ADSM's HSM at ECMWF

F. Dequenne

September 1999

ECMWF

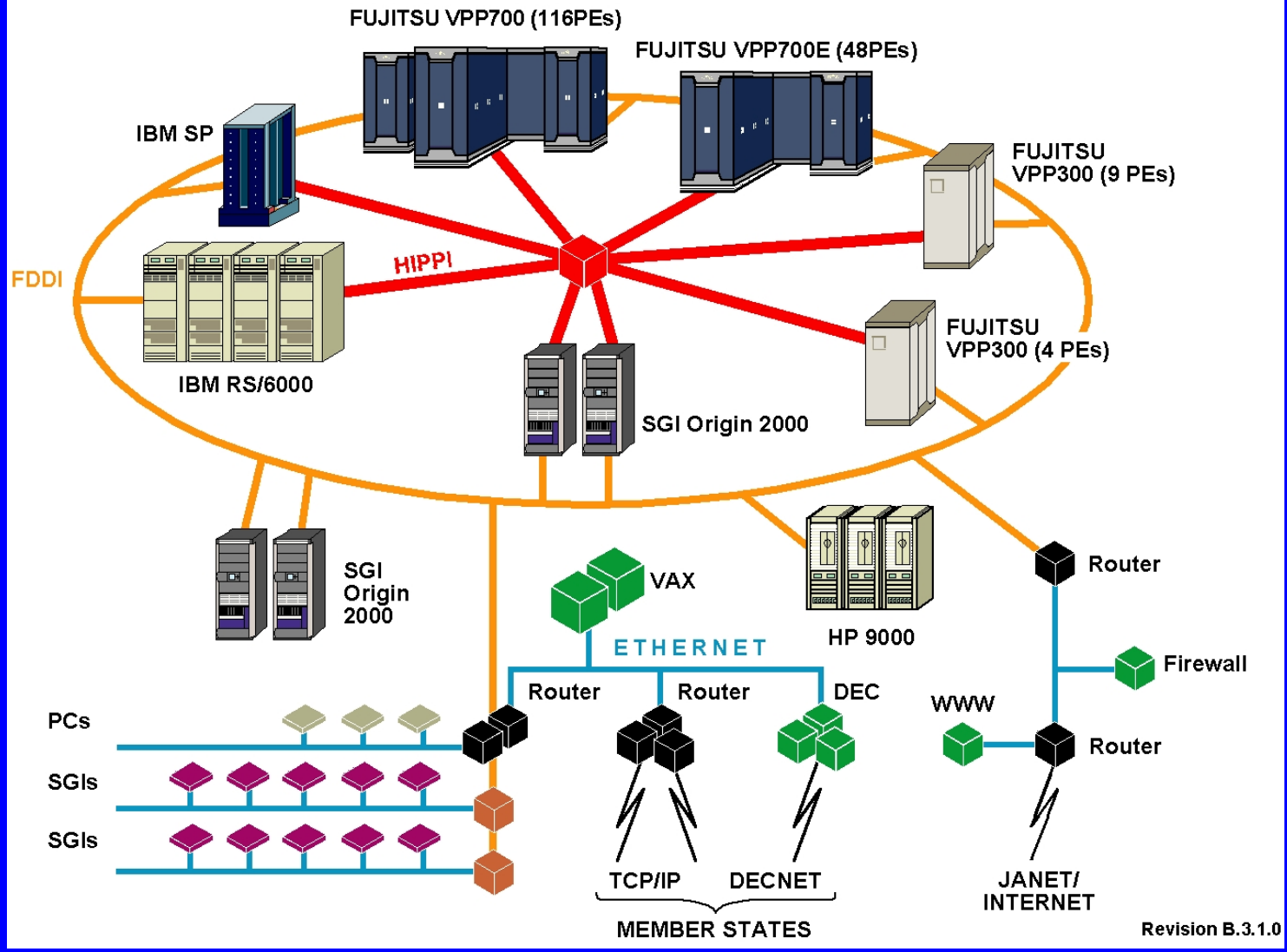
Introduction.

What is ECMWF?

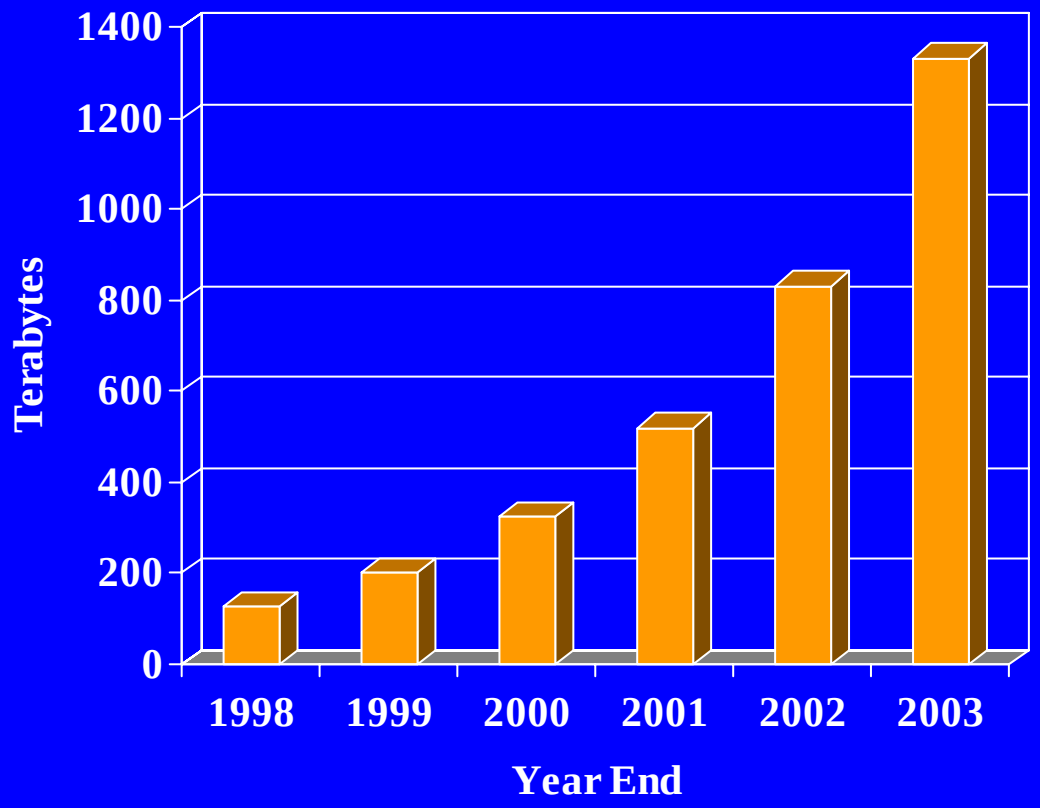
- *European Centre For Medium-range Weather Forecast.*

- International organisation founded by 18 European national weather organisations
- Our Job:
  - Daily production of 10-day global weather forecast
  - Meteorological research in a strong computational environment
  - Provision of access to a large database of meteorological information.
  - ...

ECMWF computer configuration - June 1999



Estimated growth in stored data



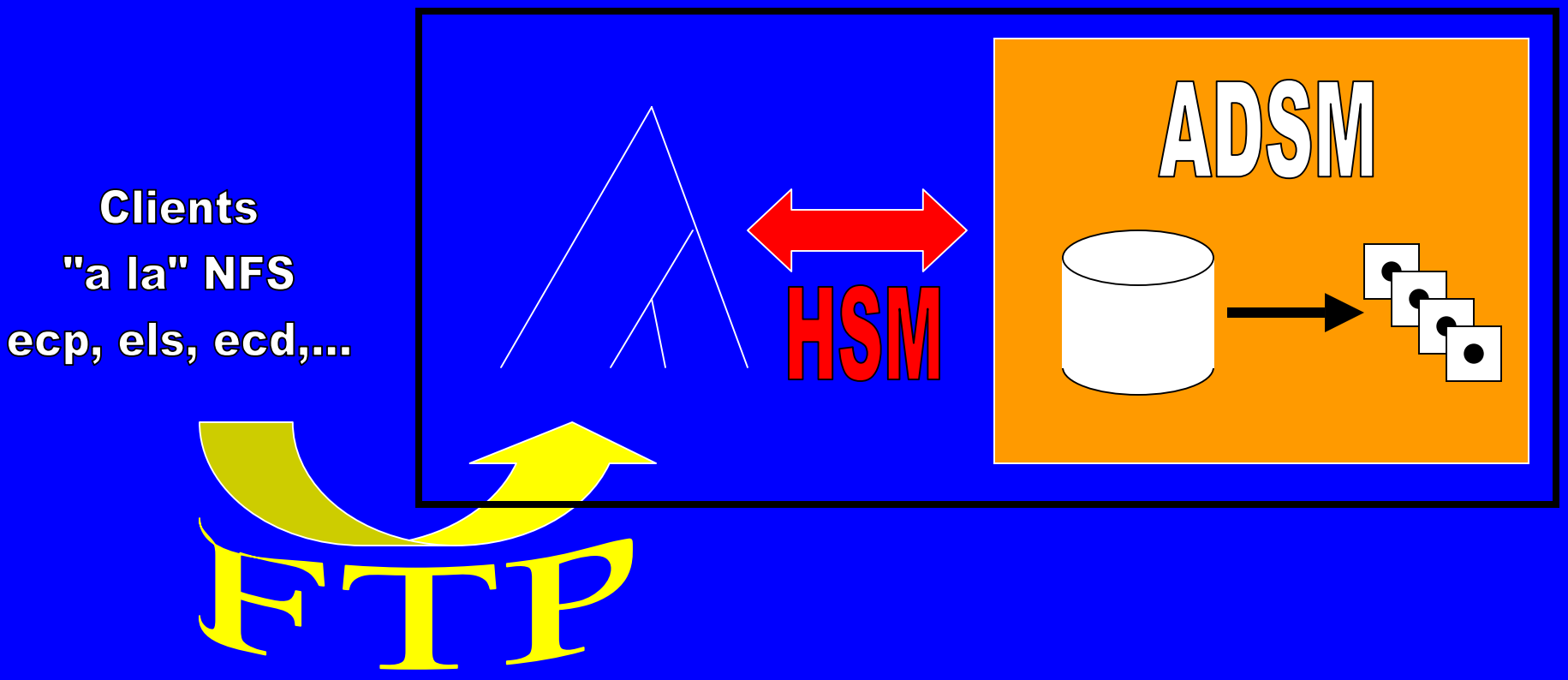
+- 60 % annual growth.

Data Handling System

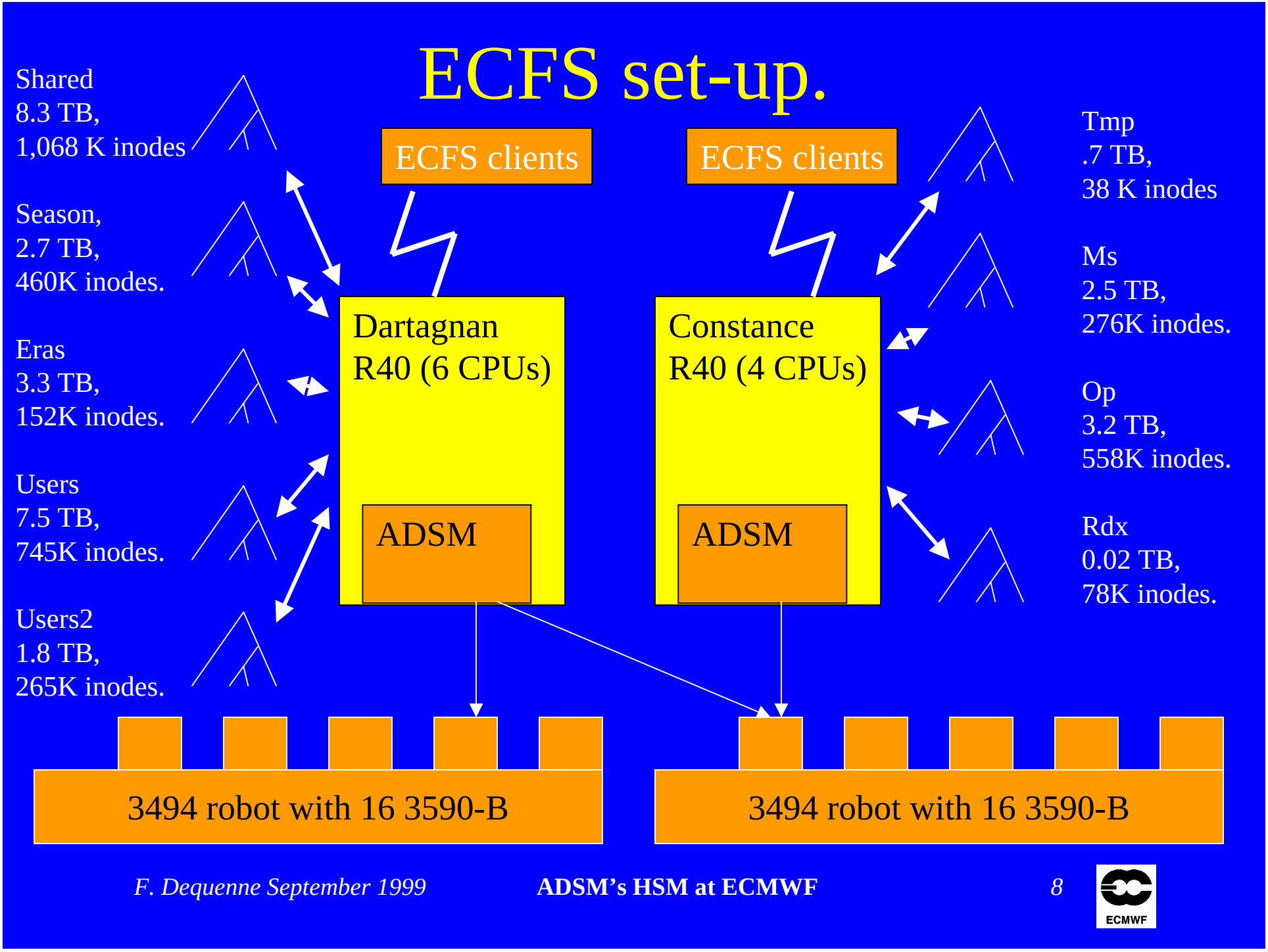
- Based on several ADSM servers running on RS/6000 and SP nodes.
- 2 main components:
  - Meteorological Archive and Retrieval System.
    - Application geared towards efficient storage and retrieval of meteorological data. 90TB +backups.
  - ECFS.
    - General purpose file archival system.

DHS Services: ECFS

- Ad-hoc files archiving system



ECFS set-up.


Shared
8.3 TB,
1,068 K inodes
Season,
2.7 TB,
460K inodes.
Eras
3.3 TB,
152K inodes.
Users
7.5 TB,
745K inodes.
Users2
1.8 TB,
265K inodes.
ECFS clients
ECFS clients
Dartagnan
R40 (6 CPUs)
Constance
R40 (4 CPUs)
ADSM
ADSM
Tmp
.7 TB,
38 K inodes
Ms
2.5 TB,
276K inodes.
Op
3.2 TB,
558K inodes.
Rdx
0.02 TB,
78K inodes.
3494 robot with 16 3590-B
3494 robot with 16 3590-B

Characteristics.

- 30 TB of data (+ backup copy)
- 3 Million files
- Files size: from a few bytes to 2GB (avg. 10 Mb).

- 2 RS/6000 R40
- 2 3494 libraries
- 32 Magstar (3590-B)tape drives
- 5500 Magstar tapes.

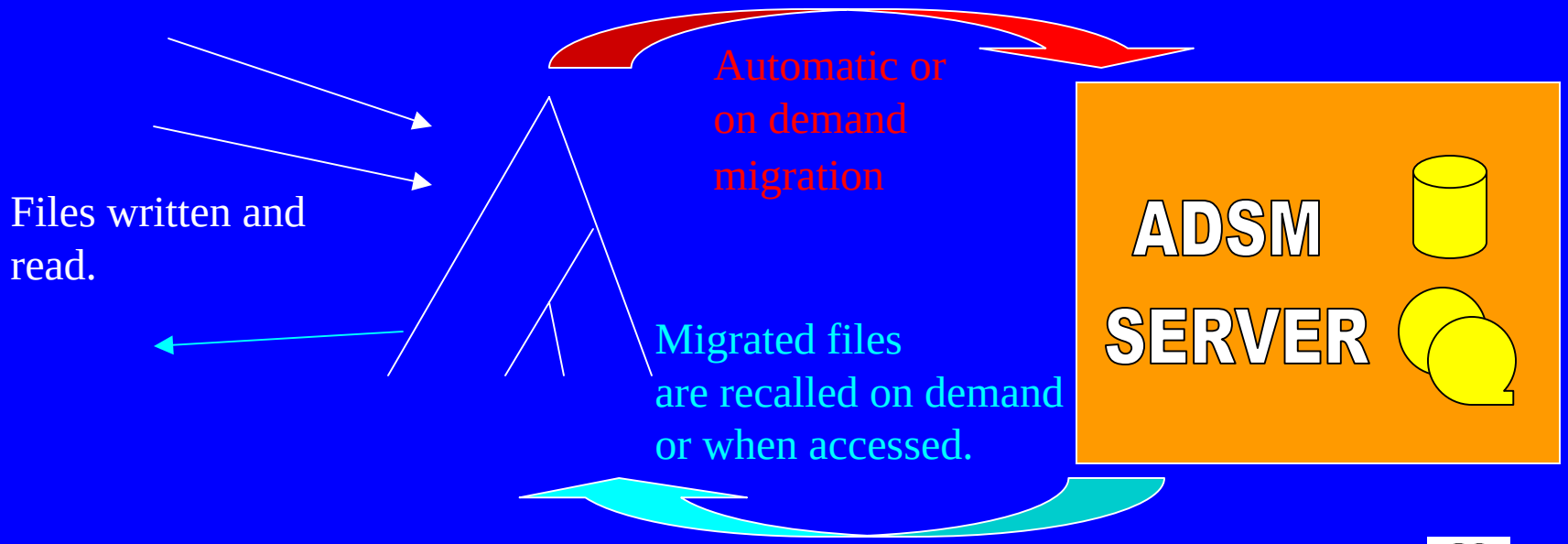
Characteristics.

- Some of the file systems are large.
- Active and dynamic file systems.
  - 1000 transactions/hour, 15GB/hour transfers.
  - 50GB added to then archives on a daily basis.
  - Some of the data his volatile, some will stay forever.
- 24/7 availability.
- Intense retrieval activity.

HSM, A crash course.

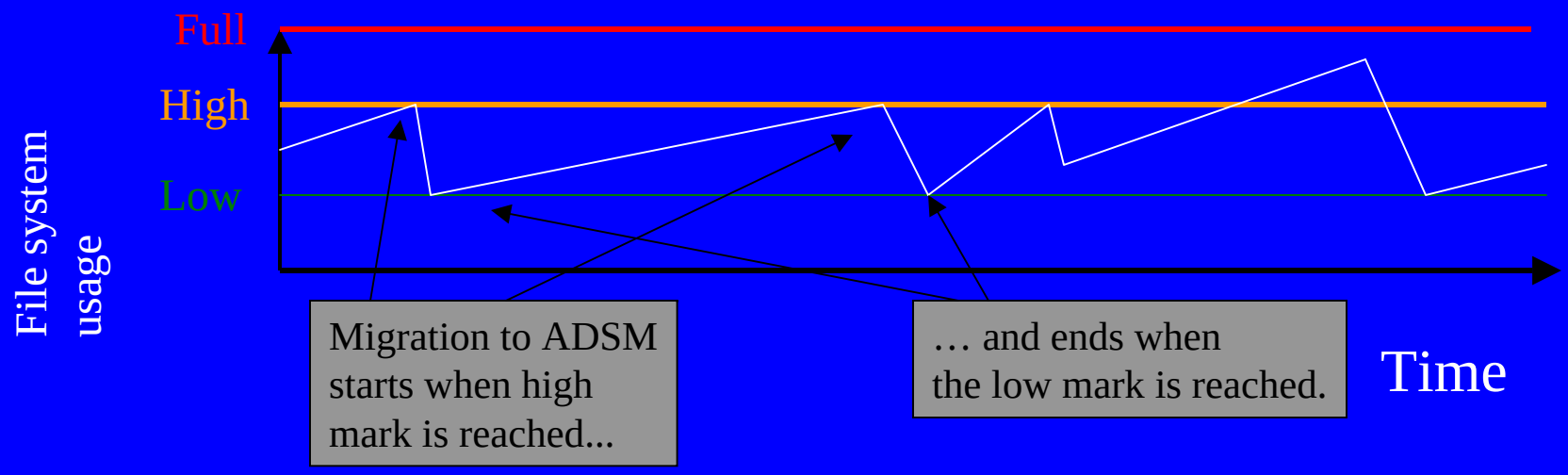
What is it?

- An ADSM client, composed of a kernel extension and a set of utilities.
- Use VFS to “enrich” the functionality of a standard JFS file system.



Automatic migration.

- For each HSM file system, a site can define high and low water marks, that control automatic migration to ADSM.



Some HSM commands.

- Dsmautomig
  - performs automatic migration to ADSM on High water mark or file system full conditions.
- Dsmreconcile
  - Creates a list of the files that can be migrated by dsmautomig.
- Dsmmigrate
  - Allow users or applications to migrate file on request.
- Dsmrecall
  - Allow users or application to re-stage files without opening them.

Some HSM concepts.

- Stubs
  - When a file is migrated to ADSM, a place holder, or stub, including metadata about the file, as well as the beginning of the file, is left in the file system.
- "Small file"
  - If a file is smaller than a stub, it will never be migrated.
- Candidates
  - Files that have been flagged by dsmreconcile as being "migratable" by dsmautomig. They are sorted by weight and stored in a candidate list.
- Premigration
  - Files with a copy in both the file system and ADSM.

HSM, The challenges.

Main benefits of HSM.

- Access to virtually huge disk space.
- Easy to deploy… at least initially.
- Appears as a standard JFS file system
  - but beware of catches (e.g. special characters).
- Applications using HSM need only to understand disk IOs.

Main issues.

- However, in a our environment, we have been confronted with several challenges:
  - Auto migration.
  - Dsmreconcile challenges.
  - Administration.

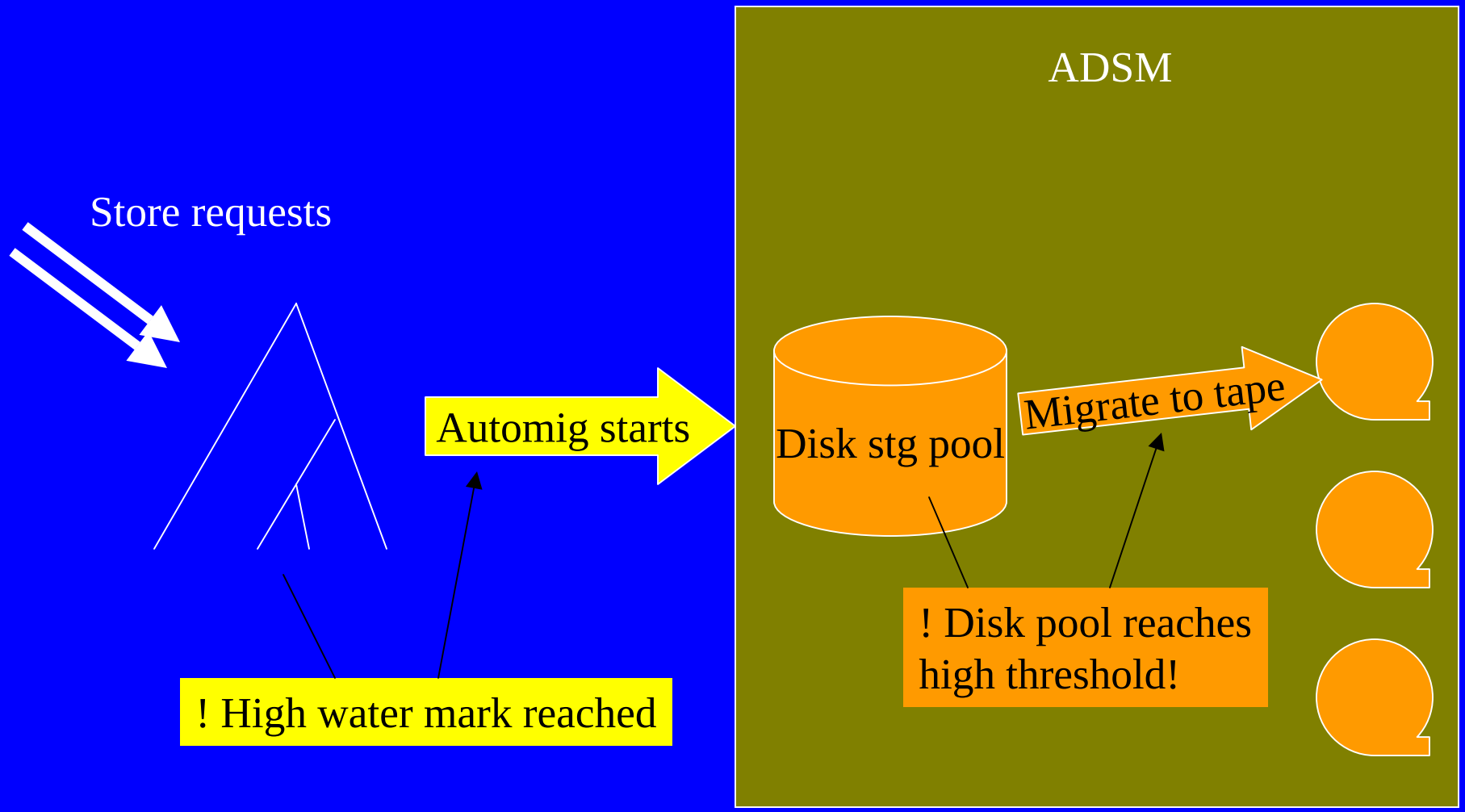
Dsmautomig issues

- Tool in charge of clearing space in a file system when the file system reaches its high water mark.
- Victim of retrieval centric ADSM server
- Single threaded.
- Requires a candidate list.

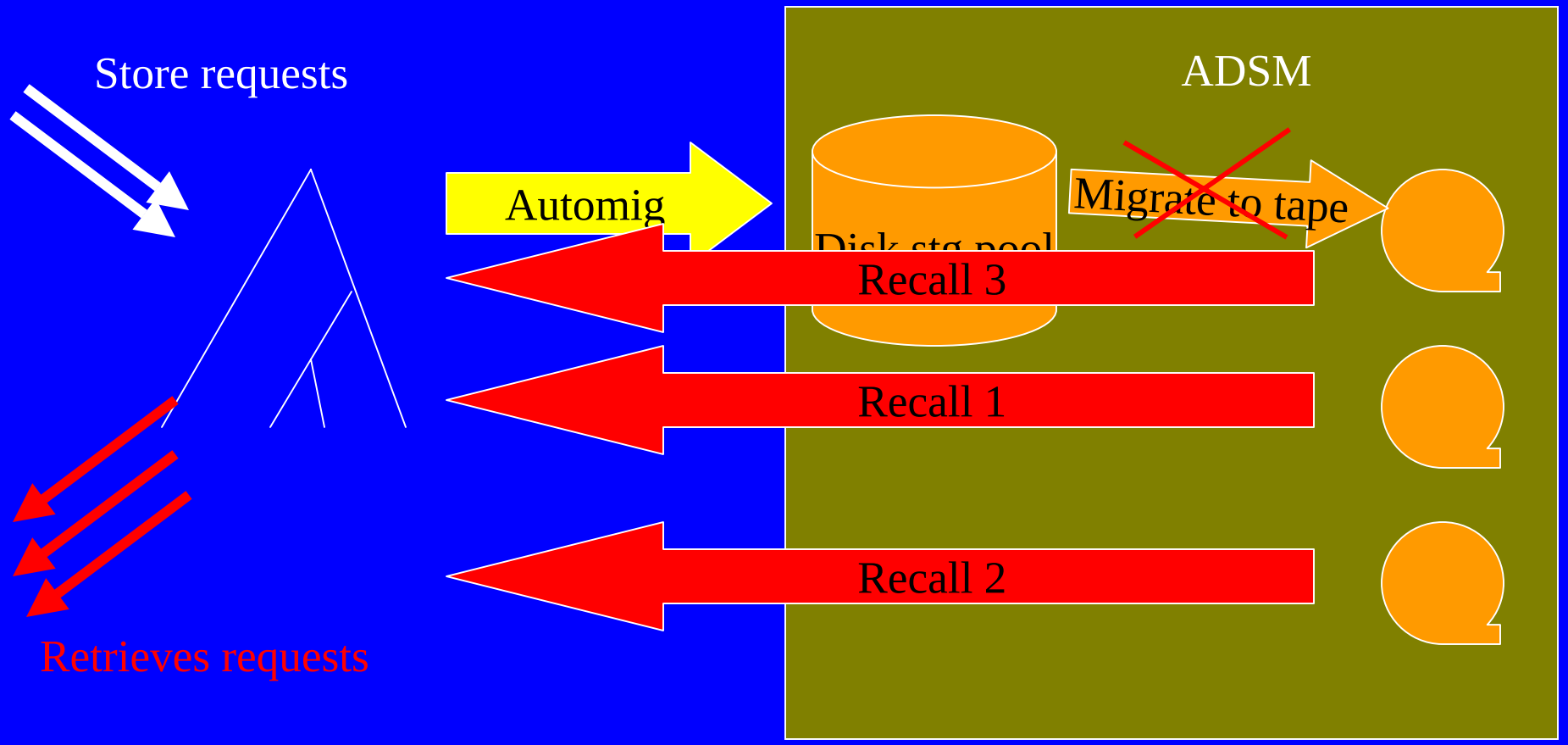
Retrieval centric ADSM server

- ADSM attach “hardcoded” priorities to sessions and processes.
- Retrievals have higher priority than saves.
- If tape drives involved, lower priority sessions can be cancelled to allow higher priority ones to run.

Retrieval centric server...



Retrieval centric server...



Retrieval centric server

- In a busy system,
- with a mixed load of HSM retrieves and archives,
- the file system will fill up, and automig will start.
- However, the underlying “mover to tapes”, sessions or processes, will be cancelled in order to allow retrievals (recalls) to take place…
- which will accelerate the filling of the file system.

- Eventually the whole file system will fill up.

Our workaround.

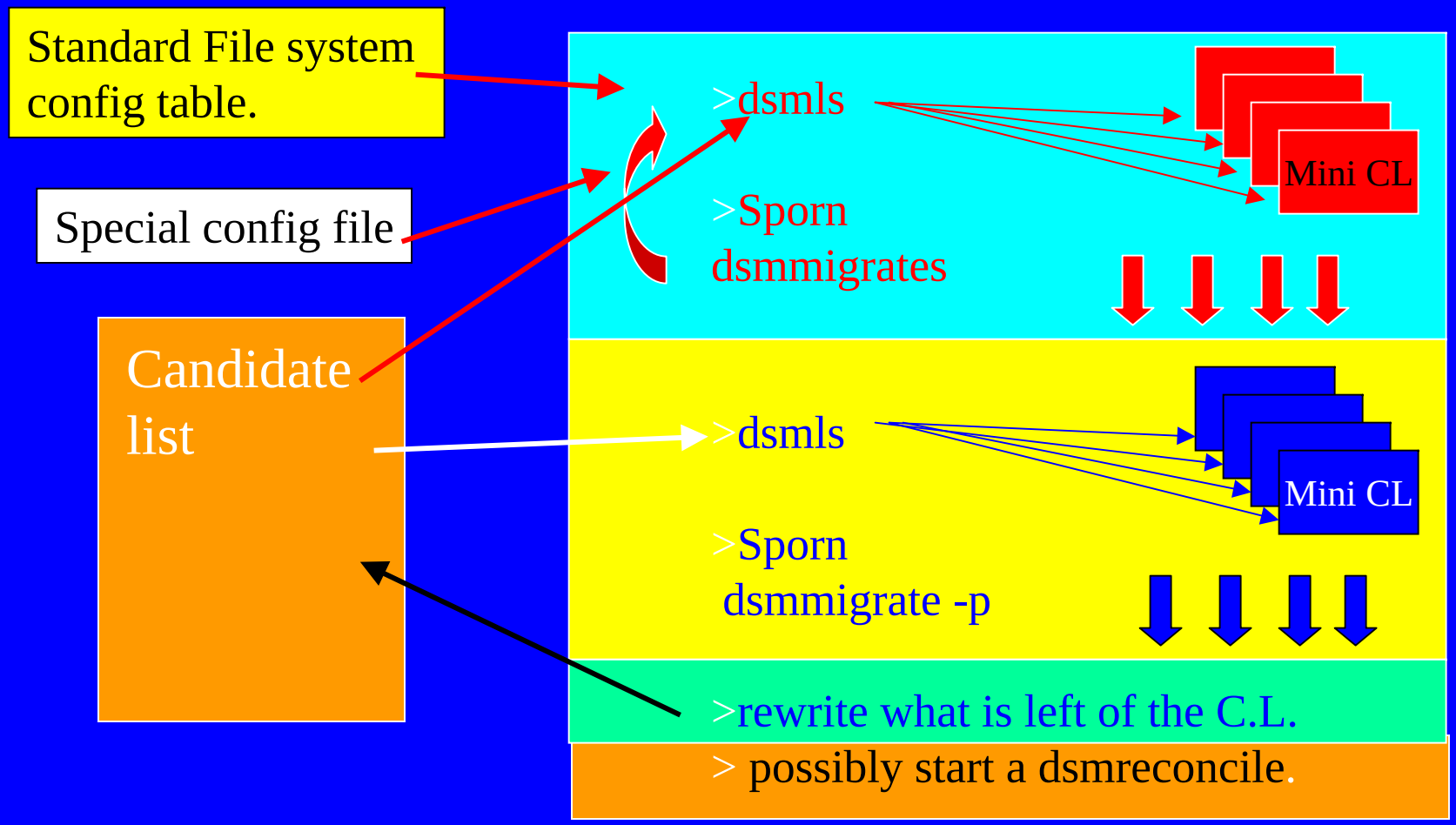
- We “solved” this by enforcing low number of dsmrecalls daemons, so that:
  - filling of the file system as a result of recalls is not overpowering.
  - We ensure that some tape drives are available for writing.
- We also make use of “migrate on close” when big files are being retrieved.
- This can generate large retrieval queues

Dsmautomig is single threaded.

- One file at a time is transferred from the HSM file system to ADSM. Rates seen are low (2-4 MB/s).
- At times of peak activity, automig is not able to empty fast enough a file system being bombarded by store and retrieve requests.

A multi threaded dsmautomig.

ECMWF wrote its own dsmautomig: 500 lines of Perl script.



Multithreaded dsmautomig pros and cons

- Up to seven concurrent streams.
- Automig runs really fast.
- We clear the candidate list of “dead” entries. (e.g. deleted or manually migrated files)
- Efficient call to dsmreconcile easily integrated.

- Requires disk staging at ADSM level.
- Use unpublished interfaces and tables structures.
- Need to be maintained.

HSM's Achilles heel

DSMRECONCILE

Dsmreconcile

- Indispensable utility used to perform 2 critical services.
  - 1) Create a candidate list.
    - Without this one automig does not know what files to migrate when the file system reaches the high thresholds.

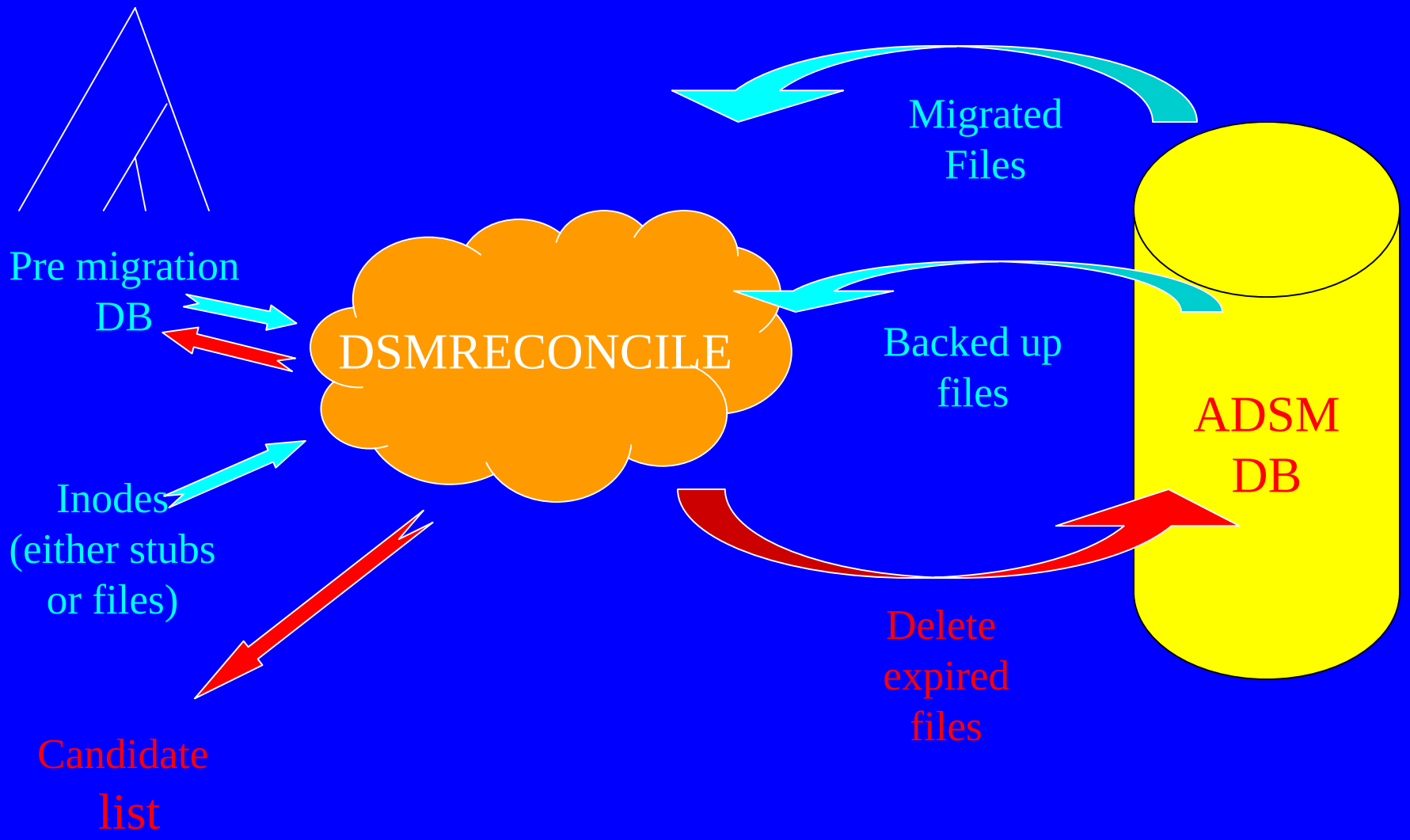
Dsmreconcile

- Indispensable utility used to perform 2 critical services.
  - 1) Create a candidate list.
  - 2) Ensure coherence between metadata existing
    - at file system level (e.g. stubs, premig DB, inodes)
    - in the ADSM DB (migrated file exist, backup has been done).

Dsmreconcile

- Indispensable utility used to perform 2 critical services.
  - 1) Create a candidate list.
  - 2) Ensure coherence between file system and ADSM metadata.
- Required to purge at ADSM level the data associated to files that have been deleted.
- Needed to recover from disaster/accidents scenarios.

dsmreconcile phases



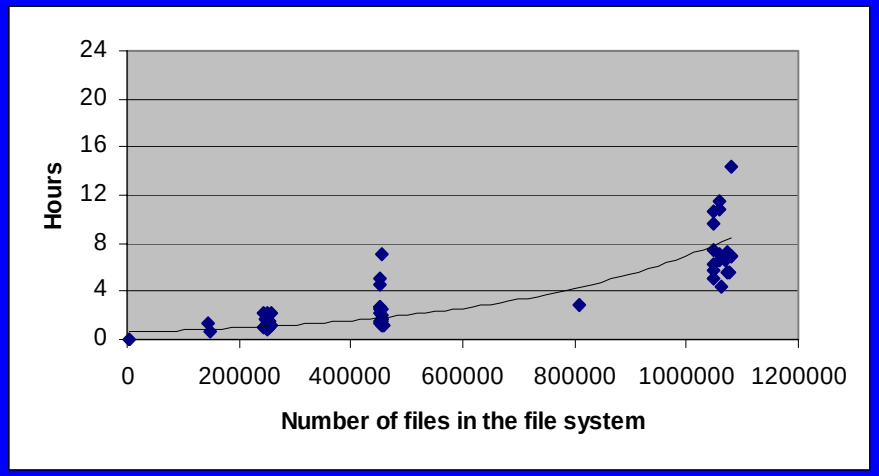
Reconcile issues.

- Full reconcile of a big file system:
  - Between 10 and 20 hours (sometimes over 1 day).
  - Resources intensive.
  - BLOCKS AUTO MIGRATION
- UNWORKABLE IN AN ENVIRONMENT WHERE
  - SEVERAL RECONCILES MAY NEED TO RUN DURING A SINGLE DAY,
  - MIGRATIONS NEED TO BE PERFORMED ON A REGULAR BASIS.

Candidate list reconciles

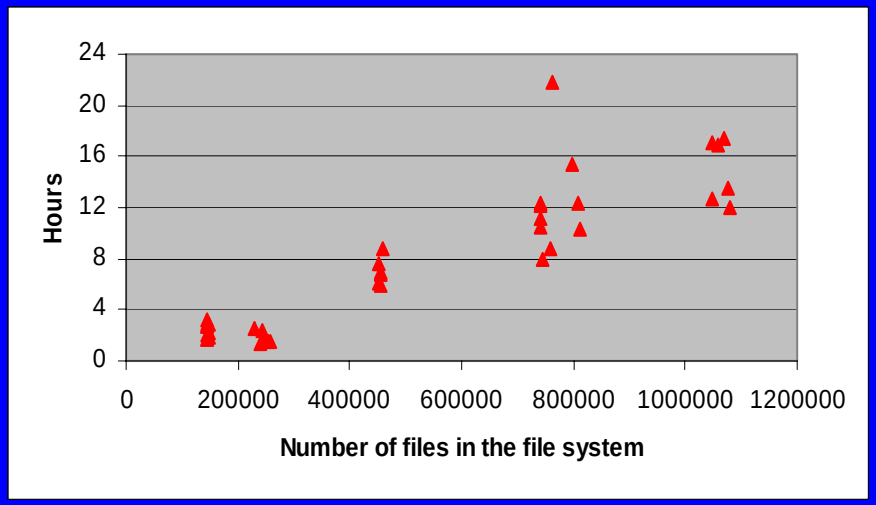
- dsmreconcile -c
- Candidate list reconciles are shorter, but
  - IBM's dsmautomig does not know about them. If you run out of candidates during a migration… good luck.
  - Full reconcile need to be run from time to time.

Reconciles timings.



Candidate list reconciles

Full reconciles

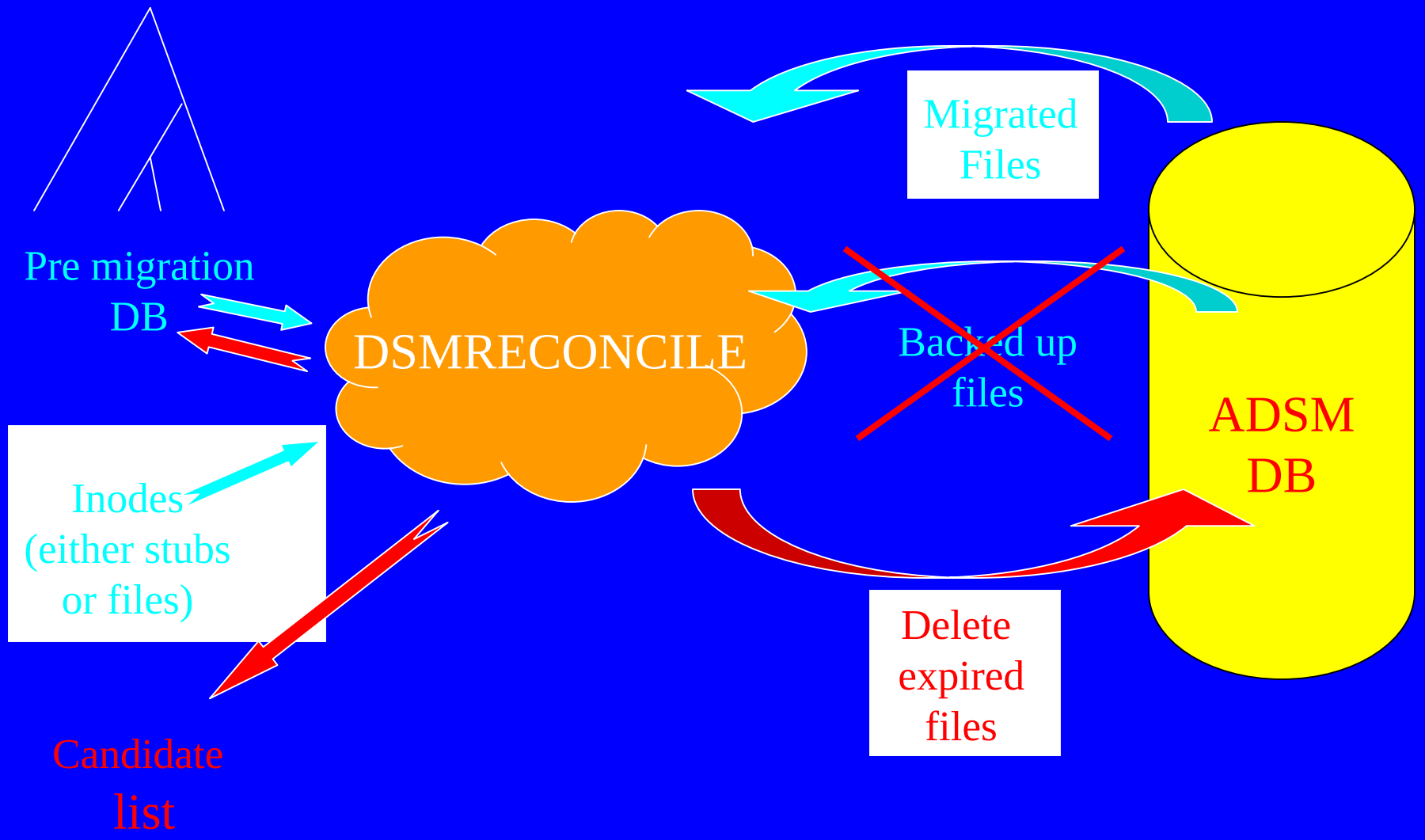


Dsmreconcile runs performed between 17/8/99 and 21/9/99

Full reconcile phases

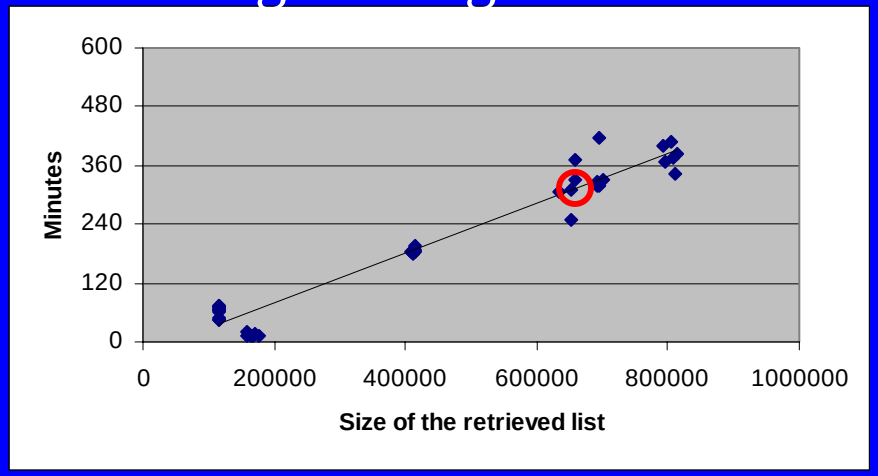
- The premigration database is read
  - Not very consequent. A few minutes at the most.
- The ADSM list of migrated files is obtained.
  - This can take several hours.
- The list of ADSM backed-up files is obtained.
  - We do not back-up in the same ADSM server, so negligible.
- The file system is traversed.
  - This can take several hours.
- Stale premigrated entries are removed.
  - Not very consequent.
- Expired files are removed from ADSM.
  - This can take several hours.

dsmreconcile phases

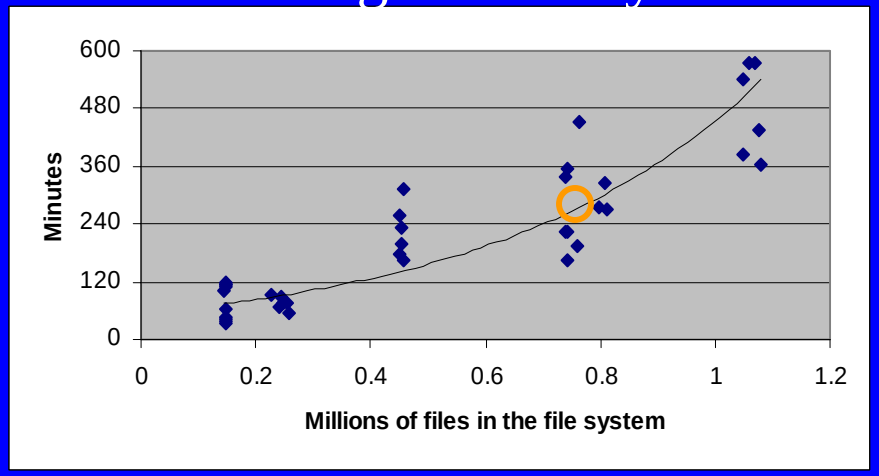


Full reconcile timings

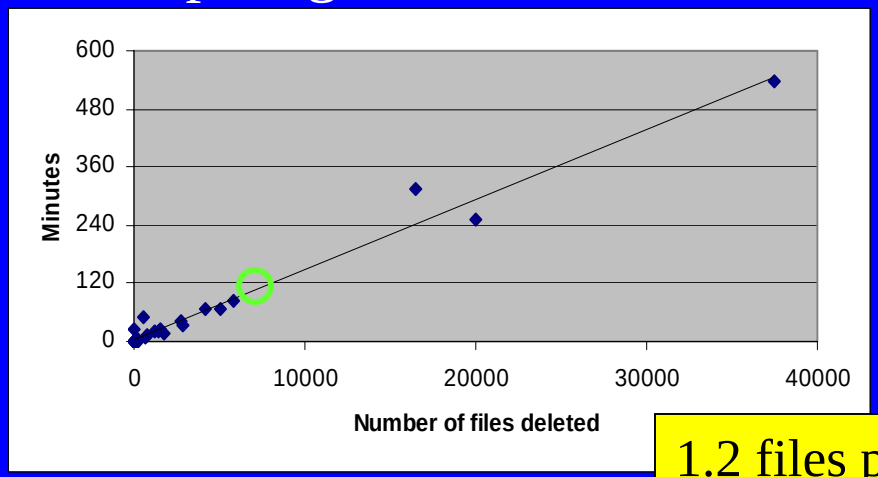
Getting the migrated files list



Traversing the file system



Expiring files in ADSM.



Example:
One file system of 750,000 files,
of which 650,000 are migrated, and
7,500 need to be deleted:

5.5 hours + 4.5 hours + 2 hours.

11 hours run.

1.2 files per second!

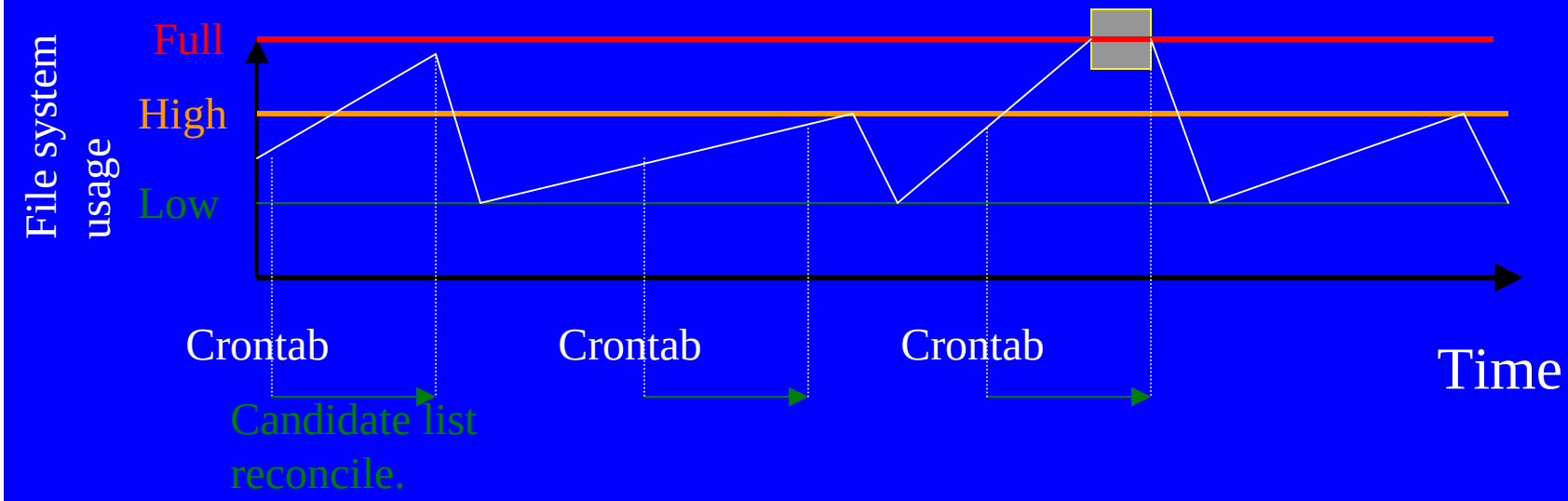
How to avoid the reconcile trap?

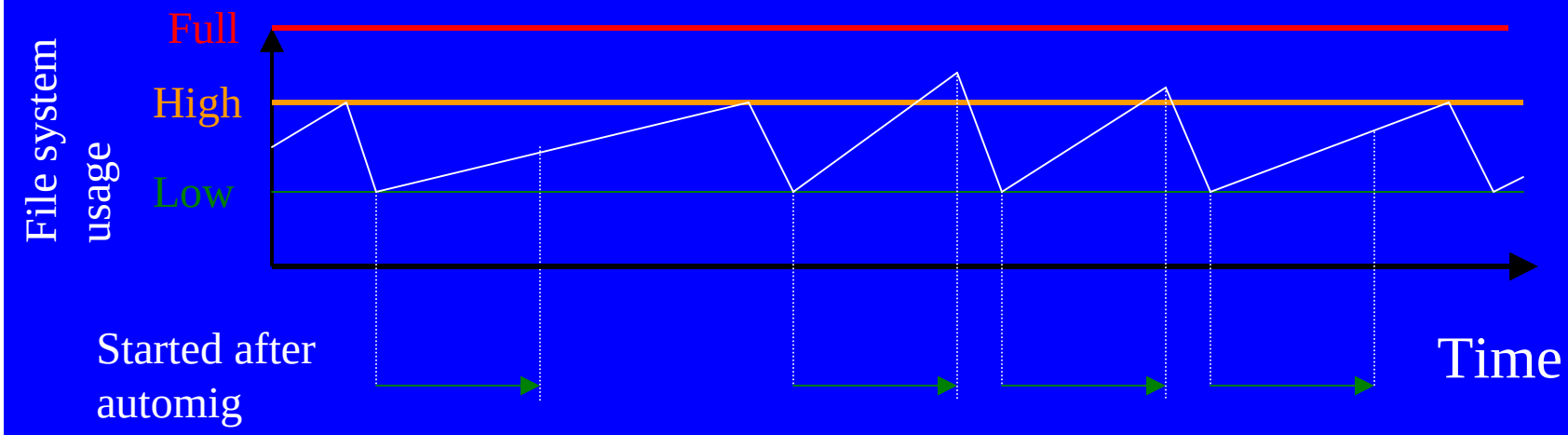
- Only perform full reconcile once or twice a week, during less busy time (night or Week-end)
- Otherwise, recreate the candidate list by using the short version of reconcile (-c option).
- Limit the need for automig runs by using “dsmmigrate on close” (only possible if the user application co-operate)

Starting reconciles.

- Use Crontab?
  - Dsmreconcile and dsmautomig can not run at the same time.
  - Often, reconcile runs are aborted because automig is already running.
  - Reconcile starts just before an automig is due…
- Too often, the file systems get full before a candidate list is recreated!

When to start reconcile?





Impacts

- Blocks auto-migration for long period of time.
- Limits the number of files stored in a file system.
- Forces us to migrate files straight after they are used.
- Does not allow us to run “normal” dsmautomig.

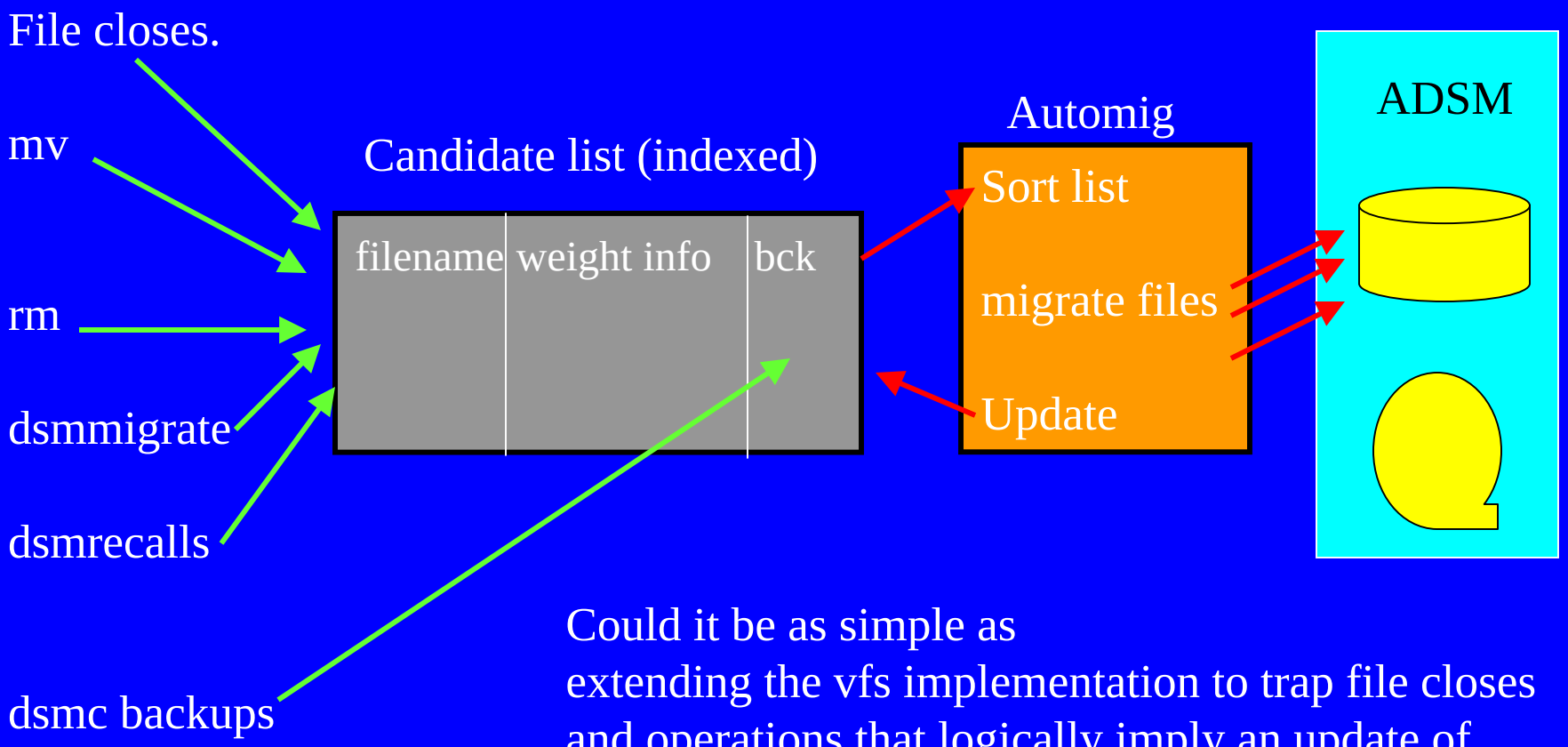
Other problems linked to reconcile

- Forced to keep high water mark low.
- In some conditions, recalls and migrates can be locked out.
- Delay reconcile until they are really needed. This result in Loss of candidate list prioritisation.

In an ideal world...

- The Candidate list should be maintained dynamically.
- ADSM DB and HSM file space should stay synchronised.
- The dsmreconcile utility use should be limited to recovery scenarios.

Maintain the candidate list dynamically



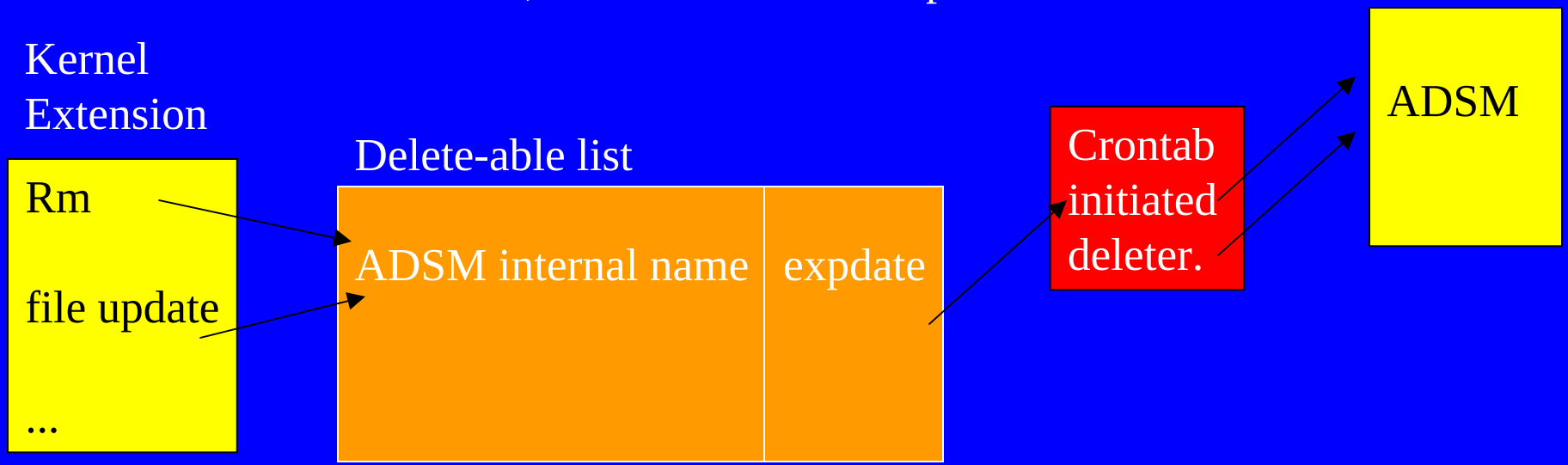
Could it be as simple as extending the vfs implementation to trap file closes and operations that logically imply an update of the candidate list?

Dsmreconcile: reducing the need for full reconciles.

The main reason to run full reconcile:

Purging in ADSM the files that were deleted/modified in the file system.

Instead, could the next concept be used?



When full reconcile is due.

- Reconcile problem: It blocks activity to get a perfect image of ADSM and the file system.

- 2 solutions:
  - partial reconciles, that only look at part of a file system (difficult)
  - Accept that the result of reconcile is ESSENTIALLY correct.

“Fuzzy” reconciles.

- To leave a few inconsistencies between an HSM file system and and its ADSM server is OK… AS LONG AS
  - only safe delete are performed,
  - the inconsistencies are rectified at a later run.

Administration issues.

- Configuration
- Day to day work.

Configuration.

- Defining the right set-up from the start is crucial.
  - How many file systems?
  - On which platform?
  - Using which ADSM servers?
- Extremely difficult to change a posteriori.

Number of file systems

- Just a few? Then their size will grow too large, and they will become un-reconciliable.

- Many?
  - More difficult to manage.
  - How to split them so that load is balanced?
  - Requires division and possibly inefficient use of disk cache.

ECMWF's approach.

- Initially:
  - one file system for users private data.
  - One file system for data shared between users
  - very few file systems for special projects.

  - Each of these with very large disk cache associated to them.

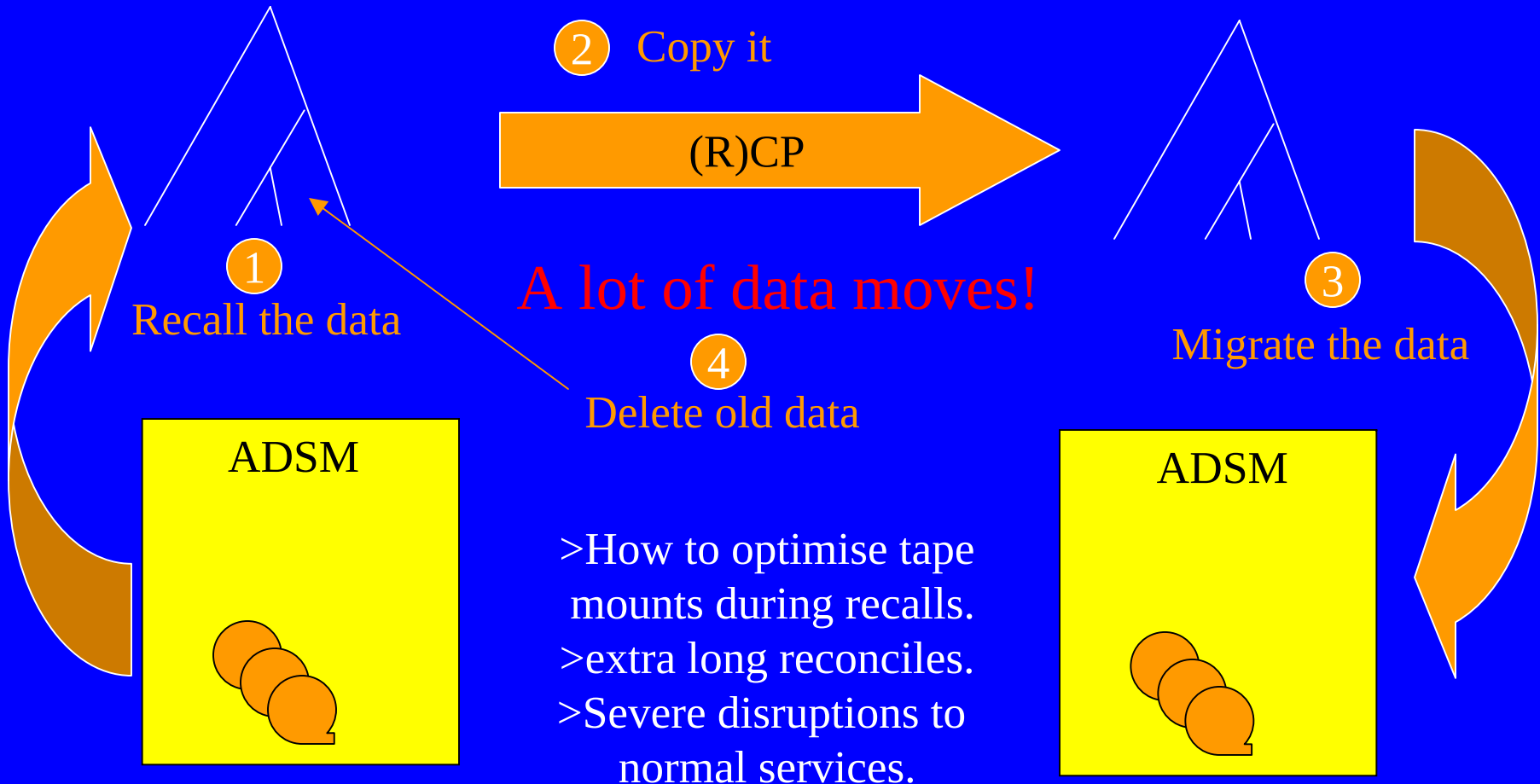
- Dsmreconcile blew this approach.

ECMWF's approach:

- Now:
  - less than 500,000 files in a file system.
    - Try to organise the file systems to balance transfer activity.
    - Keep similar type of data together.
  - The users and shared file systems had to be split.

- But…
  - requires front end diverting users request to appropriate file system
  - Split operation is very resources and time consuming.

To split a file system…



>How to optimise tape mounts during recalls.
>extra long reconciles.
>Severe disruptions to normal services.
>Need to be done with minimal denial of service.

Day to day Administration.

- System visibility.
  - Not always obvious why a recall is delayed.
- Tame large surge in activity.
  - "Let's store these few hundreds GB in ECFS"… (some users)
    - … and leave an HSM file system completely overwhelmed!
- Load balancing.
  - Today's super active file system could be dead quiet for the next two weeks.
- Reorganisation of the file systems.
  - Move directories sub-trees between file systems, plan and organise for new file systems to be used.
- Group many small files in one large tar file.

Conclusion

What we learned.

- Better served by home made dsmautomig
- Reduce as much as possible use of full reconciles.
- Any single file system becomes difficult to manage over 1/2 million files, and unmanageable over 800,000.
- Get it right the first time. Reorganising an HSM environment is no easy task.
- Avoid as much as possible to keep small files in an HSM file system.
- Do not back-up to the same server.

HSM works well...

- If the file systems are not very big,
- In a light/medium load environment.
- If batch utilities can run during off-peak periods.
- If recalls are not too frequent.

Where we are..

- 3.5 Millions files, 30 Terabytes…
  - 3 years ago, we never thought that we could stretch HSM to reach these peaks!
- However, we have reached the limits of what can be done with our current hardware, and the version of HSM that we run today.
- This requires a lot of Tender Loving Care.

Future.

- Our R40s will be replaced next year by more powerful machines, providing better IO bandwidth.

- By the end of next year, after 5 years of use of ADSM based solutions, we will also re-evaluate the various HSM and archival solutions existing on the market, in view to expand or replace our existing environment in 2001.