Scheduling in Desktop GRID:
a Review

Ilia A. Chernov, Natalia N. Nikitina, Evgeny E. Ivashko

*Institute of Applied Mathematical Research,
Karelian Research Centre, RAS*

*The 7th International Conference "Distributed Computing and Grid-technologies in Science and Education"
4 - 9 July 2016
Dubna, Russia*

Desktop grids

DG = Computational system built of different computers (PC, servers, cluster nodes, smartphones, laptops, etc) connected by common networks. Resources are usually used only when idle.
EDG = DG using computers of one or a few institutions
VC = DG using volunteers' resources, Internet links all together
BOINC = popular middleware for DG

	Enterprise DG	**Volunteer computing**
Size	Small	Large
Reliability	High	Low
Availability of nodes	High	Low
Predictability	Higher	Lower
Malicious actions	Unlikely	High risk
Heterogenity	Lower	Higher
Studied	Less	More

Choi SJ et al. A taxonomy of DG systems focusing on scheduling. 2006.

Anti-sabotage

> **«All Internet users are potentially dangerous except actually dangerous ones» / Larry Wall**

Saboteurs are DG users that intentionally send wrong results.

Reasons can be:

- desire for a reward for free;
- stealing a result;
- malicious behaviour.

Saboteurs can be:

- Independent or cooperartive;
- Intelligent or not.

Anti-sabotage: methods

- Replication, quorum: up to n identical results
- Voting: n tries, maximal result believed in
- Sampling: traps for saboteurs
- Spot-checking, backtracking, blacklisting
- Ringers: one-way functions
- Checkpoints
- Reputation-based systems
- Game-theoretic approach

Anti-sabotage: papers

- *Domingues P., Sousa B., Silva L.M.* Sabotage-tolerance and trust management in desktop grid computing. 2007.
- *Sarmenta L.F.G.* Sabotage-tolerance mechanisms for volunteer computing systems. 2002.
- *Sarmenta L.F.G.* Volunteer computing. PhD thesis. 2001.
- *Yu J., Luo Y., Wang X.* Deceptive detection and security reinforcement in grid computing. 2013.
- *Du W. et al.* Uncheatable grid computing. Electrical Eng. Comp. Sys. Paper 26. 2004.
- *Wang Y. et al.* Toward integrity assurance of outsourced computing — a game theoretic perspective. 2016.
- *Szajda D., Lawson B., Owen J.* Toward an Optimal Redundancy Strategy for Distributed Computations. 2005.
- *Christoforou E. et al.* Reputation-based mechanisms for evolutionary master-worker computing. 2013.
- *Christoforou E. et al.* Internet computing: using reputation to select workers from a pool. ArXiv 2016.
- *Goodrich M.T.* Pipelined algorithms to detect cheating in long-term grid computations. 2008.
- *Canon L.C., Jeannot E., Weissman J.* A scheduling and certification algorithm for defeating collusion in DG. 2011.
- *Wander M. et al.* Detecting opportunistic cheaters in VC. 2011.
- *Kuhn M.* Distributed asymmetric verification in computational grids. 2008.

Reliability

- Reliability is tolerance to faults, errors, etc.
- No-error OR have result.
- Not necessary sabotage!
- If sabotage, then non-cooperative and unintelligent.
- Hardware, software faults; algorithmic malfunction; bad initial guess; divergence, more.

Reliability: methods

- Job replication = copying tasks
- Component replication = lib, software, etc.
- Data replication = same chunks of data
- Checkpointing = save not to loose
- Scheduling = time quanta OR space distribution
- Heartbeat = «Are you alive?»
- *Sathya S.S. Babu K.S.* Survey of fault tolerance techniques for grid. 2010.

Availability: is a node there?

- Pattern revealing
- Forecasting availability in general
- Reputation wrt availability
- Grouping and clustering
- *Durrani M.N., Jawwad A.S.* VC: requirements, challenges, and solutions. 2014.
- *Rubab S. et al.* A review on resource availability prediction... 2014.

Dependent tasks

- Dependencies described by an acyclic graph
- More problems with reliability, availability
- *Lee Y.C. et al.* Robust task scheduling for VC systems. 2010.
- *Cordasco G. et al.* On scheduling DAGs for volatile computing platforms... 2012.
- *Gao L., Malewicz G.* Internet computing of tasks with dependencies... 2004.

Heuristics for scheduling

- Makespan: t(completion)-t(submission)
- Throughput: N(jobs completed)
- Total time
- Max load
- Reliability: low risk of wrong result
- Low risk of not getting the result
- Deadlines

Estrada T., Taufer M. Challenges in designing scheduling policies in VC. 2012.

Genetic scheduling

- *Estrada T. et al.* A distributed evolutionary method to design scheduling policies for VC. 2008 — grammar, evolution on sets of rules (!)
- *Qu B. et al.* A new genetic algorithm based scheduling for VC. 2010. - makespan+reliability
- *Wang et al.* Optimizing the makespan and reliability for workflow applications... 2011. - the same))
- *Alieksieiev M. et al.* Repeating tasks scheduling in DG systems. 2014. (rus). - balancing load.

Multiple projects

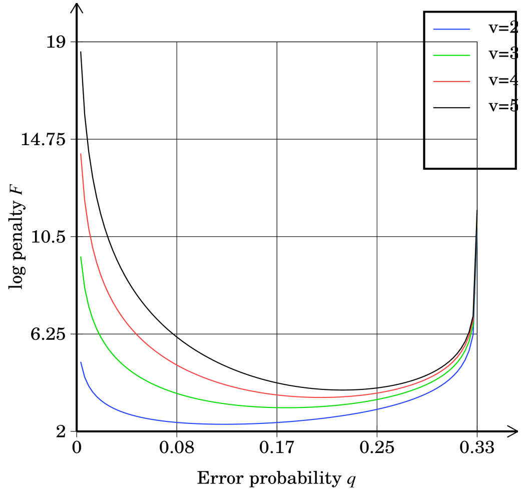
- *Atlas J. et al.* Balancing scientist needs and volunteer preferences... 2009.
- *Donassolo B. et al.* Non-cooperative scheduling considered harmful... 2011. - game.
- *Kondo D. et al.* Performance evaluation of scheduling policies for VC. 2007.
- *Anglano C. Canonico M.* Scheduling algorithms for multiple bag-of-tasks applications on DG: a knowledge-free approach
- *Bochenina K. et al.* Static scheduling of multiple workflows with soft deadlines... 2016.

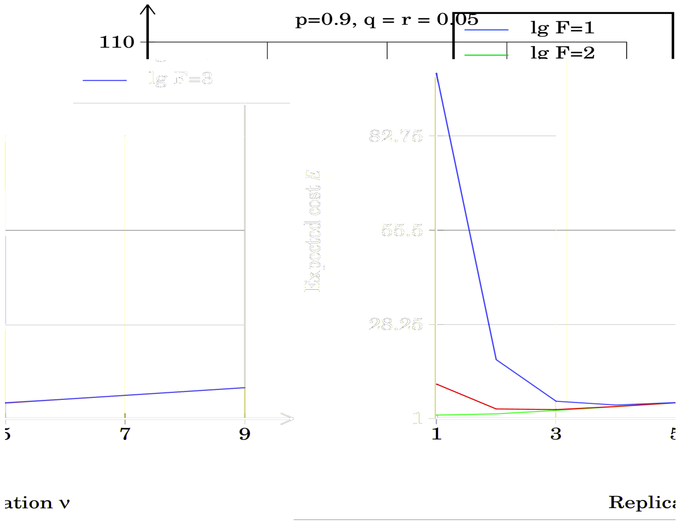
Batches of tasks

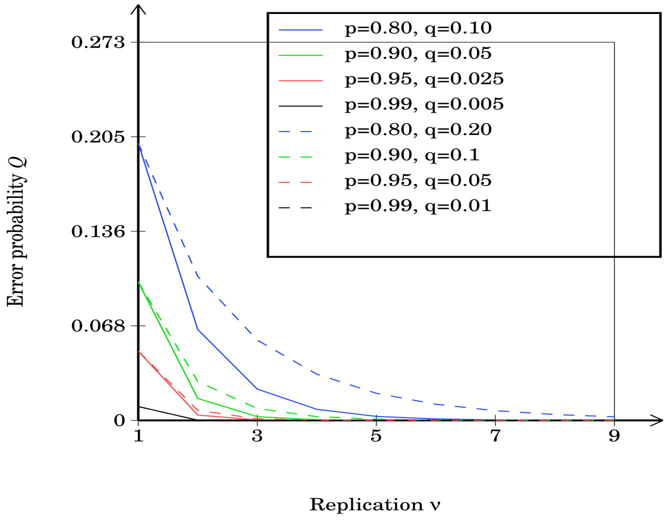
- *Maheswaran M. et al.* Dynamic mapping of a class of independent tasks … 1999 - compares 5 immediate and 3 batch scheduling policies.
- *Szajda D. et al.* Hardening functions for large-scale distributed computations. 2003. - antisabotage by inserting checks to a batch
- *Yu J., Luo Y., Wang X.* Deceptive detection and security reinforcement in grid computing. 2013. - antisabotage, checks in batches.
- *Du W. et al.* Uncheatable grid computing. 2004. - antisabotage, checks in batches, hash.
- *Heien E.M. et al.* Computing low latency batches with unreliable workers in VC environments. 2009. - scheduling batches.
- *Toth D., Finkel D.* Improving the productivity of VC by using the most effective task retrieval policies. 2009. - buffering tasks.
- *Mazalov V., Nikitina N., Ivashko E.* Hierarchical two-level game model for task scheduling in a desktop grid. 2014. - Batches to reduce the server load.

Our approach

- Minimizing the total cost of calculation
- TC = work + penalty if wrong
- Replication with quorum reduces the risk
- But increases the work amount
- Quorums, penalties, etc can differ for answers







Our approach: critical penalties

- Penalties usually are not known precisely
- Though, critical penalties exist
- A critical penalty changes an optimal quorum
- They are reciprocal to powers of risk level
- and thus differ much.
- So it suffices to know penalties approximately

Our approach: papers

- *Chernov I., Nikitina N.* Virtual Screening in a Desktop Grid: Replication and the Optimal Quorum, 2015.
- *Chernov I., Nikitina N.* Optimal quorum for a reliable desktop grid, 2015.
- *Chernov I.* Optimal quorum for the model of computational grid with redundancy, 2014.
- *Chernov I.* Optimal duplication of tasks in a computing system (Russian), 2014.
- *Chernov I.* Replication in desktop grid computing for minimizing the mean cost, 201[67] - <u>to be submitted</u>.

That's it

Thank you for your attention!