Performance and Scalability on Itanium

Supported by UNSW, NICTA, HP and Google through the Gelato Federation
RAID 0, sequential asynchronous

![Graph showing bandwidth (KiB/s) vs processes for different scheduling policies: antici, cfq, cfq-no, deadline, fifo, noop.](image)

- antici
- cfq
- cfq-no
- deadline
- fifo
- noop
Single disk, random

![Graph showing aggregate bandwidth (KiB/s) vs. processes for various disk scheduling algorithms: AS, CFQ, DL, FIFO, NOOP. The graph illustrates the performance under random access patterns.]
Single disk, 1MiB sequential

Aggregate Bandwidth (KiB/s)

Processes

- antici
- cfq
- cfq-no
- deadline
- fifo
- noop
10 disk RAID 0, random

![Graph showing aggregate bandwidth (KiB/s) vs processes for different disk schedules: antici, cfq, deadli, fifo, noop, and cfq-noidle.](image)
10 disk RAID 0, 1MiB sequential

Aggregate Bandwidth (KiB/s)
What Parameters are Useful?

- Queue depth
- Underlying storage device
- RAID topology
- ...?
HOW TO DISCOVER THE PARAMETERS?

➔ User input
➔ Measurements
➔ Ask other layers
Queue Depth

- Measurements or API
- Measurement a bit suspicious
- ... but why is queue depth useful?
- For anticipation, it’s not
- What we really want is device type
Single disk, 1MiB sequential, TCQ depth 64

Aggregate Bandwidth (KIB/s)

Processes

Queue Depth
Single disk, 1MiB sequential, TCQ depth 1

![Diagram showing aggregate bandwidth (KiB/s) vs. processes for different queue depths and I/O schedulers. The X-axis represents the number of processes, ranging from 0 to 600, and the Y-axis represents the aggregate bandwidth in KiB/s, ranging from 0 to 50000. The graph includes lines for antici, cfq, cfq-no, deadlock, fifo, and noop, each with a different color. The line for antici shows a sharp decline in bandwidth as the number of processes increases, while the other lines show more gradual declines or remain relatively stable.]
**DEVICE TYPE**

➤ Ask the driver

➤ What is a suitable level of abstraction?
   ① Random access
   ② True parallelism
   ③ Just device type (H/W RAID, SSD, etc)
RAID TOPOLOGY

- Stripe boundaries and width
- Request → disk mapping
- Take measurements
- Per-spindle scheduling
- Better resource accounting
AUTOMATIC TUNING

→ Seek profile → anticipation expiry
→ Read/write ratio + deadlines