Files Common Across Jobs Part II

The Crimes of Bandwidth

or

The Two Transfers

Todd L Miller

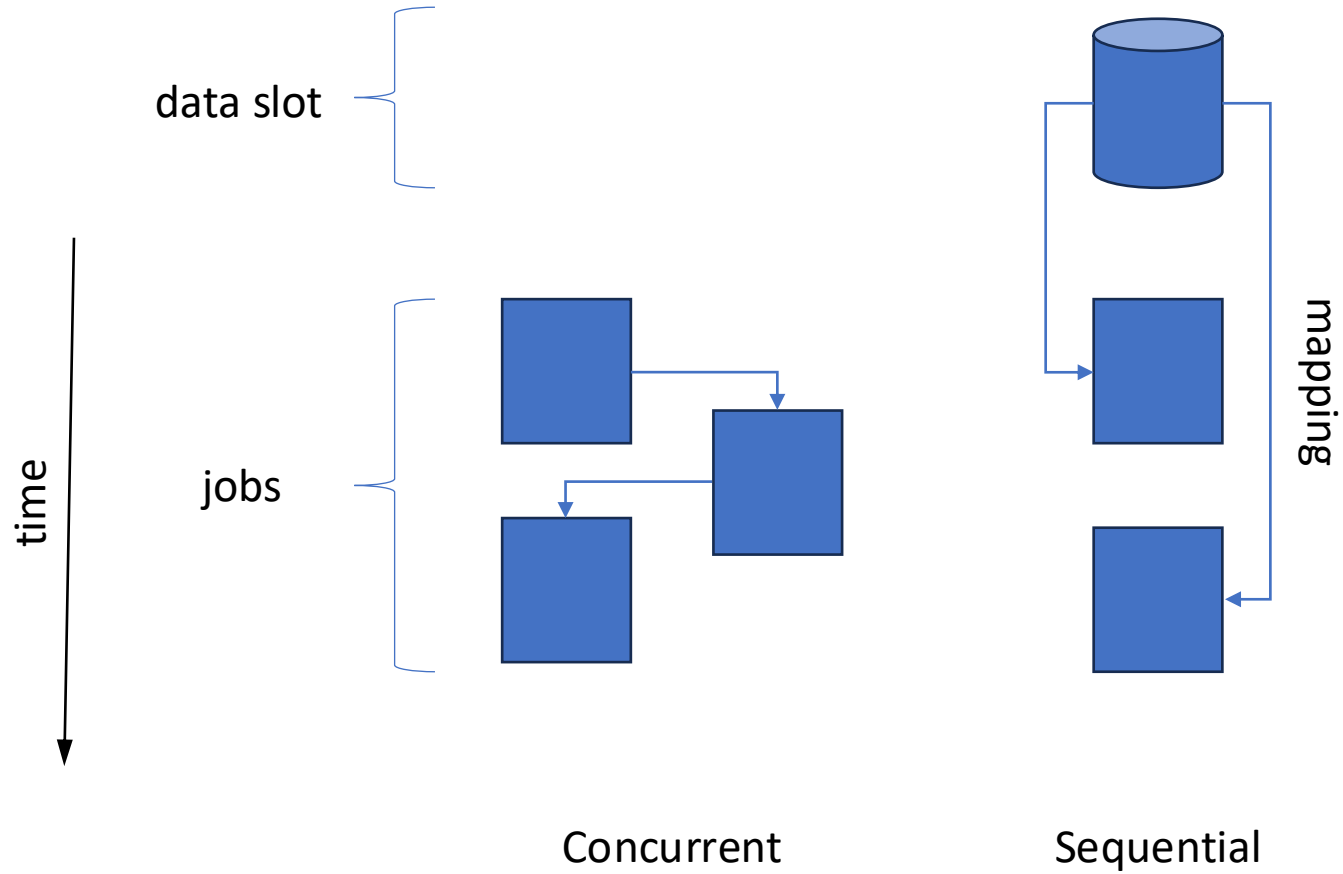
HTC26

CHTC HTCondor Software Suite PATh

Start at the Beginning

- Jobs need files to run.
- Many of these files are the same from run to run.
- Many jobs are submitted as part of a group (either a job list or a DAG).
- Jobs from the same group often run:
  - on the same machine at the same time; (HTC25)
  - or on the same machine one right after the other. (HTC26)
- HTCondor should not – and now does not – need to transfer every such file every time.

Transfers Great / Less Filling



HTCondor no longer needs to transfer every file every time

- **Bonus** If you are not using LVs (that is, have not enabled `STARTD_ENFORCE_DISK_LIMITS`), HTCondor will also not need to make a copy of every file for every job.
  - … if all of your execute directories are on the same filesystem.
- This means that you can not just fit more data-heavy jobs onto an EP, but that it will be much more efficient to run them there.

What’s Coming in 26.0.1

(other than sequential re-use)

CHTC HTCondor Software Suite PATh

Common Container Images

- Jobs can specify that their *container image* is common:
  `container_is_common = True`
- The administrator can say that's true by default:
  `CONTAINER_IMAGES_COMMON_BY_DEFAULT = True`
  - This should be 100% transparent to your submitters.*
  - You may notice a slot called `data1_2@ep.domain`, though.
- Non-container common files can (still) be specified explicitly.

Limitations

Common files …

- … must be immutable (read-only).
- … (except container images) must be explicitly specified.
- … only work in vanilla and container universes.
- … only work in partitionable/dynamic slots.
- … do not work on Windows.
- … can not be re-used between submitters any more than any other form of data transfer shares you data with random strangers.

- Catalogs can not vary within a namespace.

Catalogs & Namespaces

- Common File Catalog
  - A named list of common files and the unit of re-use.
  - Examples (job ad):
    - `_x_condor_container_catalog = “busybox.sif”`
    - `CommonInputFiles = “database.sqlite3, nsf-acknowledgement.txt”`
- Catalog Namespace
  - A scope for catalog names and a value within that scope.
  - Examples:
    - ClusterID == 170000
    - DAGManJobID == 180000

DAGMan & Common Files

- Each job in a DAG is in the same group, so when a job in that DAG refers to a list of common files by name, each job is referring to the *same* list, and we don't have to transfer it again.
- Back-end mechanism only.

The Mean Time to Failure Is Now

- Common file transfer is always optional.
  - The AP will fall back to normal file transfer if given an old EP.
- If a common files transfer fails:
  - all jobs waiting for that transfer will stop waiting and return to idle, and
  - all future jobs using that catalog will fall back on normal file transfer.
- If a common file mapping fails:
  - the job goes back to idle, and
  - the second time a mapping fails, all future jobs will fall back.
- This is deliberately fragile, in part because reporting is difficult.
- If a data slot vanishes, all current jobs will continue unaffected.

Summary of Visible Changes

- Common-transfer events in job event log.
- The B-for-blocked state. Indicates that a job with assigned resources is waiting for a common file transfer to finish.
- “Data” slots. An EP’s resources have previously always been either available (in a p-slot) or assigned to a job (in a d-slot). In order to avoid upsetting that expectation, disk resources used to store common files are represented as “data” slots.
- Reduced bandwidth usage (probably).

What’s Coming in 26.x?

CHTC HTCondor Software Suite PATh

Further Efficiency Gains

- A no-copy mode (using bind mounts) for LV-enabled EPs.
  - As disk -space and -bandwidth efficient as hardlinks.
  - Kernel does not guarantee availability.
- Better scheduling?
  - Steering of jobs to staged common files? (e.g. RANK)
  - Prefer consolidation of slots on an EP to one AP?
- Object-retention policies? Presently fixed-duration lease.
- Common transfer + copies = mutable “common” files?
  - Trading disk space and bandwidth for network bandwidth might not be better in every situation.

Usability Improvements?

- “user” scope for catalogs?
  - Re-use files from any job for any other job with the same submitter.
  - Maybe only those specified ahead of time?
- “project” scope for catalogs?
  - Allow a researcher to set experiment number 7’s runtime and designate grad students to help shepherd the corresponding workflows.
  - Avoids need to manually synchronize updates to the runtime.
- Multiple catalogs per job?
  - Maybe easiest way to support subdags, splices, and categories.
  - Probably the best way to support composability (for transforms).

Questions? Comments?

Thank you for your time.

include : nsf-acknowledgement.txt

This work is supported by NSF under Cooperative Agreement OAC-2030508 as part of the PATh Project. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.