Desktop Grid Computing in Materials Science Lab - Example of Development and Execution of Application for Defect Aggregation Simulations

*Olexander Gatsenko, Olexandra Baskova, and Yuri Gordienko**

G.V. Kurdyumov Institute of Metal Physics of the National Academy of Sciences,
36 Vernadsky Street, Kiev 03142 Ukraine *E-mail: gord@imp.kiev.ua

Introduction: The processes of monomer aggregation in clusters are investigated in many branches of science: defect aggregation in materials science, population dynamics in biology, city growth and evolution in sociology, etc [1-3]. The typical simulation of a cluster aggregation scenario takes several days on a single modern processor, depending on the number of Monte Carlo steps (MCS). The parallel execution environment can reduce the waiting time and scale up the simulated system to the desirable realistic values.

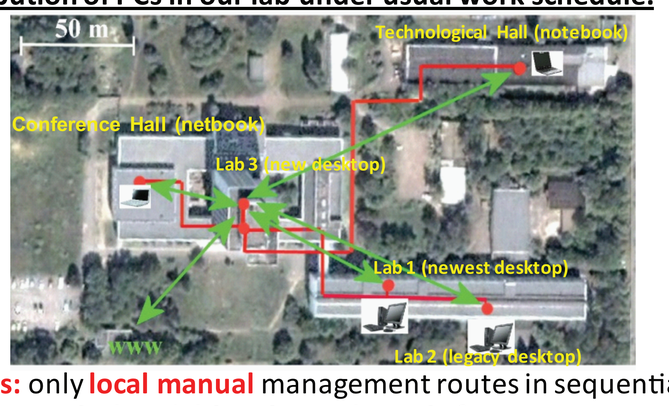
The main aim of the work was to test the applicability and everyday efficiency of the Desktop Grid computing technology in an ordinary materials science lab with heterogeneous distribution of desktop computing resources. For this purpose, the ported parallel version of the sequential application was developed and tested on the worker nodes of Desktop Grid (DG) installed on BOINC software platform [4] and Distributed Computing Application Programming Interface (DC-API) [5].

1. Problem Description

Typical heterogeneous distributed computing infrastructure

Category	Floating point MIPS (Whetstone)	Integer point MIPS (Dhrystone)
Newest (4-core CPU)	2317	4914
New (2-core CPU)	3072	5388
Notebook (2-core CPU)	1924	4063
Netbook (2-core CPU)	674	1738
Legacy	878	1383

Distribution of PCs in our lab under usual work schedule.



Red lines: only **local manual** management routes in sequential run

Main hardships in such an infrastructure

are caused by human factor -> some users…

- **do not like to provide remote access** to their PCs and housekeeping operations can be done manually and on site only;
- **forgot** about background sequential application and **switch-off their PCs after working hours**;
- **do not allow use PCs in the absence of their owners**.

It is very hard to do input/output housekeeping operations, and even plan them reliably.

How to resolve these hardships? (Don't worry, see to the right side...→)

2. Efficient Way to Solution -> Desktop Grid

What do you need to do:

0.Download **free** software:
- Apache Web-server (**freeware**);
- MySQL database (**freeware, GPL**);
- BOINC software (**freeware, LGPL**);
- DC-API libraries for DG by SZTAKI (**freeware**);

1.Install Desktop Grid (DG) with a BOINC DG-server
2.Port your sequential application to parallel version.
3.Deploy an application on the DG-server.
4.All BOINC-clients **themselves** connect to the DG-server.
5.Run your application.
6.Enjoy **automatic** input/output housekeeping.

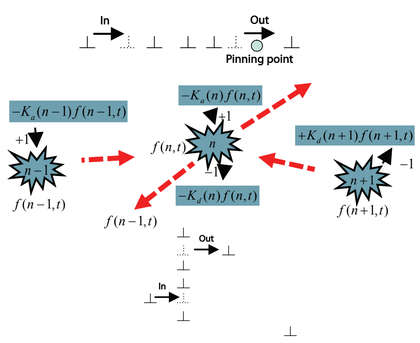
Green arrows: Desktop Grid **automatic** operations (local and **global**) in parallel run

Main advantages:

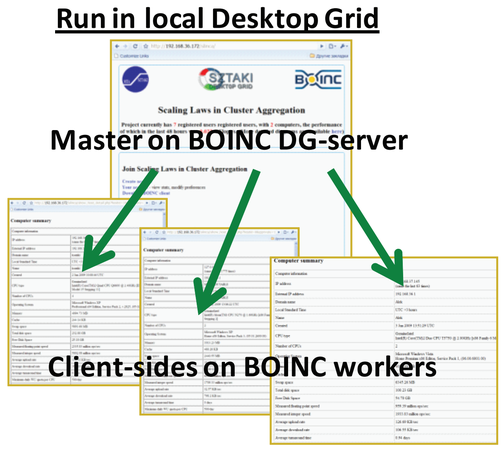
- **performance speedup** of the DG-enabled application **is much higher than our expectations** even in our low-scale local DG testbed (please, see example below),
- **huge economy** on personnel working time due to the **better usability** and **more efficient housekeeping,**
- **unlimited scaling-up** by volunteer PCs,
- **automatic setup** of client workers,
- more **flexible management** of idle/busy client workers.

3. Example of Typical Simulation in Our Material Science Lab with Heterogeneous Distribution of Computing Resources

Create Kinetic Monte Carlo model

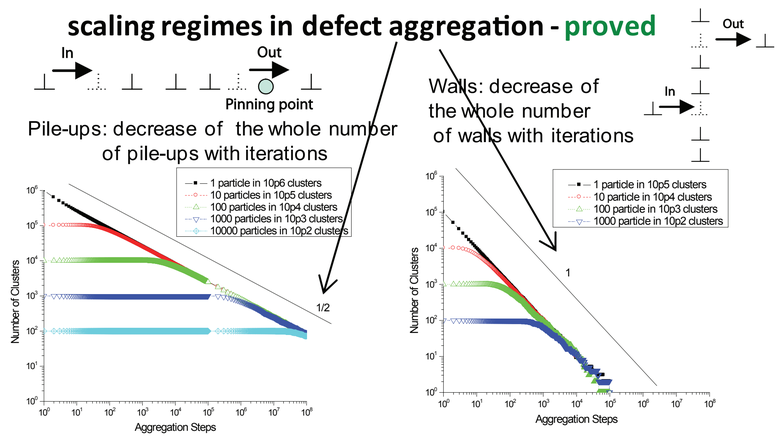


Run in local Desktop Grid



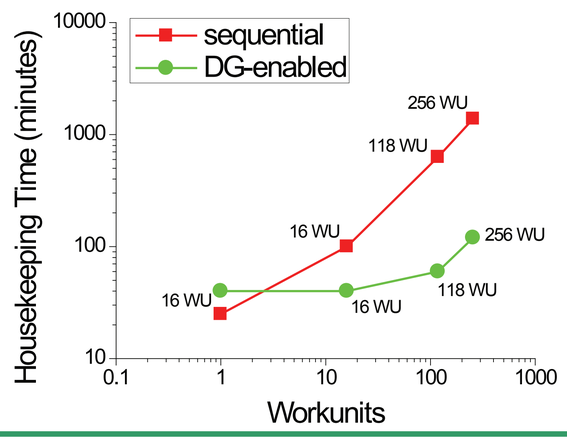
Obtain Scientific Results:

scaling regimes in defect aggregation - proved

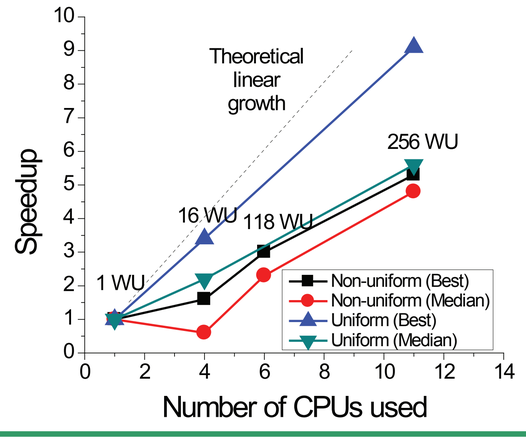


4. Performance Analysis of Desktop Grid Solution

Much lower housekeeping time



Linear speedup



5. Drawbacks

- **blind distribution** of workunits among workers with different power;
- **problems** with **"ad hoc"** dynamical memory allocation;
- **bottlenecks** in **heterogeneous** distributed computing infrastructure created by slow and greedy PCs.

Fortunately, these disadvantages could be resolved by proper configuration of application from BOINC master-side.

6. Acknowledgements

The work presented here was funded by the FP7 EDGeS project. The EDGeS (Enabling Desktop Grids for e-Science) project receives funding from the European Commission within Research Infrastructures initiative of FP7 (grant agreement Number 211727) (www.edges-grid.eu).

[1] I. M. Lifshitz, V.V. Slyozov: The Kinetics of Precipitation from Supersaturated Solid Solution; J. Phys. Chem. Solids, 19, [1/2] , pp. 35-50, 1961.
[2] S. Ispolatov, P.L. Krapivsky, S. Redner: Wealth Distributions in Models of Capital Exchange; Eur. Phys. J., B 2, pp. 267-276, (cond-mat/9708018), 1998.
[3] F. Leyvraz , S. Redner: Scaling Theory for Migration-Driven Aggregate Growth; Phys. Rev. Lett., 88, 068301, (cond-mat/0108289), 2002.
[4] BOINC — Berkeley Open Infrastructure for Network Computing (http://boinc.berkeley.edu/).
[5] SZTAKI Desktop Grid set up by the Computer and Automation Research Institute (SZTAKI) of the Hungarian Academy of Sciences (MTA) (www.desktopgrid.hu/).