Scaling-up MATLAB® Application in Desktop Grid for High-Performance Distributed Computing — Example of Image and Video Processing

Olexandra Baskova, Olexander Gatsenko 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: Recently, the distributed computing model becomes very popular due to feasibility to use donated computing resources of idle PCs by means of the BOINC software platform [1] and availability of simple and intuitive Distributed Computing Application Programming Interface (DC-API) [2]. Usually, a sequential application by slight modifications in its code could be ported to the parallel version for worker nodes of a distributed Desktop Grid (DG). It is possible for an application with independent processing of big volume of data in a client-server model, for example, for batch image and video processing. The main aim of the work was to develop an example and test the applicability of integration of MATLAB® objects and codes in a DG for high-performance distributed computing on the example of image and video processing in solid state physics [3].

1. Typical Experimental Workflow

High-Speed Video Camera
Output is Video File (>10^9 frames, >1 Mpixels)

Modeling of evolution of metal surface

Output: Scientific results (plots, datasheets)

2. Possible Ways to Increase Video Processing Perfromance by Parallel Computing with MATLAB®

- standard commercial approach -> MATLAB Parallel Computing Tools
  - Parallel Computing Toolbox™ (in package);
  - Distributed Computing Server™ (MDCS) (commercial);
  - MATLAB MDCS™ license for the cluster (!)
  
- our original approach -> MATLAB Compiler + DC-API + BOINC DG
  - BOINC software (freeware);
  - DC-API libraries for DG by SZTAKI (freeware);
  - MATLAB Compiler + MATLAB Compiler Runtime (MCR)
  (in package, if you have MATLAB, then you already have them)
  
Typical obligatory procedures:
1. Install a server (so-called “MATLAB Client”):
   - obtain a license file, install MDCS, start the License Manager on a server;
   - test licenses of workers (log in to all workers).
2. Develop your parallel MATLAB application
3. Configuring worker nodes (“MATLAB Workers”):  
   - install the “mdce” services (on all workers);
   - start the Job Manager and manually list all workers.
4. Run application with manual housekeeping.

3. Performance Analysis of Desktop Grid Enabled Solution

4. Drawbacks

- need to install MATLAB Compiler Runtime on workers;
- unpredictable “black box” behavior of wrapped MATLAB functions;
- not all MATLAB functions and toolboxes are portable.

5. Advantages

- NO necessity in MATLAB DCS license, only incorporated MCompiler and MCRuntime,
- unlimited scaling-up by volunteer PCs,
- automatic setup of workers and more flexible management of idle/busy workers.

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).