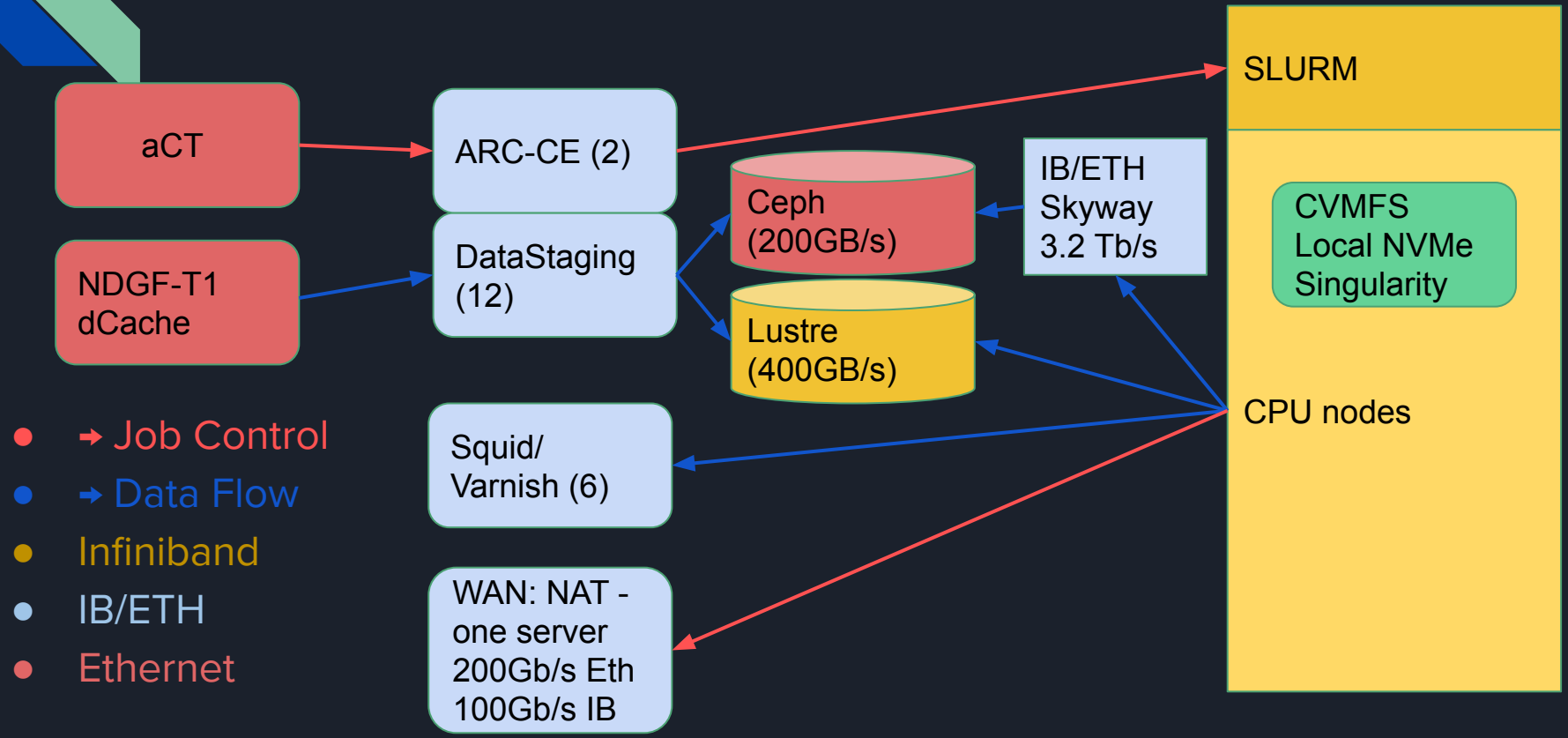
Developments for distributed computing

Gašper Jug

HPC grid overview



What is aCT (ARC control tower)?

- Grid middleware used for submitting jobs to different HPC systems running ARC
- Users submit job descriptions (executable, input files, required resources)
- aCT tracks job states and downloads outputs on their behalf
- Continued development after 3 years
- Used by F9 ATLAS researchers

aCT-Client CLI

Standard use

```
(aCT-venv) gasperj@f9pc19 ~/Desktop/bunchofjobs $ act sub Jobs/job1.xrsl Jobs/job2.xrsl Jobs/job3.xrsl
Submitting batch of 100 jobs ...
Inserted job job1 with ID 2356542
Inserted job job2 with ID 2356543
Inserted job job3 with ID 2356544
(aCT-venv) gasperj@f9pc19 ~/Desktop/bunchofjobs $ act stat
id      jobname JobID                                                                    State    arcstate
-------------------------------------------------------------------------------------------------------
2356542 job1    https://arc02.vega.izum.si:443/arex/rest/1.1/jobs/3fb345370a81 Accepted submitted
2356543 job2    https://arc02.vega.izum.si:443/arex/rest/1.1/jobs/3fb36de2fb37 Accepted submitted
2356544 job3    https://arc02.vega.izum.si:443/arex/rest/1.1/jobs/3fb3901362cb Accepted submitted
```

```
(aCT-venv) gasperj@f9pc19 ~/Desktop/bunchofjobs $ act stat
id      jobname JobID                                                                    State    arcstate
-------------------------------------------------------------------------------------------------------
2356542 job1    https://arc02.vega.izum.si:443/arex/rest/1.1/jobs/3fb345370a81 Finished done
2356543 job2    https://arc02.vega.izum.si:443/arex/rest/1.1/jobs/3fb36de2fb37 Finished done
2356544 job3    https://arc02.vega.izum.si:443/arex/rest/1.1/jobs/3fb3901362cb Finished done
```

```
(aCT-venv) gasperj@f9pc19 ~/Desktop/bunchofjobs $ act get
Results for job job1 stored in 3fb345370a81
Results for job job2 stored in 3fb36de2fb37
Results for job job3 stored in 3fb3901362cb
```

New feature

```
(test-venv) gasperj@f9pc19 ~/Desktop/bunchofjobs $ act sum
cn          cluster                          Preparing Queuing Running finished done failed tocancel cancelled
-------------------------------------------------------------------------------------------------------------------
A.Tomsic    https://arc01.vega.izum.si:443/cpu       /       /       /        / 2152      2        /         1
A.Tomsic    https://arc02.vega.izum.si:443/cpu       /       /       /        / 1810      /        /         /
A.Tomsic    https://hpc.arnes.si:443/all             /       /       /        /    /    548      236         /
G.Jug       https://arc02.vega.izum.si:443/cpu       /       /       /        /    1      /        /         2
J.Debevc    https://arc01.vega.izum.si:443/cpu       /      72       /        /    /      /        /         /
J.Debevc    https://arc02.vega.izum.si:443/cpu       /       8       /        /    /      /        /         /
J.Debevc    https://nsc.ijs.si:443/gridlong          /       /       /        /   11      /        /         /
J.Debevc    https://rebula.ijs.si:443/batch          /       /       /        /  100      /        /         /
J.Gavranovic https://arc01.vega.izum.si:443/cpu      /      14       /        /    /      /        /         /
J.Gavranovic https://arc02.vega.izum.si:443/cpu      /      42       /        /    /      /        /         /
L.Calic     https://arc01.vega.izum.si:443/cpu       /    2969       /        /    /      /        /         /
L.Calic     https://arc02.vega.izum.si:443/cpu       /    2946       /        /    /      /        /         /
L.Calic     https://pikolit.ijs.si:443/batch       660       /       8        1    /     17        /         /
L.Calic     https://rebula.ijs.si:443/batch        446       /       /        /    /     17        /         /
```

- Package deployed to pypi for easier installation (pip install act-client)

ACT improvements

Problems:

- Frequent DB deadlocks - require manual restart by admin
- Long lived connections - become stale - require restart
- Users report slow API and problems with output download
- Inconvenient deployment and session management

Implemented fixes:

- Rewrote raw SQL DB calls with SQLAlchemy library - gained mariadb, postgreSQL, SQLite portability and built in connection refresh; more readable DB schema
- Replaced locks with atomic operations and short lived connections
- Increased service API worker count, reworked downloads
- aCT now runs as system service, can also run containerized
- Added job submission throttling to prevent overloading ARC

What is nordugrid ARC (advanced resource connector)?

- ‘The Advanced Resource Connector (ARC) middleware, developed by the NorduGrid Collaboration, is an open source software solution enabling e-Science computing infrastructures with emphasis on processing of large data volumes’
- Downloads inputs to local storage (don’t waste allocated processor time on internet downloads)
- Submits jobs to LRMS (local resource management systems - responsible for job queue management, resource allocation and efficient job execution) and tracks lifespan
- Lives on dedicated nodes on HPC systems

Main features:

- Support for various LRMS - SLURM, HTCondor… Same interface, different backend
- Support for various internet download protocols Http(s)/WebDAV, xRootd, Rucio…
- Shared input cache - avoid downloading same input for multiple jobs
- Publishing prometheus metrics for grafana
- Keeping accounting data for each job

ARC development

- Open source collaborative development
- Independent development, deployment and testing on live CEs on HPC Vega with Andrej Filipčič and Claude code
- 3 merge requests - 1 approved, 2 pending, at least 2 more on the way
- Discussion with maintainers about AI use

Minor fixes:

- Fixed an unintentional long long int -> int conversion for values in the accounting DB - no more negative file sizes
- Restored functionality for the grafana data exporter

Improvement 1: TLS cache

Most user inputs use http(s)/WebDAV transfer protocol

Problem:

- TLS store (~220MB) being rebuilt too often for transfers using http(s)/WebDAV, costing 50s per file batch

Solution:

- Cache the TLS store and rebuild once every 5min-1h

This results in a test job (1000 inputs of 1MB) download time going down from 15min to 1min

Improvement 2: Remote transfer workers

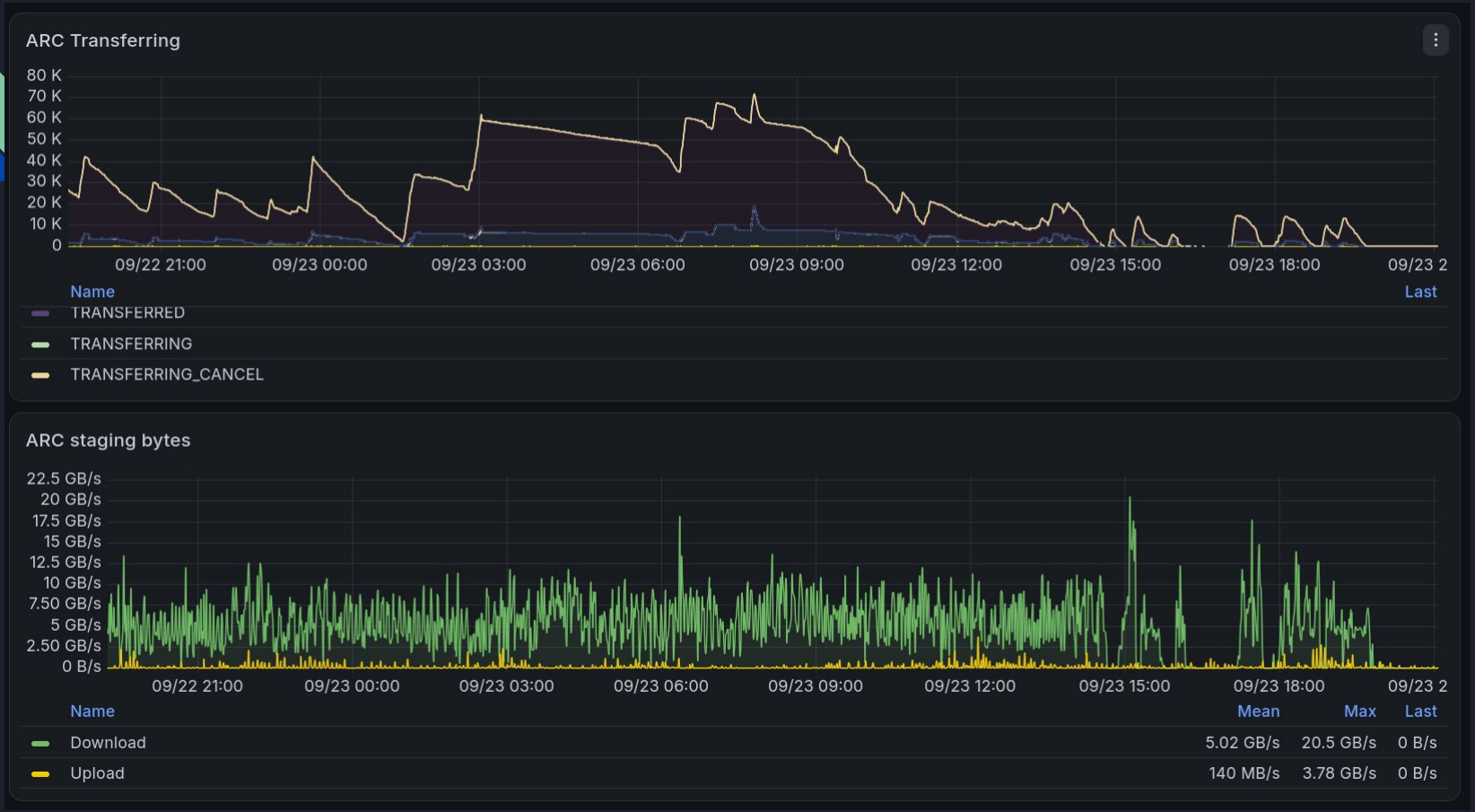
Offloading transfer work from the machine running ARC to several transfer hosts reduces CPU pressure on the machine and potentially increases transfer rates

Problem:

- The current implementation is unusable for smaller files - the transfer status of all files is polled individually/serially by the main host - the files finish downloading way before they are polled and polling cycles are too slow

Solution:

- Implemented bulk polling once for all jobs on each transfer host



Can easily handle 50k+ input load in a couple hours

Improvement 3: Fixed file cache lock behavior

Only one job at a time can process a download. ARC ensures this by keeping file locks for each input source with a hardcoded timeout of 15 min. If the download takes more than that another job takes over the lock while the first one might finish successfully.

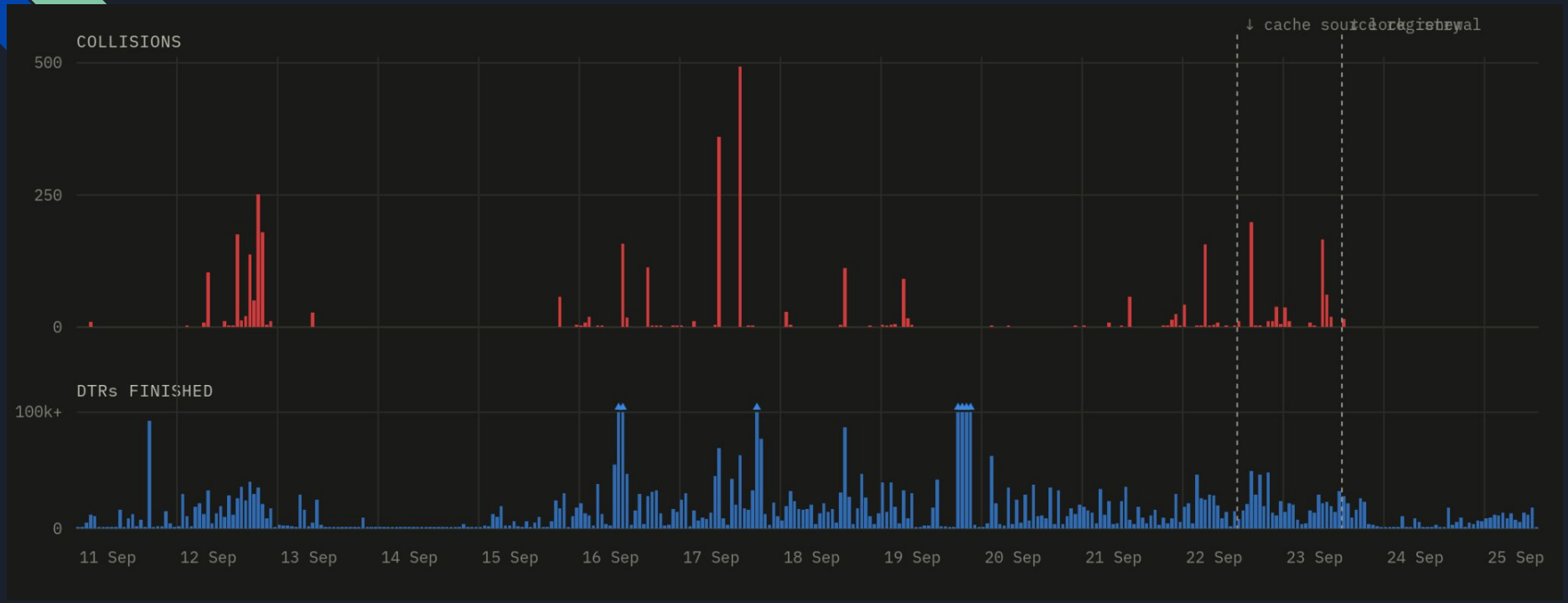
Problem:

- This causes the second job to fail

Solution:

- Refresh an active lock so it can’t be stolen

Significant job failure reduction



Hourly failures (top) vs Hourly traffic estimate (bottom)