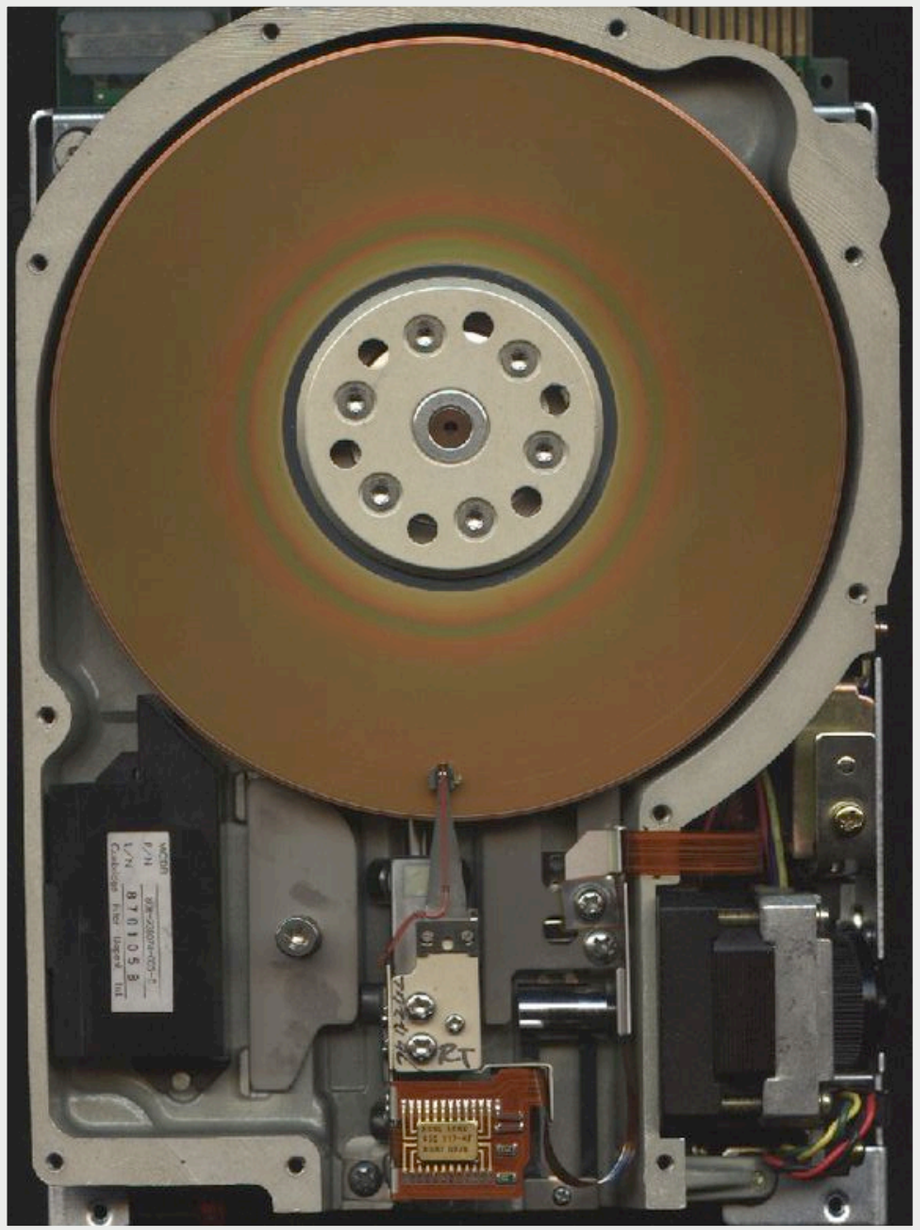
BOTTLENECKS

Past, Present, Future

IN THE "BEGINNING"

- Circa 1980:
  - 10Mb/s local network was a fair alternative to a cheap local disk with 5Mb/s transfer and 50-100ms average seek.
  - Multidrop network installations often had problems.



INEFFICIENCIES

- File fragmentation and placement limit operational throughput to around 5% of the devices' maximum 2MB/s.
- Affordable disk capacities O(1GB) outstripped affordable tape backup (6250 bpi 9-track tape).
- Poorly performing Ethernet cards were common.
- Routers could buffer tens of packets.

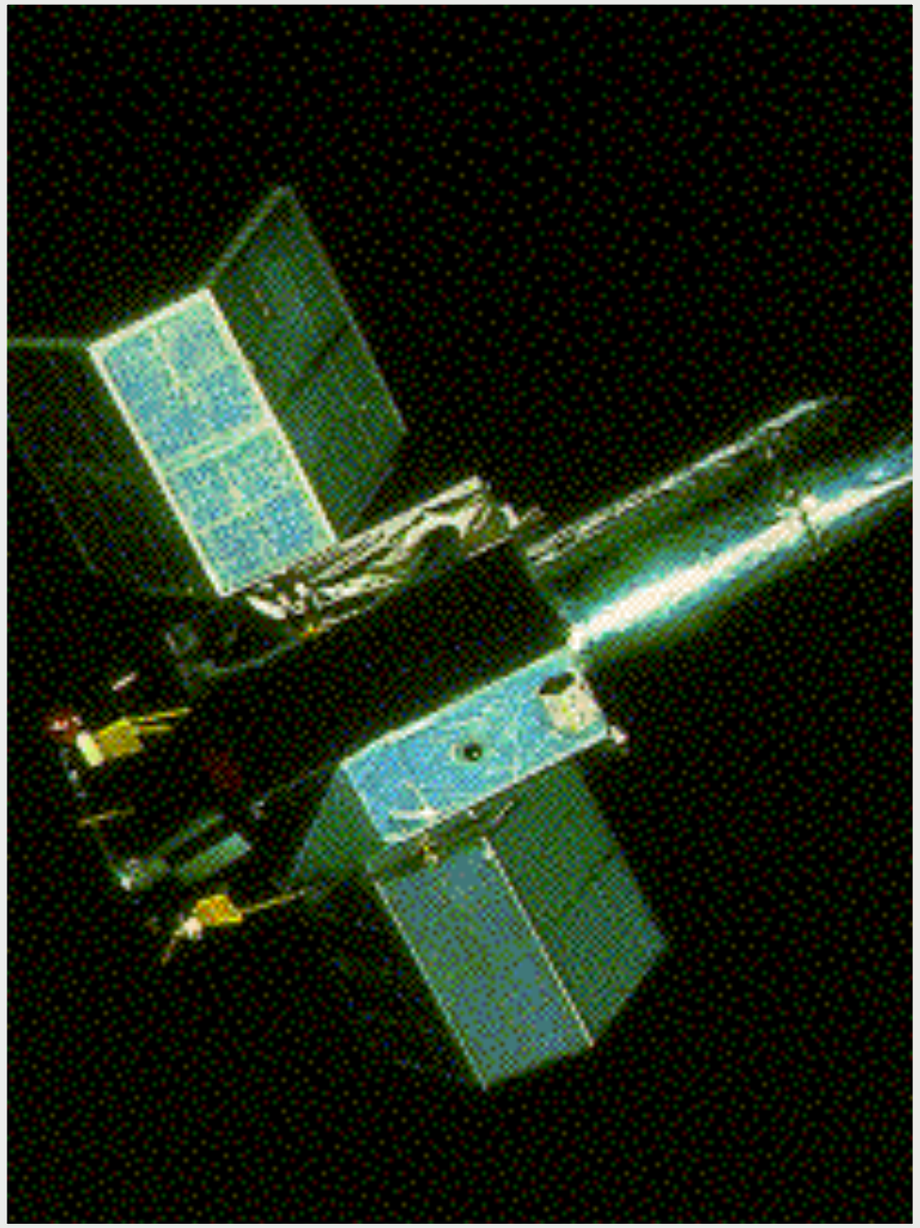
⭐System memory: 20sec. File system: 10min. In theory.

UNIX AND NSFNET

- BSD Unix brought the "Fast File System" with fewer seeks per file (or directory).
- TCP over NSFnet WAN (56kb – 45Mb/s) was so fragile, files were generally broken into 50–100kB chunks for transfer.

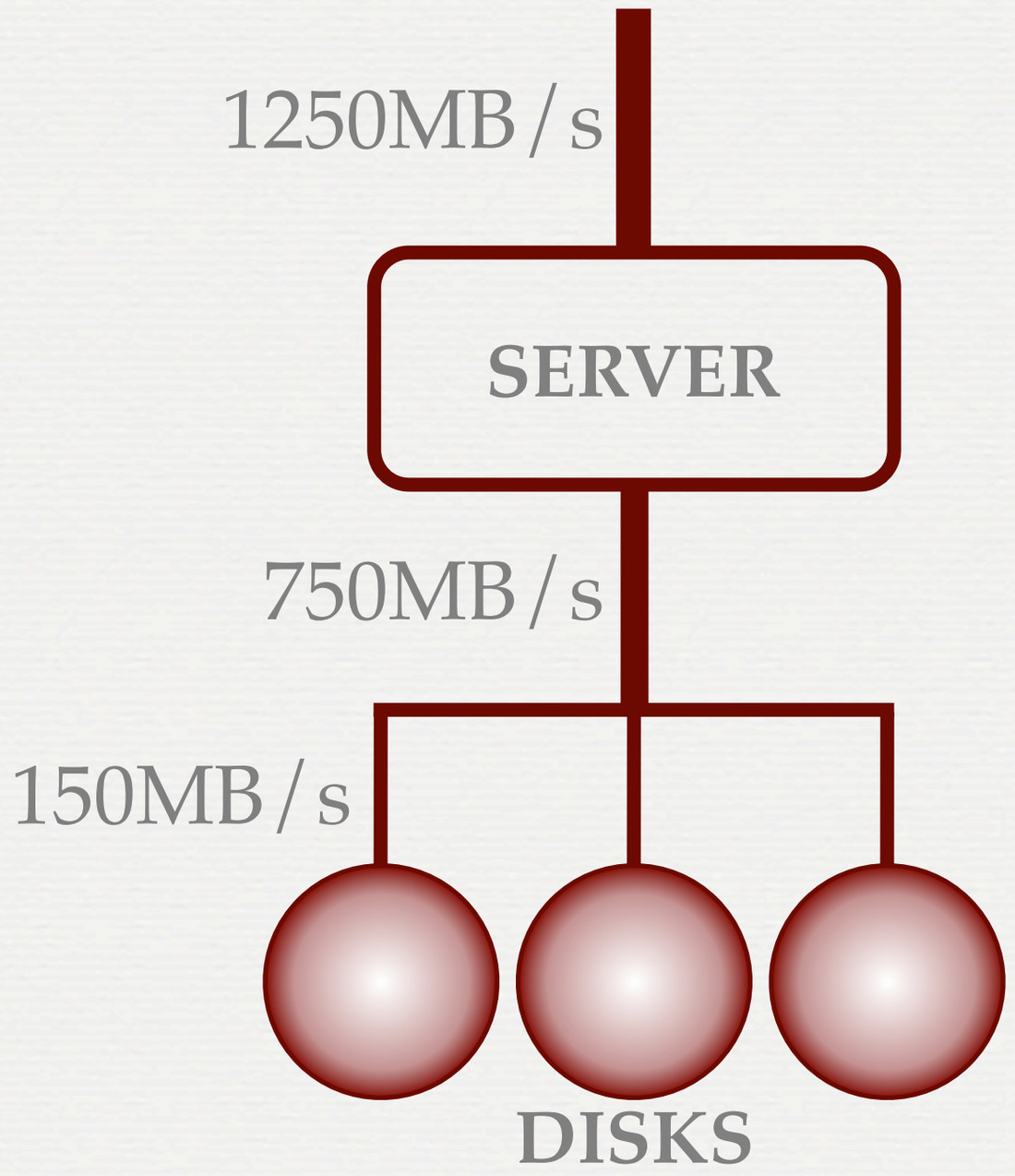
MID-80'S

- Distributed processing becomes feasible on the LAN: uux, Berknet, rsh, SUNRPC, Condor.
  - Small datasets move over the wide area in real time...on dedicated links.
  - Large data sets move over lunch, or overnight.

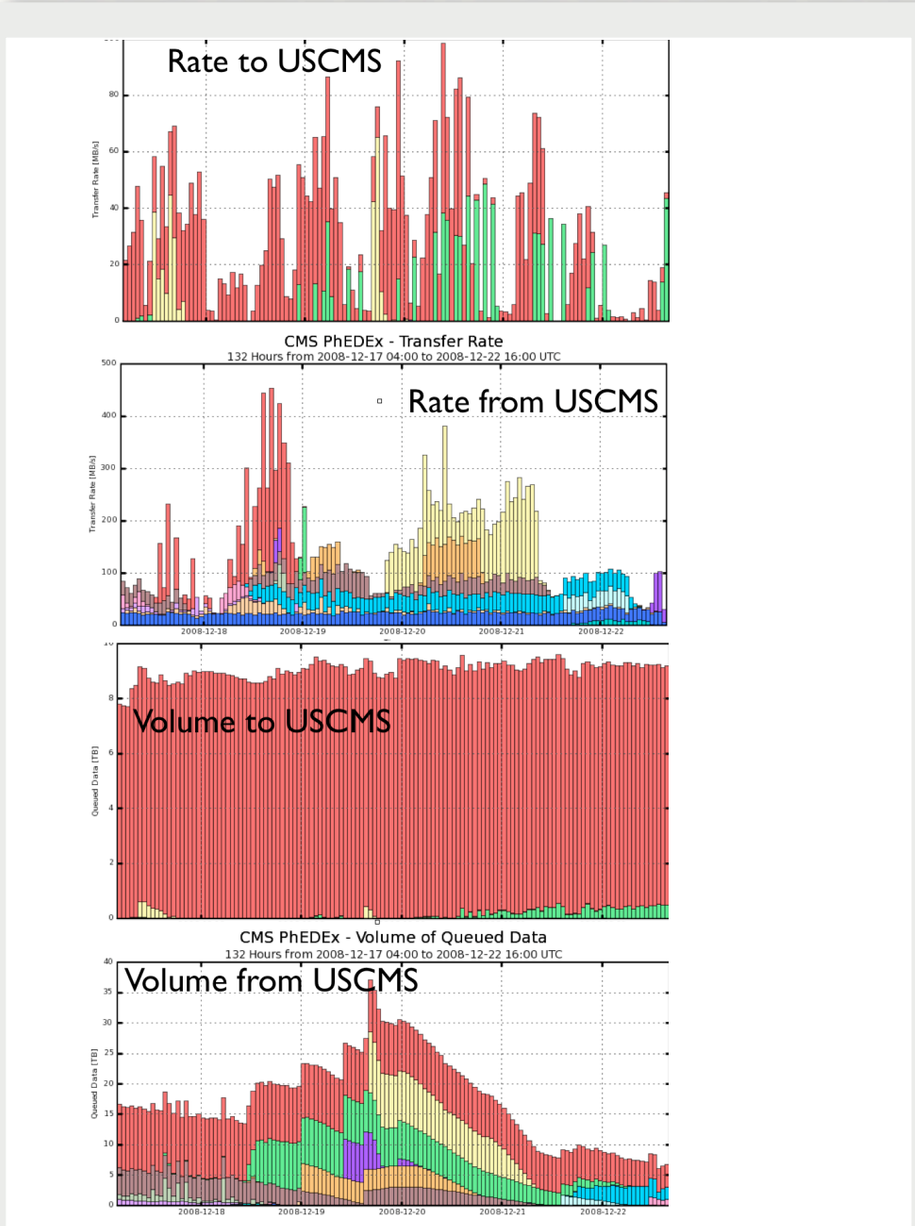


CURRENT PLATFORMS

- Servers' 1–10Gb/s network speeds are well matched to disk or disk array speeds.
- High CPU power → stealing memory for compute jobs.
- ⭐System memory: 1min. File system: 90min. In theory.



THE LHC ERA



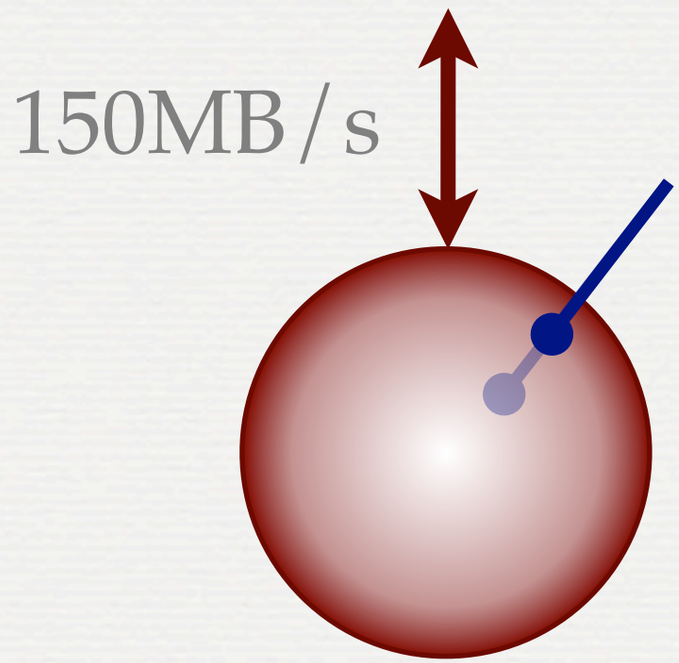
- HEP groups brought file catalogs, subscriptions, transfer managers: fewer user commands per file.
- The entire production and analysis chain is so heterogenous (and in some places, fragile), data sets are broken up into 1–5GB files for transfer and processing.

CURRENT BOTTLENECKS

File servers:

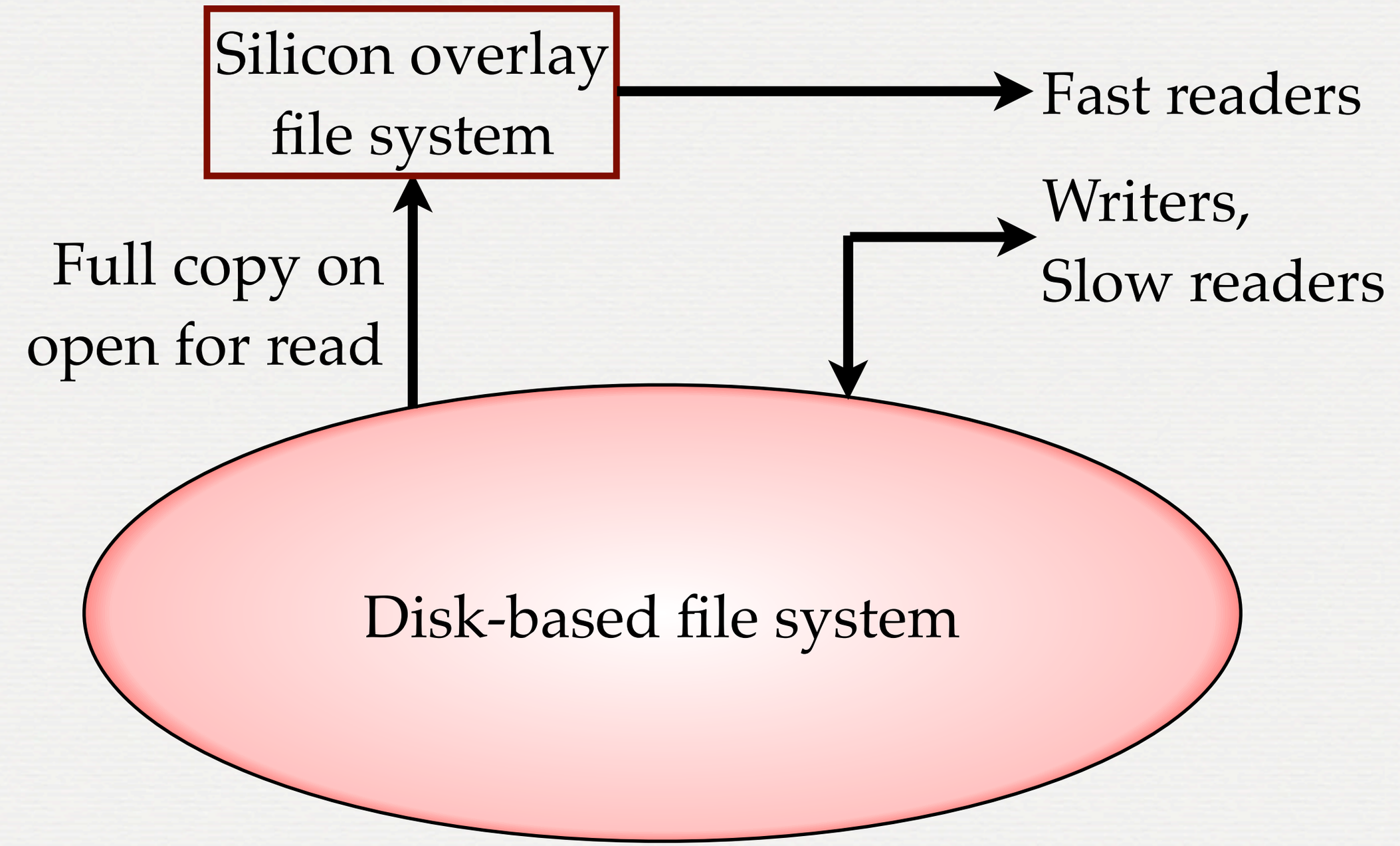
- Concurrent analysis access is ok: client jobs *act* compute bound from server's point of view.
- Concurrent whole-file (typically WAN) transfer slows all peers.

O(10ms) seek ~ 1.5MB of time. Not much ...



... unless interleaved with transfers ≲ 15MB.

MITIGATING SEEKS FOR SOME READERS



SECURITY BOTTLENECKS

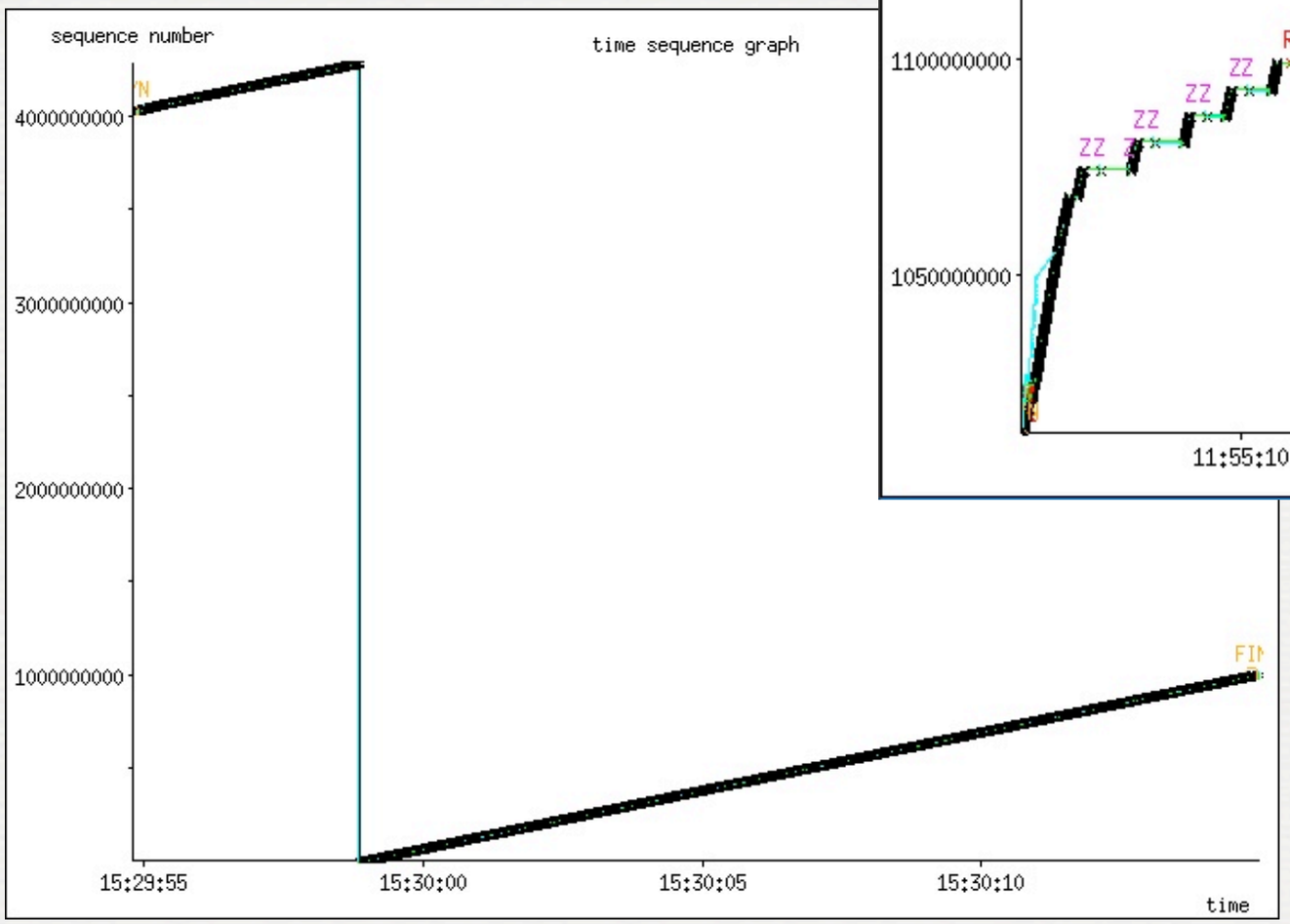
- GSI requires each endpoint to check several RSA signatures, an $O(N^2)$ operation each (for N bit keys).
- Each endpoint must also generate one signature, an $O(N^3)$ operation.
- To receive a delegation*, the server must do an $O(N^4)$ operation.

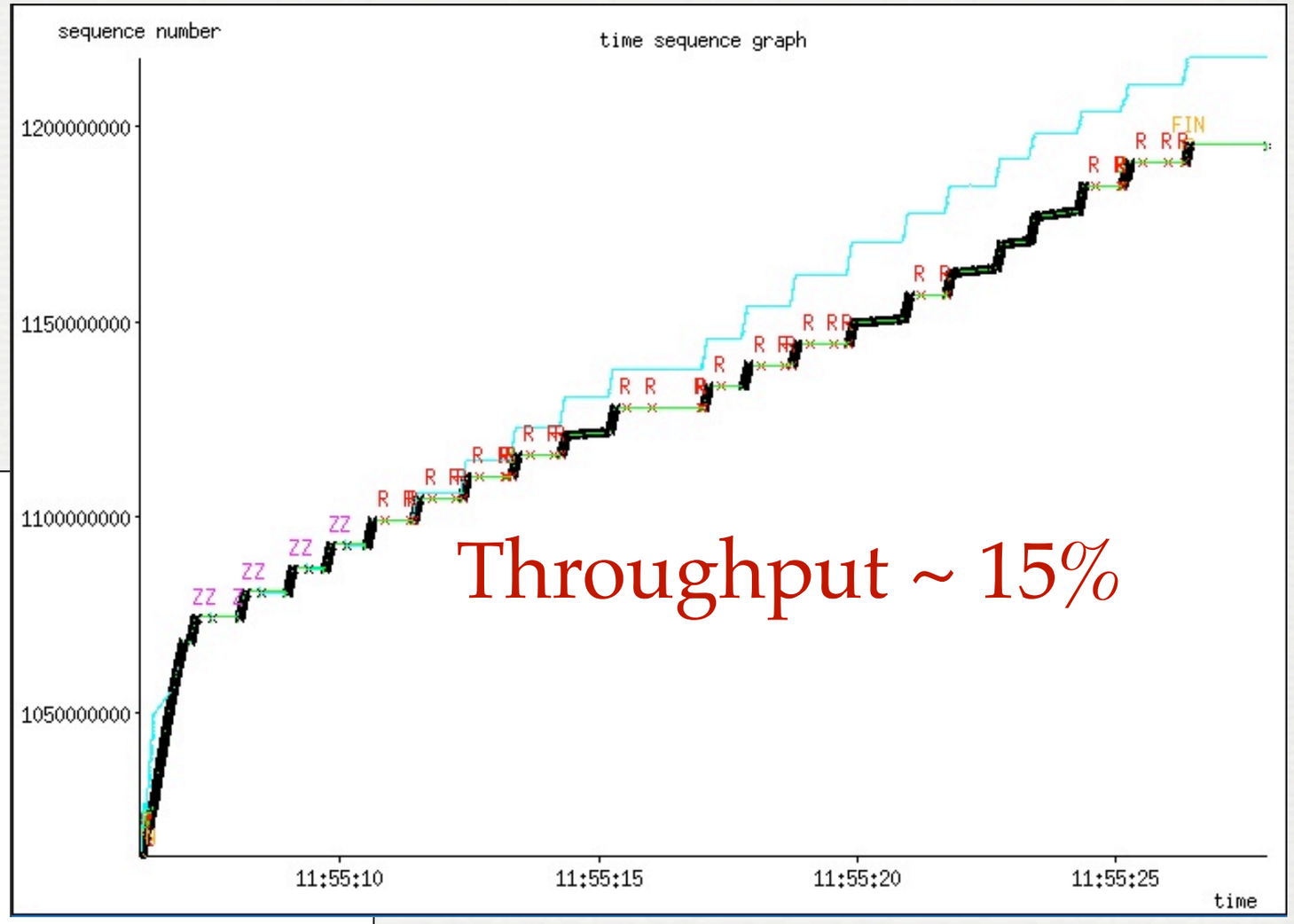
* The work is concentrated on the server.

- Kerberos does less work, and almost all on the client.

KERNEL BOTTLENECKS

Preempted network receiving→delayed TCP processing.



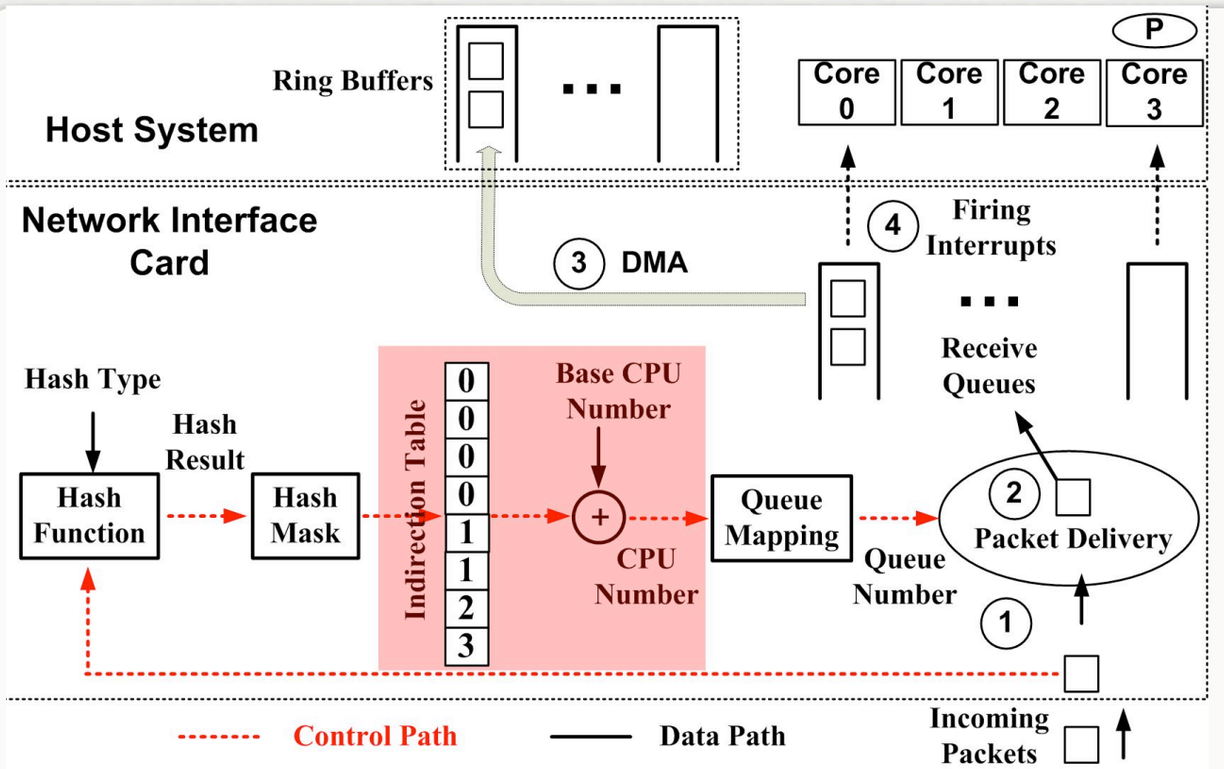


Throughput ~ 15%

Background load 10, CPU and disk only

No load

HARDWARE BOTTLENECKS

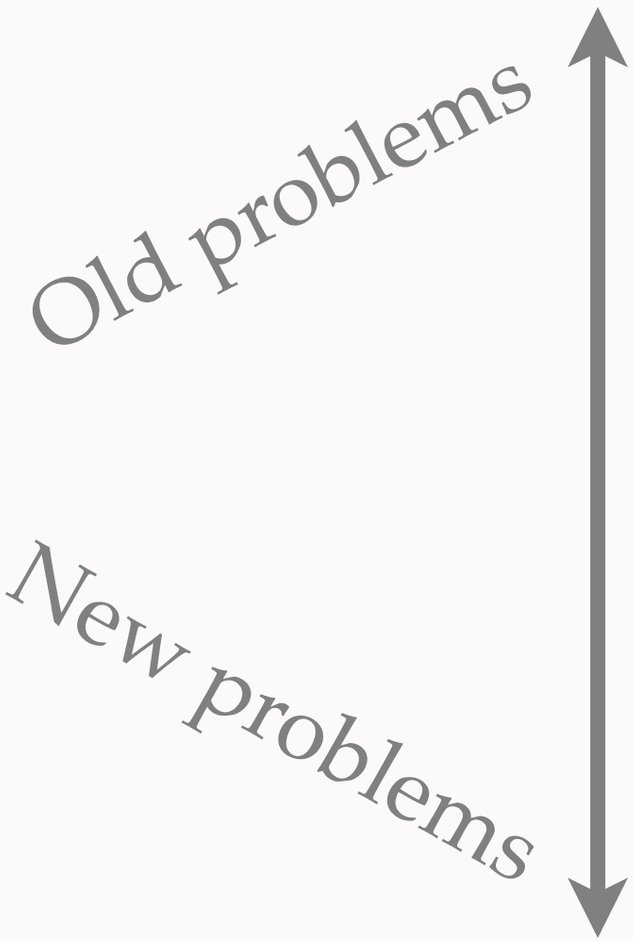


Advanced NIC functions

- RSS: multiple queues, distributed interrupts.
- Flow Director: packets of a flow pinned to one core.

* Linux sometimes does TCP in interrupt context, sometimes in process: large reordering happens.
* None of these pins the flow to the *Application!*

PUSHING THE BOTTLENECKS DOWN



* Network quality
* File system inefficiency
* Protocol implementation
* Network capacity and reliability
* Fragile middleware
* Kernel scheduling vs. protocol
* Multicore, multi-cache, interrupts