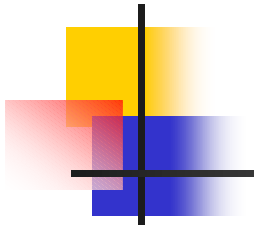
Disk Farm

Distributed farm disk storage

Updated: October 4, 2001

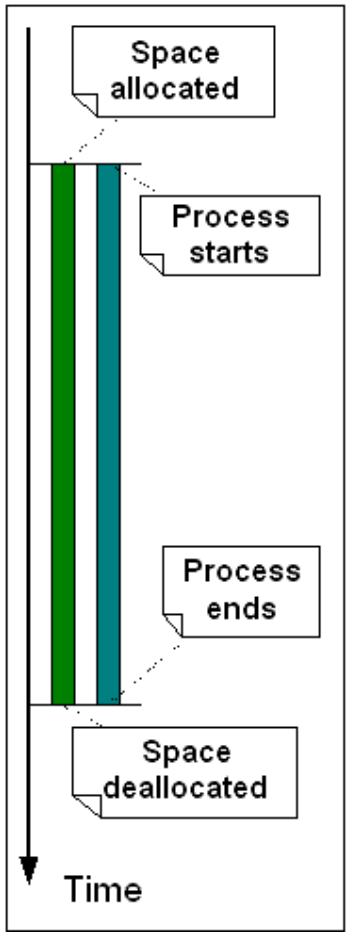


Farm resources

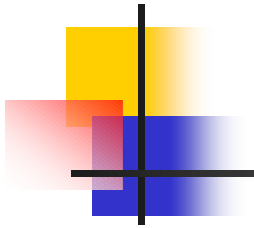
- On I/O node
  - Some CPU
  - Big disk
  - Tape drives
  - Network to worker nodes
  - Outside network
- On “worker” nodes
  - CPU
  - Local disk
  - Network to I/O node

- Used ?
  - Yes
  - Yes
  - Yes
  - Yes, easy to overload
  - Yes
  - Yes
  - Fraction, only as scratch
  - Yes

Scratch disk

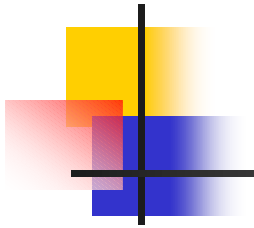


- Resource management philosophy on batch farms:
  - Resources are given to a batch process temporarily, for as long as the process runs
  - After the process finishes, resources are given to the next batch process
  - No resources are allocated "permanently"
  - Local disk is one of temporary resources -> no data can be permanently stored on worker nodes
- Batch process:
  - Allocates resources (including local disk) at start
  - Downloads input data, stores it on local disk
  - Produces output, stores it on local disk
  - Must ***push*** data out before exit
  - Releases all resources
- Data lifetime tied to process life time



Why non-scratch use of "local" disk is difficult ?

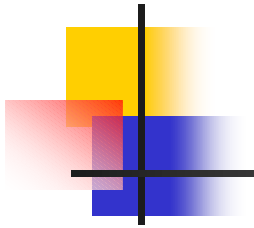
- The resource is scattered in "small" pieces over
  ~100 nodes * 2-4 partitions
- Data "address" would consist of:
  - node, physical path on node
  - group/project, logical path
- Worker nodes are "unreliable", "expendable"
- Hard to coordinate usage by different users, groups, projects
- Local disk space is:
  - Unorganized
  - Unreliable
  - Unmanaged
  - Unused
- Exception: scratch disk



Non-scratch data

- **Final data**
- Destined to
  - Analysis cluster
  - MSS for permanent archival
  - Off-site
- Has to go through I/O node
  - Fast network
  - Connection to MSS
  - Semi-permanent disk storage
- Indefinitely long life time
- Hard to reproduce
- *Can be parked on worker nodes in some cases*

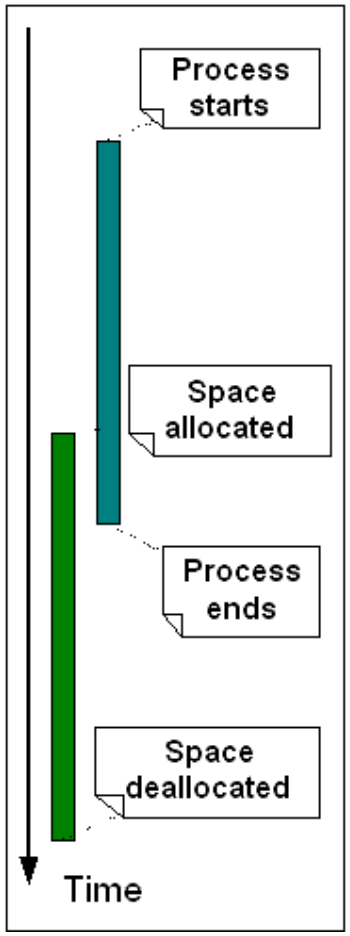
- **Intermediate data**
- Never leaves the farm
  - Input data
  - Concatenation, filtering, packing…
  - MC data to be processed by reconstruction code
- Consumed by a worker node
- Has limited life time
- Often relatively easy to reproduce
- *Perfect for storing on worker nodes*



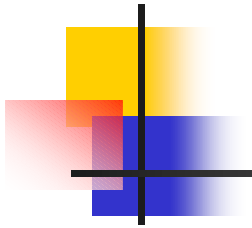
How Disk Farm helps ?

- Name space is organized into "virtual file name space"
  - Virtual path: /E123/data/file.5 – this is what user knows
  - Physical path: fnpc221:/local/stage2/XYZ123 – this is what disk farm knows so that user does not have to
- User operates in familiar UNIX-like file name space using familiar commands
- Solution for node unreliability problem: replicate data
  - Make 2,3,4… copies of the file on different nodes
    - Data is easy to reproduce or has short life: 1 copy
    - Data is precious: 2,5,10… copies
  - Disk Farm replicates data in off-line
- Still, data is not backed-up, not guaranteed
  - Replication extends data lifetime, but does not make it infinite.

Benefits of using Disk Farm

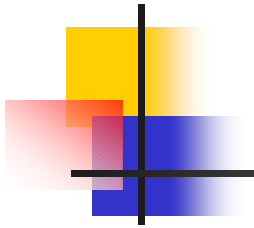


- Disk becomes organized -> manageable -> usable resource
- Extra ~1TB of disk space can be used on FNSFO farm, ~3TB on "big" farms. Will grow with GB/$.
- Shifts load from I/O node to worker nodes:
  - From star topology to point-to-point topology
  - FNSFO: 250 GB/CPU
  - Worker node: 12 GB/CPU
- Improves workers' CPU utilization
  - In most cases, data upload is local
    - Fast
    - Cheap
- Decouples disk allocation/deallocation from process start/end
- Allows data "pull" along with "push" solutions
  - Process running on I/O node can decide when to pull data out



User interface

- Basic UNIX file system commands operate in virtual file name space
  - `dfarm` **`ls`** `<path>|<wildcard>`
  - `dfarm` **`mkdir`** `<vpath>`
  - `dfarm` **`rmdir`** `<vpath>`
  - `dfarm` **`put`** `[-v] [-t <timeout>] [-n <ncopies>] <local path> <vpath>`
  - `dfarm` **`get`** `[-v] [-t <timeout>] <vpath> <local path>`
  - `dfarm` **`rm`** `<vpath>|<wildcard>`
  - `dfarm` **`ln`** `<vpath> <local path>`
- Additional commands
  - `dfarm` **`info`** `<vpath>`
    - Prints where the file is stored
  - `dfarm` **`ping`**
    - Prints list of available disk farm nodes and their load (response time, transactions)
  - `dfarm` **`stat`** `<node>`
    - Prints status of individual farm node (disk space availability)



FNSFO installation

- Disk Farm runs on 35 nodes (excluding KTeV nodes)
- Each node has 2 logical storage areas 7 GB each on /local/stage1,2
- /local/stage3 is scratch partition, managed by FBSNG
- /local/stage4 is reserved
- Total space:
  - 35 x 2 x 7GB = 0.45TB
  - Can be expanded to 35 x 3 x 8GB = 0.84TB
  - To 50 x 3 x 8GB = 1.2TB
- Every user is given 50GB quota to start with