Provenance-aware optimization of workload for distributed data production

Dzmitry Makatun [1, 3] Jérôme Lauret[2] Hana Rudová [4] Michal Šumbera [1]

[1]Nuclear Physics Institute, The Czech Academy of Sciences

[2]Brookhaven National Laboratory, USA

[3]Czech Technical University in Prague, Czech Republic

[4]Faculty of Informatics, Masaryk University, Czech Republic

makatun@rcf.rhic.bnl.gov

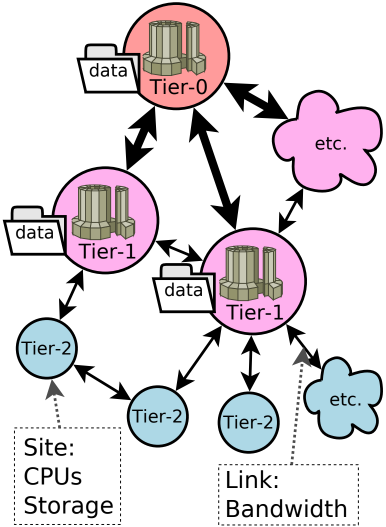
Planning of data production at remote sites

- A given dataset to be processed once
- Computational resources {`CPUs, storage`} are available at multiple geographically distributed sites (Tier-0/1/2)
- Some sites have (partial) data replicas, some not
- Realistic network: shared links, alternative transfer paths

? **How much data should be processed at each site? How and when to transfer it? Which data-source to use?**

✘ General scheduling approaches: either focused on a single aspect or do not scale well

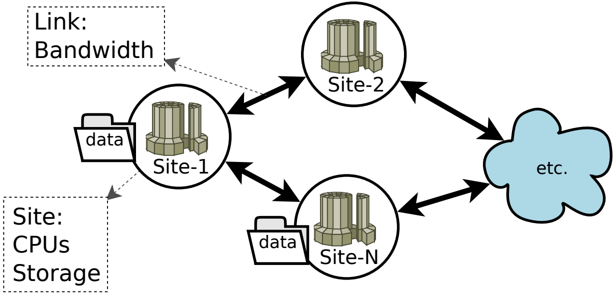
✘ Custom setups: difficult to re-adjust for changing infrastructure (addition/withdrawal of sites, cloud resources)



Our scheduling approach

Idea: since production jobs are "predictable":

- Plan resource load and then distribute data accordingly
- Plan for a limited time ΔT (e.g. 12 hours) for adaptive feedback, repeat cyclically



Planner input

data location, state of resources, network structure and load

Planner output

data flows over each link

✓ **Network flow maximization** approach → polynomial complexity (good)

How is the plan executed?

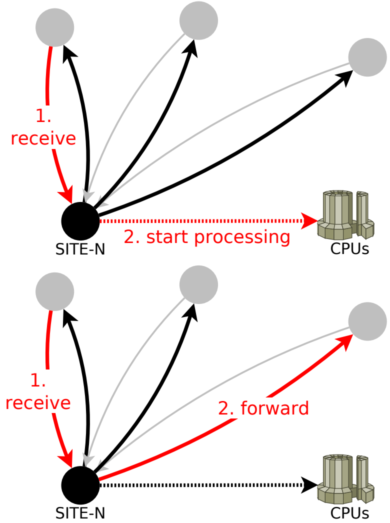
Independent handlers act to comply with planned **data flows**

Handler

Service running at each site
When a new file arrives:

Process the file
OR Forward it to a neighboring site
OR Store it for future use

Handler takes data replication into account



How do we test our approach?

Common tools for simulations of distributed computations (GridSim) [Buyya and Murshed, 2002]

Our previous simulations

- ✓ Basic setups
- ✓ Background traffic over shared network links
- ✓ Tier-1s network of one of the largest HENP experiments (40k CPUs)
- ✓ Random large-scale networks (50 sites)

Recent simulations

- ★ Multiple sources of input data
- ★ Data replication

Input for simulations

Scheduling policies

1. PLANNER
2. PULL: each resource access input data from the closest (by ping) source

Initial data location

- Each file has a copy at Tier-0 and one of Tier-1s. Each Tier-1 has equal amount of data.
- Output is sent to Tier-0

Job parameters

Log records of data production for STAR at KISTI (June – September 2014) [Hajdu et al., 2015]

Parameters of sites and network

Online monitoring tools of CERN experiments [REBUS, 2015] [MonAlisa, 2015], [LHCOPN, 2015]

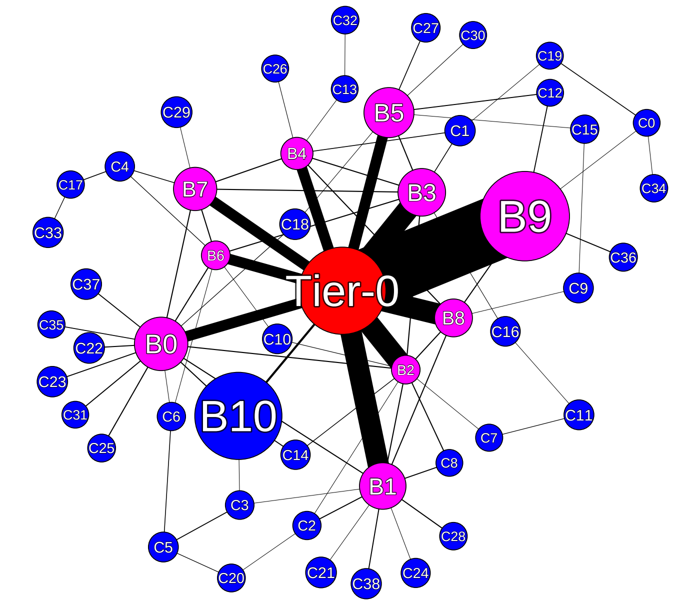
Simulated network

- Tier-0/1 network of the largest HENP experiments (downscaled)

  + dummy Tier-1 site with poor connection to Tier-0 (B10)

  + random scale-free network of Tier-2s

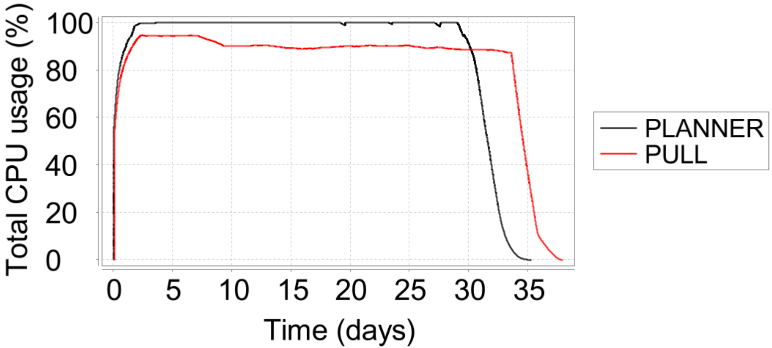
- 51 sites, 36k CPUs, 600 k files, 2,7 PB



Legend

Tier-0 (source/destination only)
Tier-1 (source + processing) ∼1k CPUs
Tier-2 (processing only) ∼100 CPUs
node size ∼ #CPUs
edge thickness ∼ bandwidth

Simulation results: total CPU usage



- PLANNER provides higher CPU utilization
- Overall makespan improvement is 7 %

- Better utilization of sites with poor connectivity

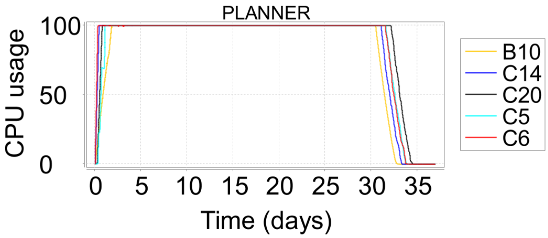
∼10 ms to create a plan for 12 hours of data production

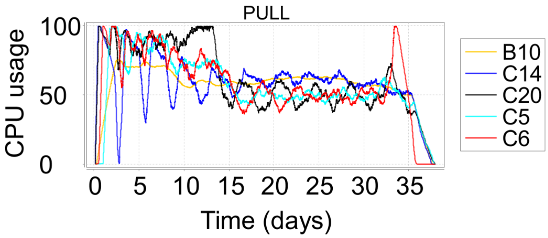
Simulation results: CPU usage per site

CPU usage at 5 “worst” sites

PLANNER reaches 100 % CPU usage at all sites due to the better utilization of low bandwidth:

- Data flow is distributed between alternative transfer path
- Avoid over-commit of network bandwidth (congestion)
- Data are transferred to computing site before the job starts





Conclusion

Previous work

- New job scheduling approach for data production - global optimization of resource usage {CPU, disk, network bandwidth}
- Adaptive, can deal with loaded (shared) networks and self-discover alternative network path
- Demonstrated our model systematically provides better makespan than common approaches (PULL, PUSH, ...)

Recent results

- Extended to reason on multiple input sources and data replication
- Simulations in a realistic large-scale heterogeneous infrastructure added few "problematic sites" (non-otpimized) to challenge the algorithm
- **Our approach has consistently showed significant improvements over standard ones and can make best use of sites with limited connectivity**

The end

References

Buyya, R. and Murshed, M. (2002).
GridSim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing.
The Journal of Concurrency and Computation: Practice and Experience (CCPE), 14.

Hajdu, L., Lauret, J., Didenko, L., Amol, J., Betts, W., Jang, H. J., and Noh, S. Y. (2015).
STAR experience with automated high efficiency Grid based data production framework at KISTI/Korea.
In *HEPiX Spring 2015 Workshop*. Oxford University, UK.

LHCOPN (2015).
LHC Optical Private Network.
http://lhcopn.web.cern.ch/lhcopn/.

Makatun, D., Lauret, J., Rudová, H., and Šumbera, M. (2015).
Model for planning of distributed data production.
In *Proceedings of the 7th Multidisciplinary International Scheduling Conference (MISTA)*, pages 699–703.

Makatun, D., Lauret, J., Rudová, H., and Šumbera, M. (2016a).
Multi-resource planning: Simulations and study of a new scheduling approach for distributed data production in high energy and nuclear physics.
Journal of Physics: Conference Series.
(Accepted for publication).

Makatun, D., Lauret, J., Rudová, H., and Šumbera, M. (2016b).
Network flows for data distribution and computation.
Proceedings of the IEEE Symposium on Computational Intelligence in Scheduling and Network Design.
(Submitted).

MonAlisa (2015).
MonAlisa: Grid online monitoring data of the ALICE experiment.
http://alimonitor.cern.ch/.

REBUS (2015).
WLCG REsource, Balance & USage.
https://rebus.cern.ch/.