Essentials of Large Volume Data Management - from Practical Experience

George Purvis

MASS Data Manager

Met Office

There lies trouble ahead…

Once upon a time a Project Manager was tasked to go forth and procure a Storage Management System that was COTS, and scalable to a Petabyte over 5 years…and the budget was £1M!...

But….

Purpose

- To identify the main challenges to be addressed for large volume active archives, and to disentangle them, as far as this is possible, from technology

What is a Large Volume Active Archive?

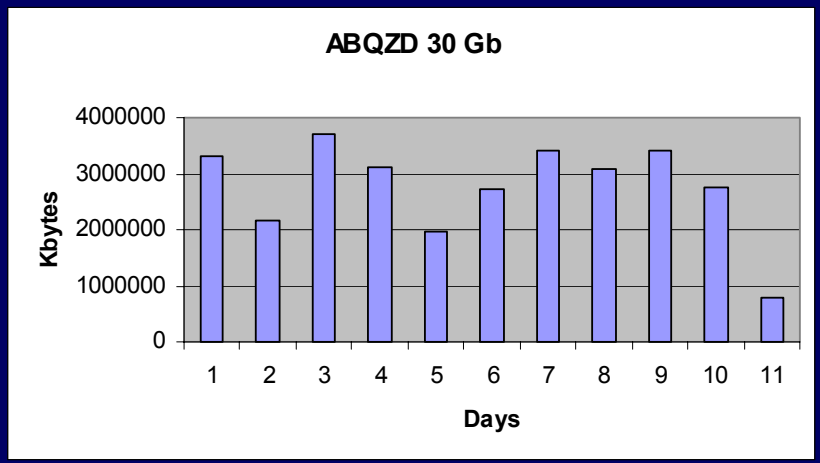
- **A large volume active archive is one where cost compels most of its data to reside on inexpensive (or tiers of, for very long lived data) media while *still in its active phase.***
- **Why? Because:**

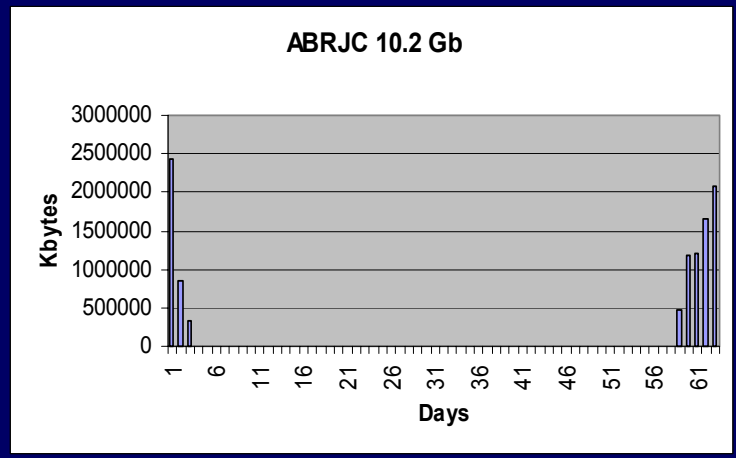
  - The collection of individual sets of data is extended in time, and sometimes open-ended

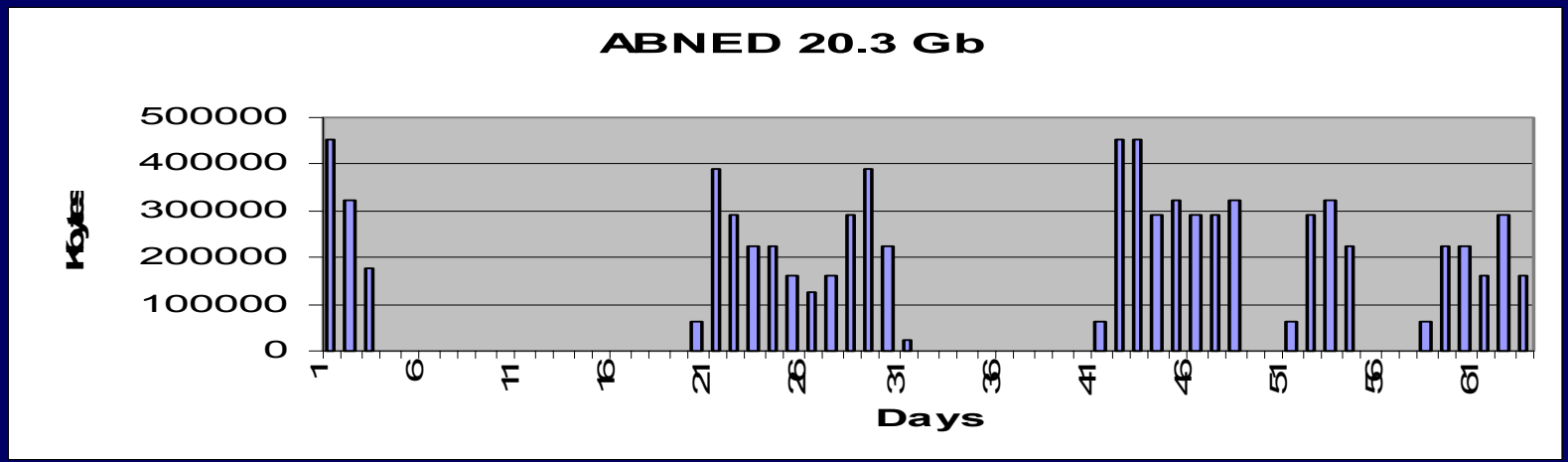
  - Data is mostly accessed when sets are complete

  - And many individual sets of data are being collected at any given time.

Some Typical "Active" Data Experiment = Set of data







Default Assumptions

- Individual sets of data which are extended in time are databases; OR…
- Individual sets of data are NOT extended in time; i.e. a “set” arrives all at once
- A “set” which arrives all at once is a file
- Files are independent units which are not parts of sets, and are managed as such…
- …typically by HSMs etc.

There are two default storage models (1):

- Databases
  - **which are sets of associated data, with powerful access methods**
  - **which reside on disk or…**
  - **are decomposed into files to reside offline**
  - **offline data has to be restored to disk to use the powerful access methods**

There are two default storage models (2):

- Files
  - **which are independent units**
  - **which can be on disk and offline…**
  - **and can be managed transparently (HSMs etc)**

Pros & Cons

- Databases
  - **are very good for retrieving sub-sets of associated data from; e.g. time series for Met data…**
  - **but do not scale beyond tens of terabytes**
- HSMs etc
  - **size doesn't matter**
  - **but no good for retrieving sub-sets of associated data from**

Requirement

- A Database which is scalable through tiers of storage media (including shelf and beyond), with an interface that forces users to build data structures that optimise retrieval from offline media
- Or something else…(ECMWF's DHS?)

An alternative?

Disk prices keep falling so…

- **you could have an ever growing collection of files accessible at disk I/O speeds (so data associations could be built at access time)**
- **Example: a growing collection of direct access type files (mini-databases)**
- **Problem (but not on disk?): whole files have to be accessed to get at their contents**
- **Or variations on this theme (ECMWF…?)**

Costs of Disk Storage

- Even RAID has to be backed-up, or the amount of RAID doubled, trebled etc.
  - **More cost either way**
- The running costs of disk are often overlooked
  - **Newer disks cost between £1,700 and £3,400 per 100Tb per year (including cooling costs)**
- The running costs of tape are about 1/72nd that of disk

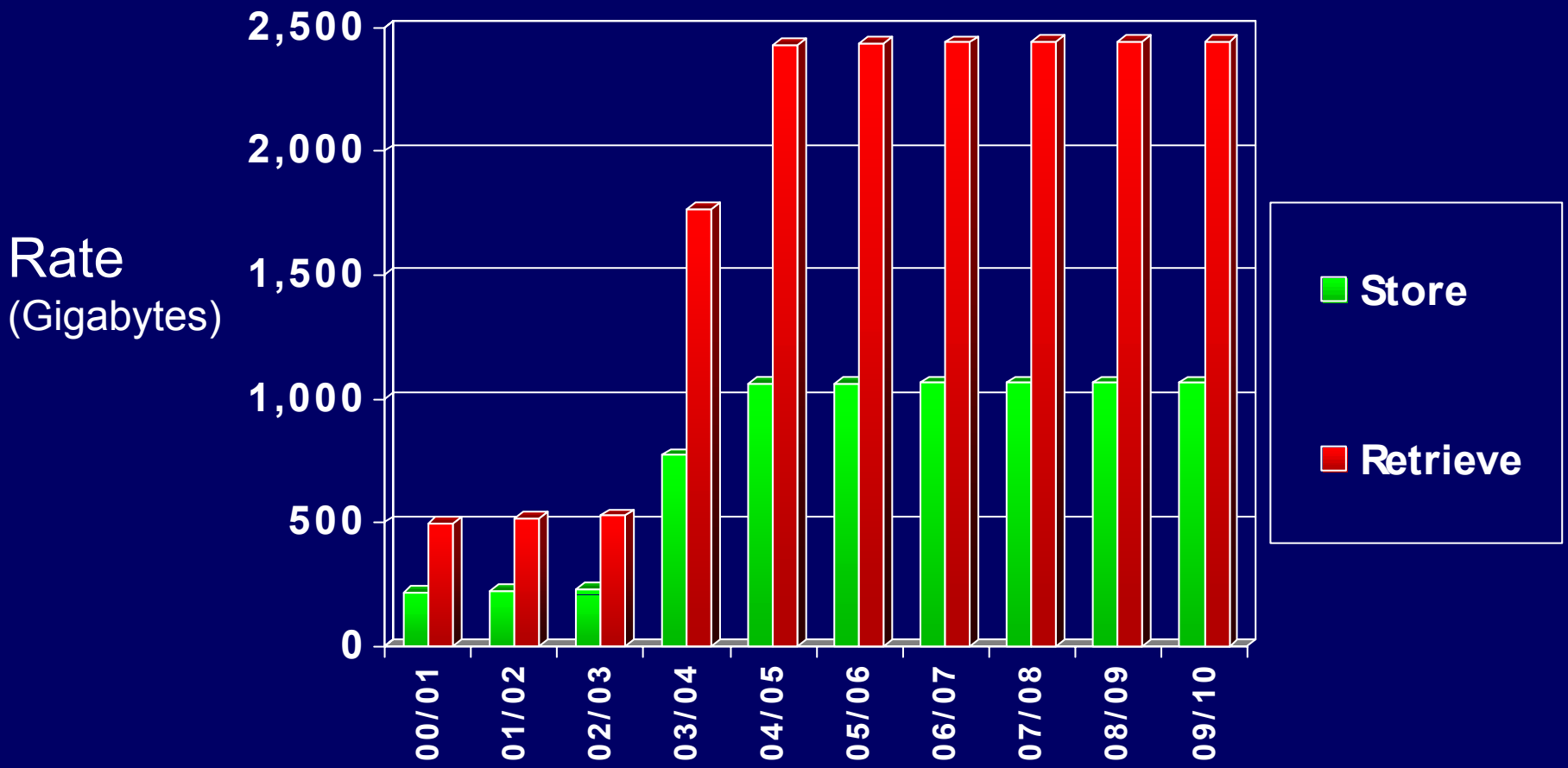
At present disk only is not an option

- This causes more problems for the “file” model than the database model because “files” have to be staged to disk to get at their content – which adds a layer of complexity
- And many files may have to be retrieved to build say, a time series comprising a small fraction of the data retrieved
- Files are independent – unless made not so, but this is another layer of complexity

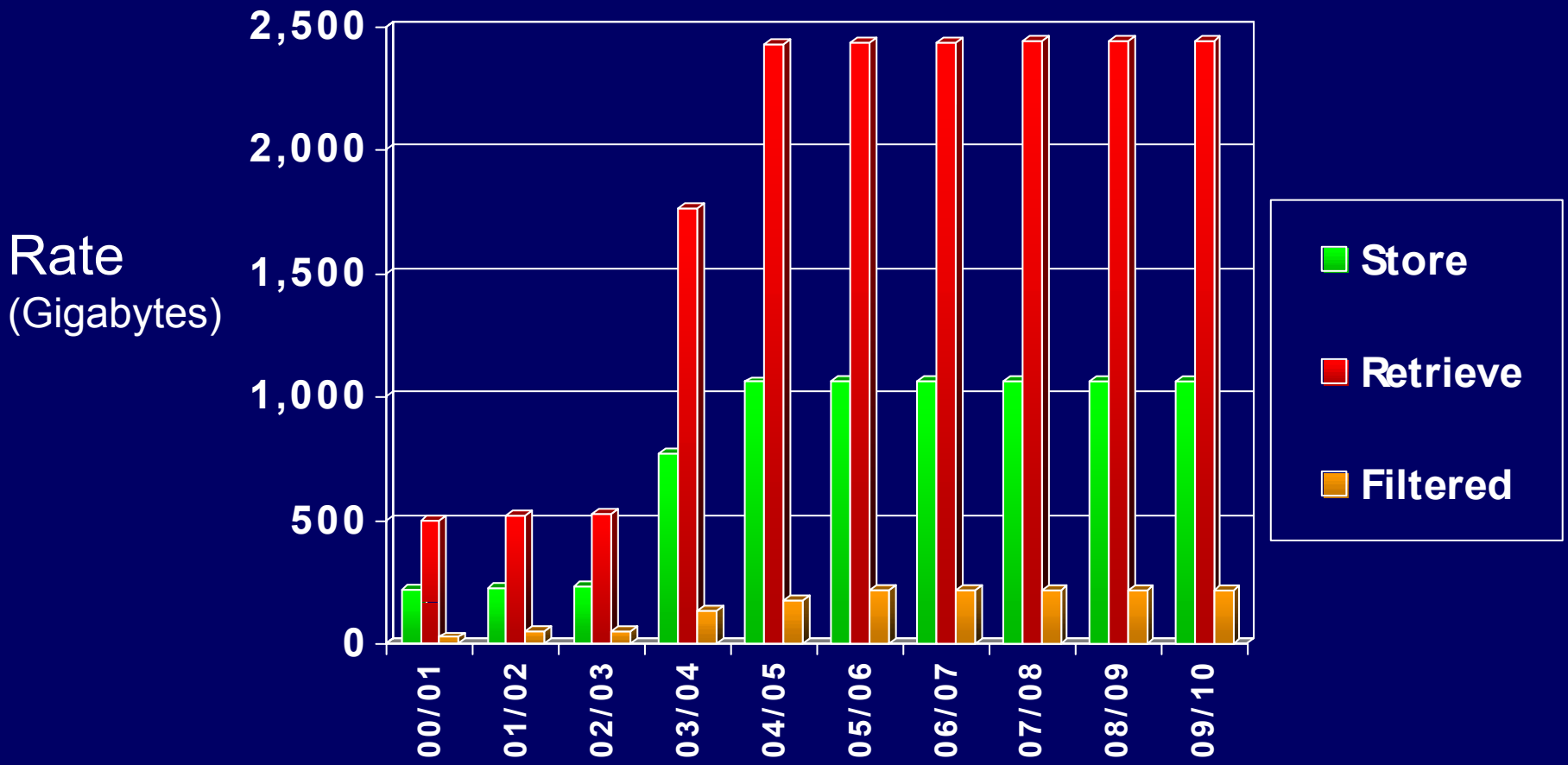
Some jargon and contrasts

- We call retrieving sub-sets of data from files Filtering
- Tables in a database equate to files
- Filtering is a one step process in a database
- Filtering is a many step process for files not in a database
- Two examples:…

Data Storage & Retrieval Rates (Daily) - for files not in a database



Data Storage & Retrieval Rates (Daily) - for files (tables) in a database



But there are common problems when non-disk tiers of storage are introduced

The IT industry evolved tiered storage haphazardly…and probably still do…

The evolution of tiered storage - first there was processing with limited memory

- Memory grew with processing power
  - **to accommodate the growth in data**
- …and became too expensive
- A cheaper tier, disk was added
- …and so on
- But operating systems never integrated the tiers beyond virtual memory…

Hence the problems we have today , which are…

- That storage management is technology driven (just add another layer)
- Badly integrated (carbuncles on carbuncles)
- And its cost is always underestimated

So, is closer Integration the way to go...?

There is a dark side to be aware of…

From Gartner w.r.t

- "…data replicated directly into the file system…While some vendors may appreciate this, others may not, as it means moving data out of their own proprietary store into one controlled by Microsoft."
  - From IT Week, 03/11/2003, p1 w.r.t Microsoft's next operating system: Longhorn

An Offer you can't refuse…?

IT salesman:

> "…I can sell you these new tapes which are 100 x the capacity of your present ones at half the cost of the new robot you would otherwise need to match your data growth…"

Note: IBM have announced a 1TB tape

Is this a bargain?

Some Problems Identified

To "cash-in" higher capacity media it has to be fully utilised. This is non-trivial!

Consider two extremes:

1) A giant tape of indefinite capacity

- **Advantages**
  - **One tape always mounted and one drive**
- **Disadvantages**
  - **As the tape fills the seek time becomes prohibitive**
  - **Lose the tape and the whole archive is lost**

Some Problems (continued)

2) Tiny tapes of one file capacity

- Advantages
  - **Cheap and works brilliantly for an archive of one file**
  - **100% utilisation**
- Disadvantages
  - **As file numbers increase you need more and more drives, then more and more Robots, then bigger and bigger buildings…**

Observation

Media utilisation is a barometer of the efficiency of a storage management system.

Experiment:

Decrease utilisation until Exchange rate peaks. At this point the utilisation value gives the optimum (one size fits all "activity") tape size (capacity)

The perfect storage management system has

a) **Retrievals to spec under normal workload**
b) **The minimum number of tapes and drives**
c) **The minimum sized robot working at its maximum exchange rate**
d) **The minimum amount of disk**
e) **Media to be 100% utilised**
f) **A transparent facility to relocate data to new media**
g) **To be scalable throughout its lifetime**
h) **…and to be the smallest system compatible with all these**

These are all linked.

Some Questions:

- What is “normal” workload?
  - **Is it peak load or “average” load?**
  - **Over what timescale?**
  - **Does it include relocation of data?**
- Is a scheduler essential?
  - **To even out peak loads, which will minimise system size**
- Can the system be sized with media futures in mind?
  - **And can it be incrementally scaled accordingly?**

Questions continued

- What is the optimum utilisation value?
  - **too high and one “size” tape fits all “activity” then**
    - » **access times will be too high**
    - » **system may have to be “bigger” to cater for data relocation load (to maintain high utilisation)**
  - **too low will:**
    - » **waste media**
    - » **increase the number of tapes and drives required**
    - » **Which will exceed robot capacity in one way or another (size and exchange rate)**

Utilisation can be maximised if…

- Tiers of varying capacity media are available
  - **so "active" data is on low capacity "quick" tapes**
  - **Less active data is on higher capacity "slow" tapes**
- Be careful…
  - **High capacity media put more data at risk**
  - **…which may force duplexing to be done**
  - **…which might double the number of media required (and system throughput etc. etc.)**

More questions

- **Delete data or not?**
  - **It's not worth deleting 10% - because you'd have to relocate 90% to get the dead space back**
  - **It may be worth deleting some bigger value**
- **"File" numbers**
  - **Can the system scale to the millions required?**
- **Data Granularity**
  - **"Bigger" means faster access and relocation but…**
  - **Higher network loads**
  - **Too "small", too slow!**

Relocate data

- But only when the time is right to:
  - **Minimise the amount moved while maintaining, or improving access times**
  - **Future proof (new media/technologies)**

Beware Vendor lock-in

- **Because to move data to new system in a short enough time (before the new system becomes and "old" system) requires too much investment in the old system**

An “infinitely” scalable system that avoids vendor lock-in?

- The “leap-frog” model
  - **Two, or more systems, to give enough time to migrate from system to system in a cost effective way**
  - **But rapid technology change might not give enough time before old technology is no longer supported**
- **BBC example**

Functional Test Requirements

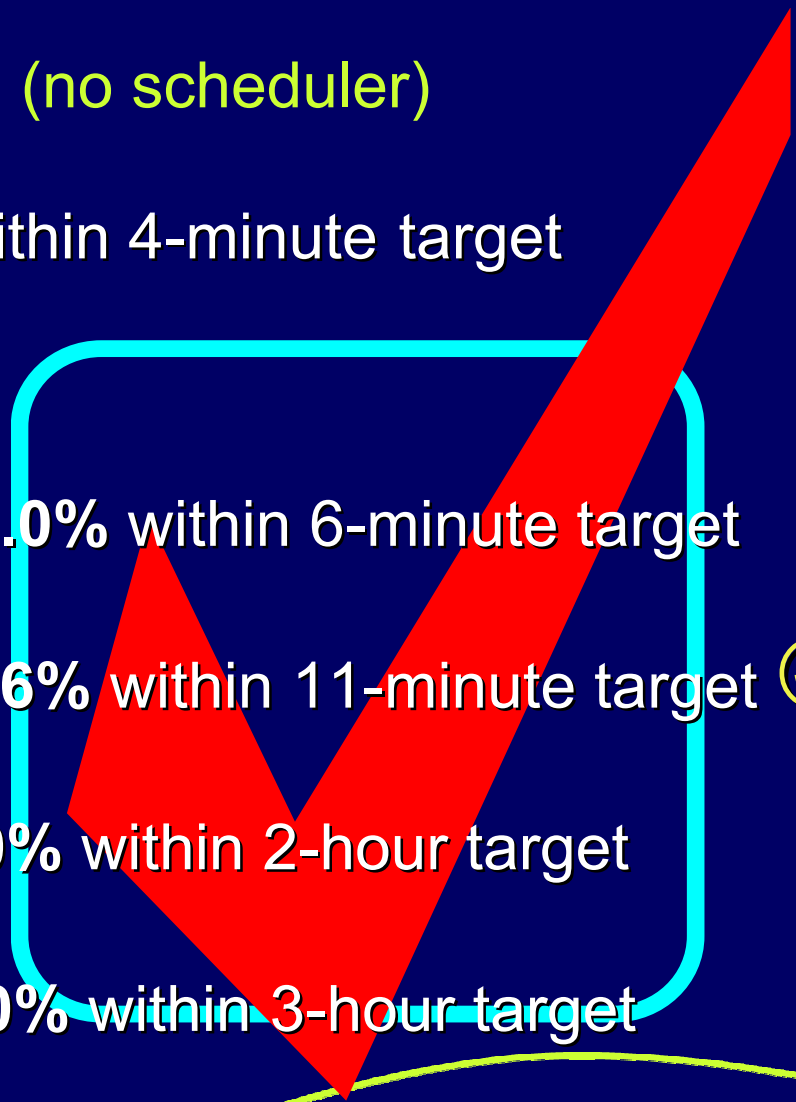
Class	Files read	Rate (h^{-1})	Filter (%)	MB. h^{-1}	Over (h)	In$^{(\dagger)}$ (95% level)
Tiny	1	18	100	563	6	6 min
Small	2-6	6	12.5	781	6	6 min
Medium	7-12	32	10	8,031	6	11 min
Large	25-50	55	5	123	24	2 h
Huge	75-150	45	5	50	24	3 h

DAY

24 h

Functional Test Results Summary: (no scheduler)

- **Storage:** **97%** within 4-minute target 😐
- **Retrieval:**
  - Tiny/Small - **98.0%** within 6-minute target ☺
  - Medium - **98.6%** within 11-minute target ☺
  - Large - **100%** within 2-hour target ☺☺
  - Huge - **100%** within 3-hour target ☺☺☺



Conclusion

- No COTS solution?
- Expensive
- Impossible long term?

The End

That's all for now folks!

… any questions?