Principles of High Performance Map Rendering

No single answer to fast serverside maps!

Rather a balance of considerations...

Latency - Scalability - Stability

Latency - Scalability - Stability

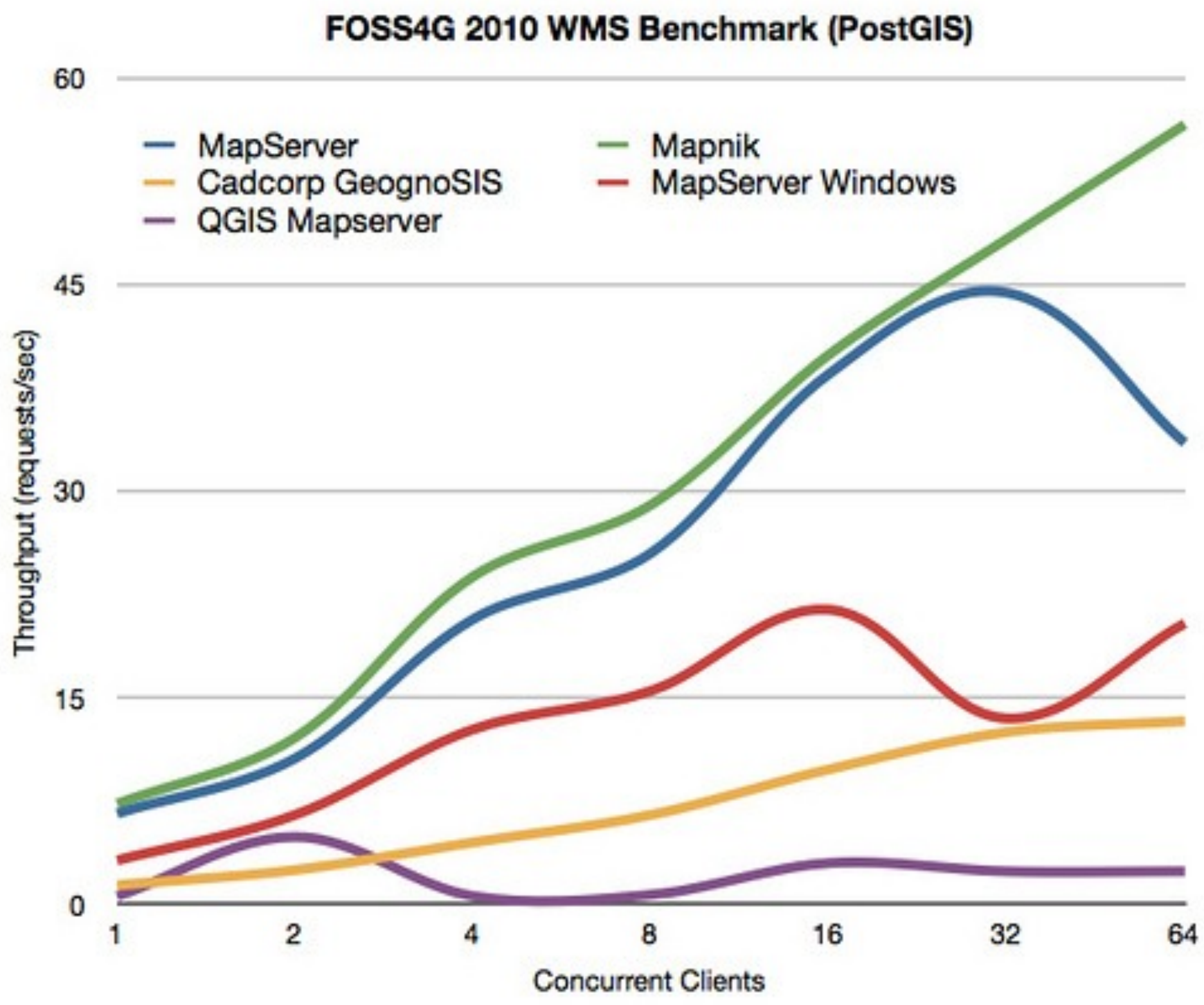
FIFO priority	distribute priority	no queue, just limit
No pooling/throttling	pooling	throttling
High CPU/Mem	High CPU/Mem	High CPU/Mem
SSD's	Good Disks	io wait vs swapping
Lots in-memory	careful with memory	really careful
Optimize low zooms	Optimize high zooms	Optimize everything
Chip speed GHz	Number of Cores	Number of Machines

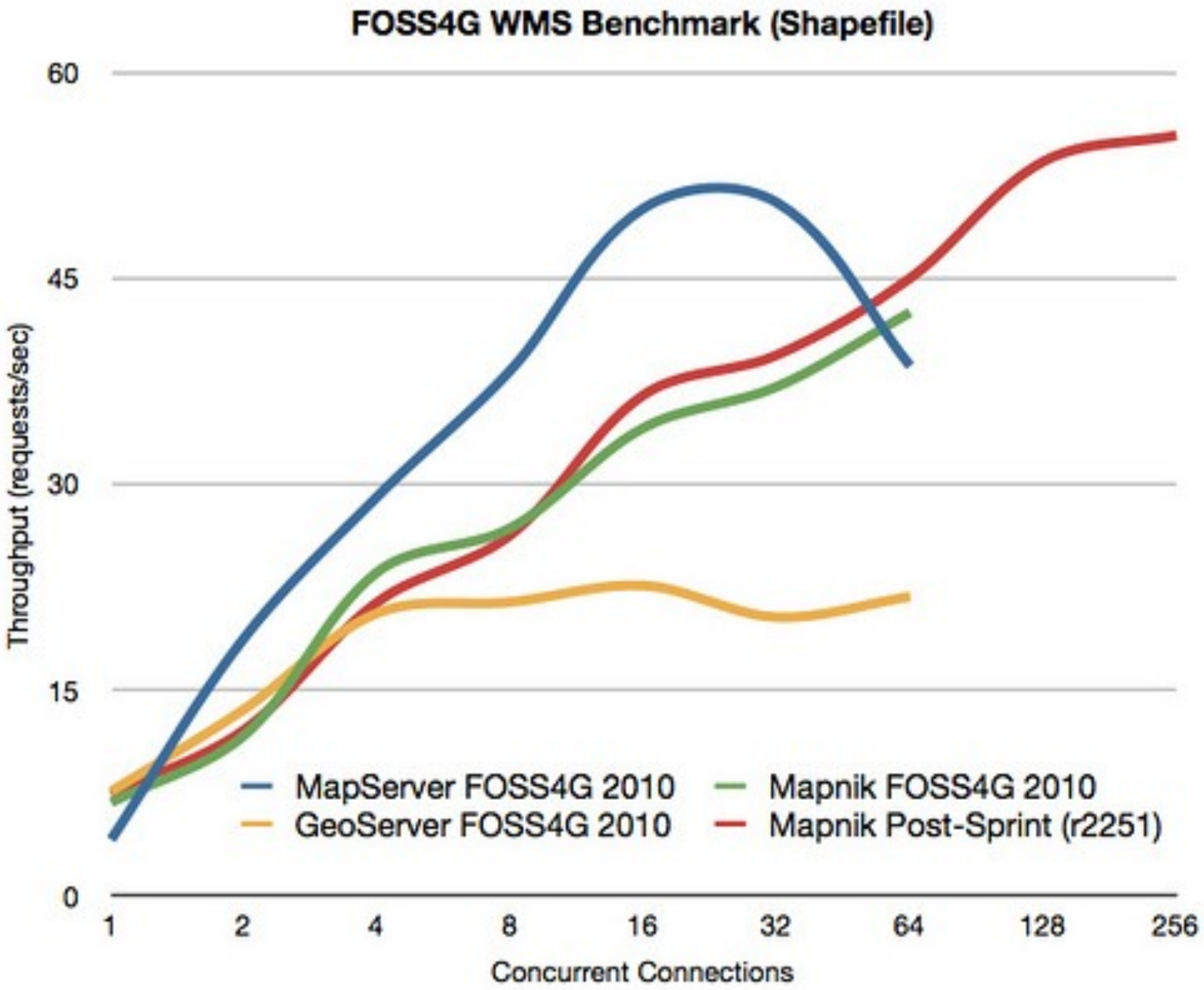
Latency - Scalability - Stability

FIFO priority	distribute priority	no queue, just limit
No pooling/throttling	pooling	throttling
High CPU/Mem	High CPU/Mem	High CPU/Mem
SSD's	Good Disks	io wait vs swapping
Lots in-memory	careful with memory	really careful
Optimize low zooms	Optimize high zooms	Optimize everything
Chip speed GHz	Number of Cores	Number of Machines

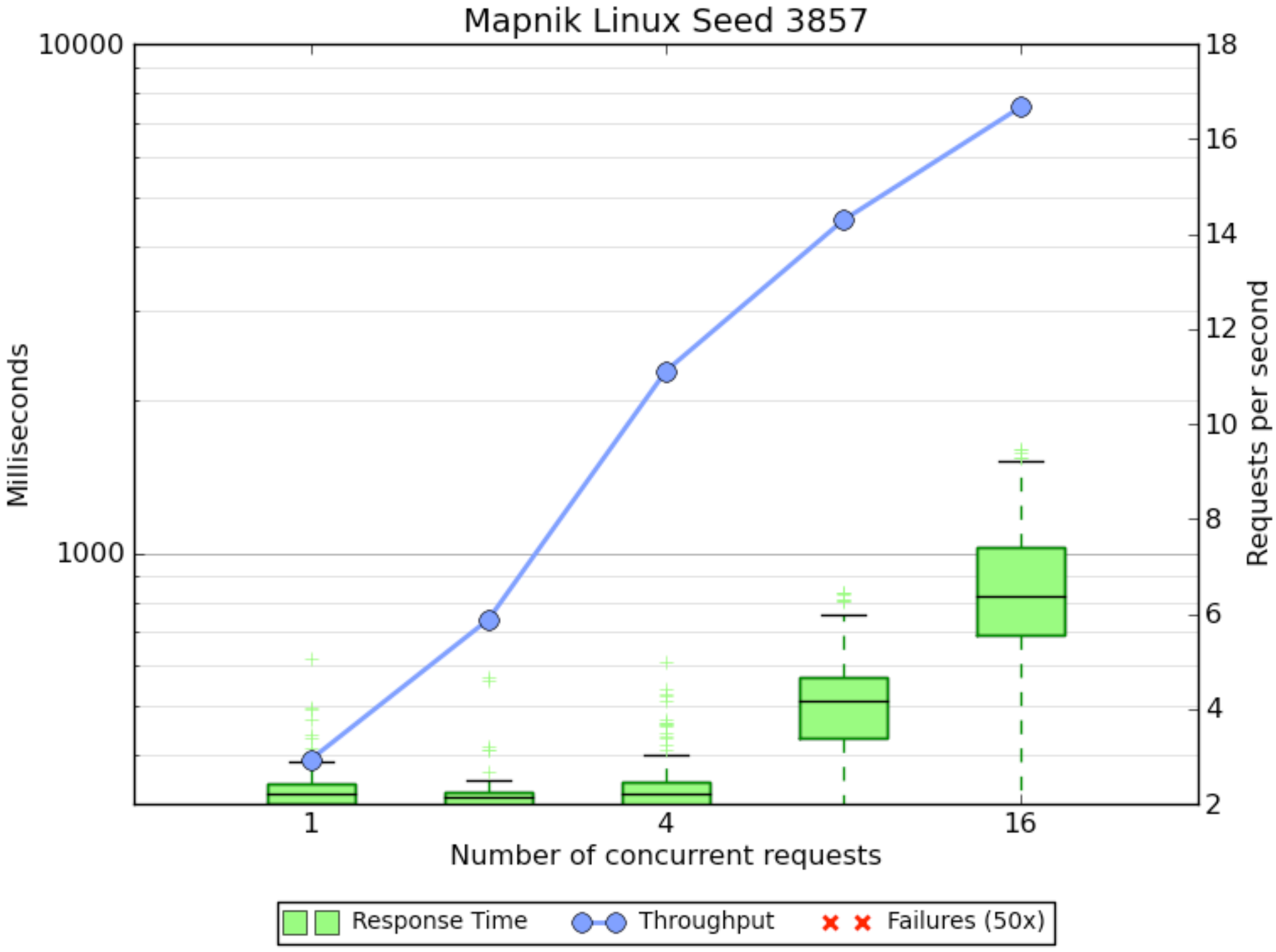
Latency - Scalability - Stability

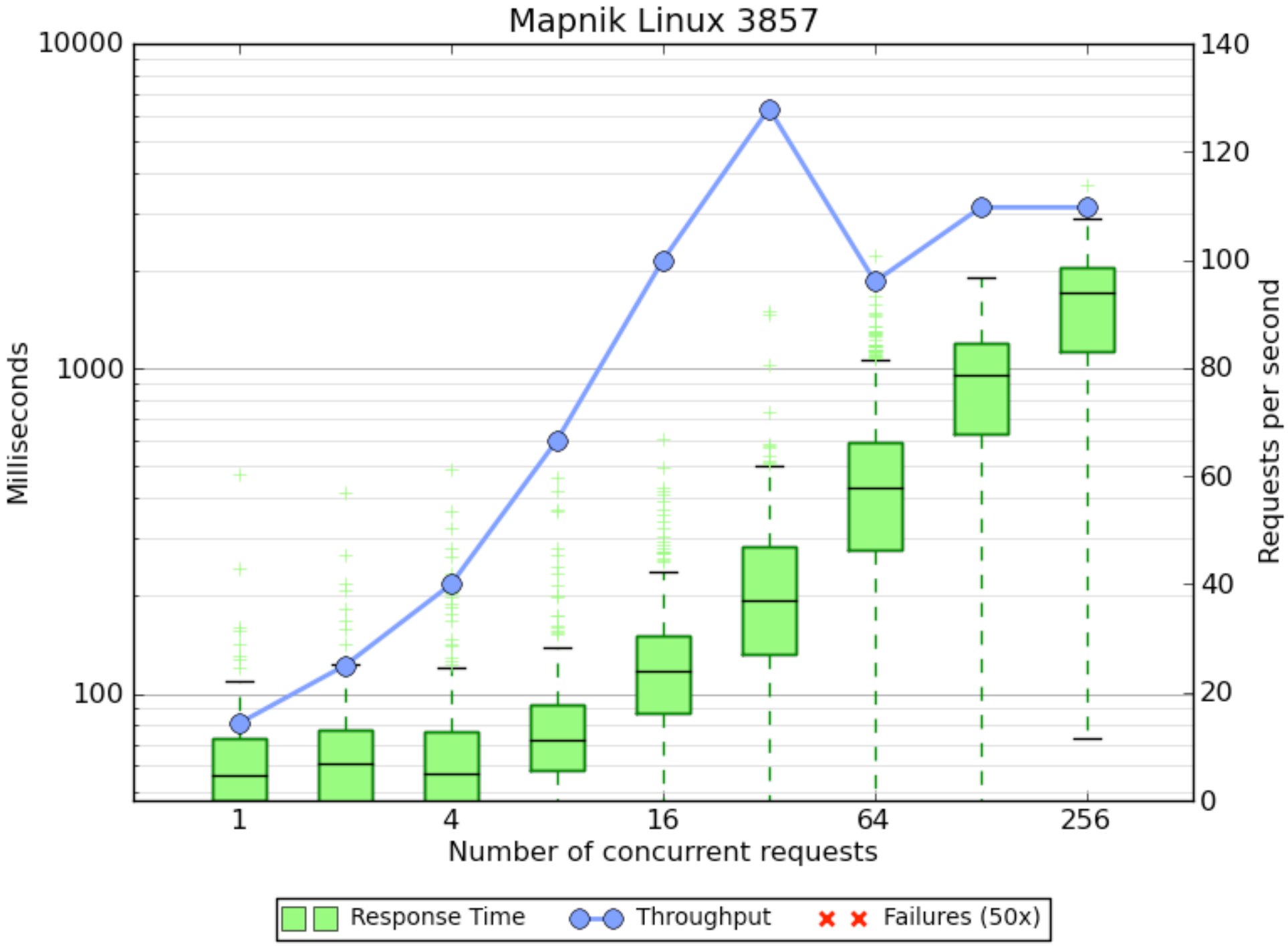
FIFO priority	distribute priority	no queue, just limit
No pooling/throttling	pooling	throttling
High CPU/Mem	High CPU/Mem	High CPU/Mem
SSD's	Good Disks	io wait vs swapping
Lots in-memory	careful with memory	really careful
Optimize low zooms	Optimize high zooms	Optimize everything
Chip speed GHz	Number of Cores	Number of Machines


FOSS4G 2010 WMS Benchmark (PostGIS)
MapServer
Cadcorp GeognoSIS
QGIS Mapserver
Mapnik
MapServer Windows
Throughput (requests/sec)
60
45
30
15
0
1
2
4
8
16
32
64
Concurrent Clients



http://developmentseed.org/blog/2010/oct/19/qa-mapnik-performance-just-important-its-beauty/


Mapnik Linux Seed 3857
10000
1000
Milliseconds
18
16
14
12
10
8
6
4
2
Requests per second
1
4
16
Number of concurrent requests
Response Time
Throughput
Failures (50x)


Mapnik Linux 3857
10000
1000
100
140
120
100
80
60
40
20
0
Milliseconds
Requests per second
1
4
16
64
256
Number of concurrent requests
Response Time
Throughput
Failures (50x)