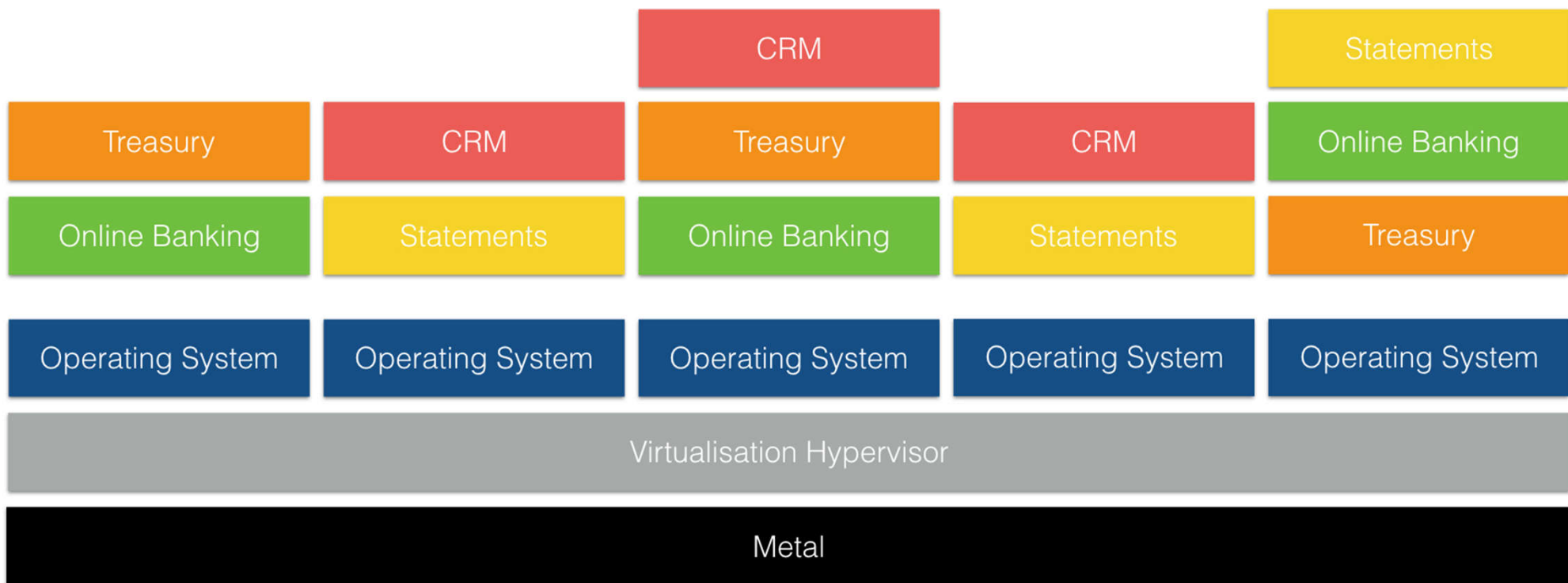


VS.

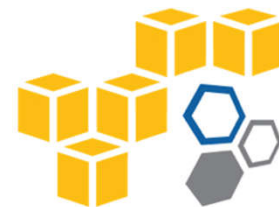


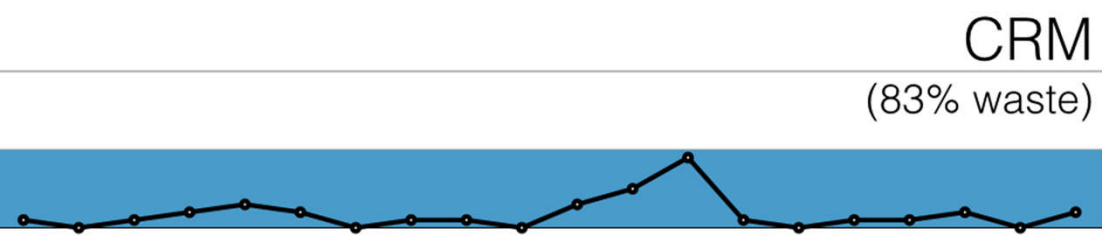
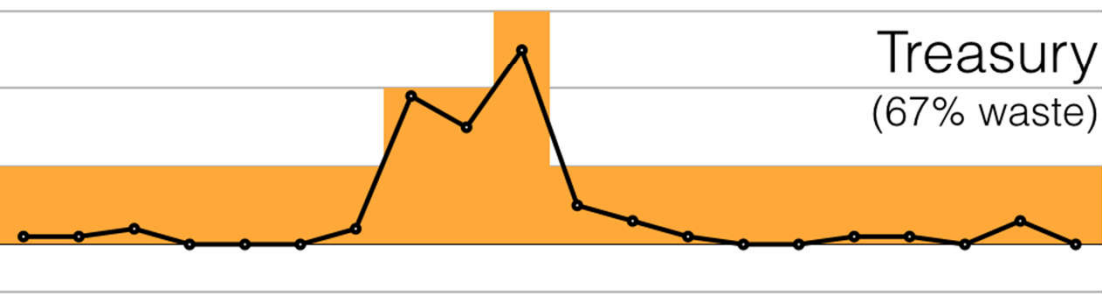
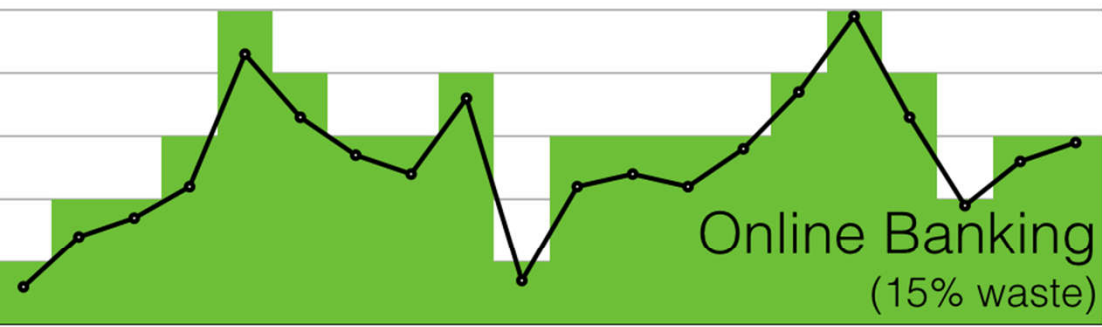
enter containers



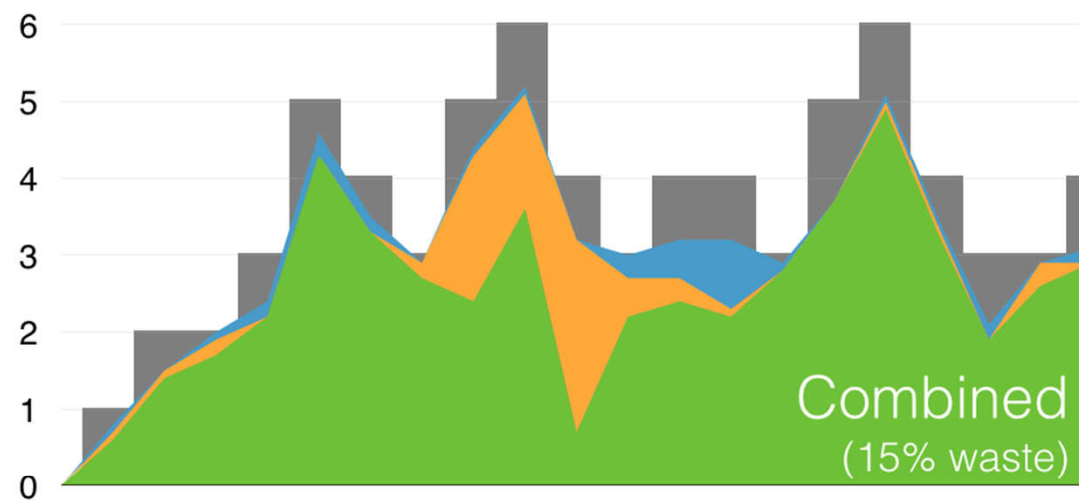


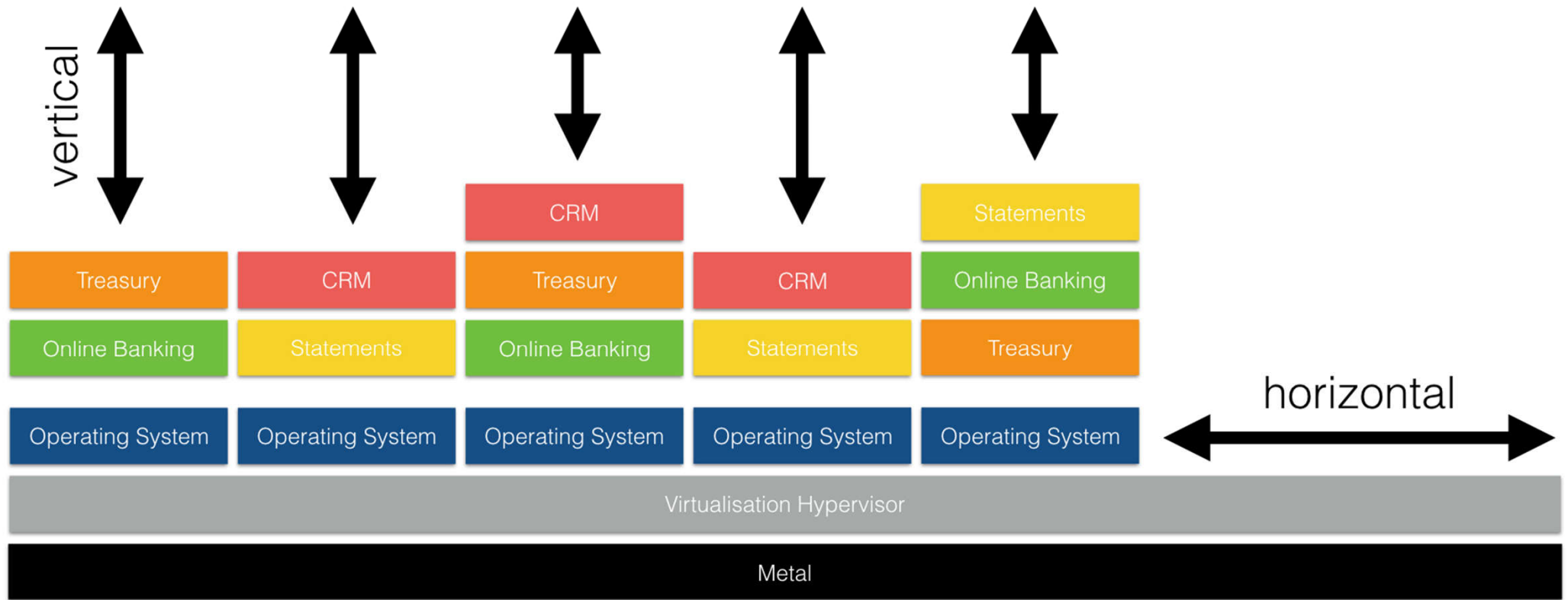
container architecture





combined usage





fluid infrastructure



Objective #1

Reduce deployment time - *automate and containerise*

Objective #2

Reduce size of divisible unit - *small containers on small hosts*

Objective #3

Encapsulate application dependencies - *portable containers*

Objective #4

Break down application silos - *fluid container infrastructure*

Thank you :)

