Performance and Scalability on Itanium

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

![Graph showing aggregate bandwidth (KiB/s) for different disk algorithms (AS, CFQ, DL, FIFO, NOOP) as a function of the number of processes. The graph indicates that AS generally provides the highest bandwidth, followed by CFQ, DL, FIFO, and NOOP.](image-url)
Single disk, 1MiB sequential

![Graph showing aggregate bandwidth (KiB/s) with processes on the x-axis and aggregate bandwidth on the y-axis. Different lines represent different disk scheduling algorithms: antici, cfq, cfq-no, deadline, fifo, and noop. The graph illustrates the performance of these algorithms as the number of processes increases.](image-url)
10 disk RAID 0, random
10 disk RAID 0, 1MiB sequential
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
Single disk, 1MiB sequential

![Graph showing aggregate bandwidth (KiB/s) for different queue depths.](image-url)
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