Porting Multiparametric MATLAB Application for Image and Video Processing to Desktop Grid for High-Performance Distributed Computing

Olexandra Baskova, Olexander Gatsenko*, Gilles Fedak**, Oleg Lodygensky***, and Yuri Gordienko**

*G.V. Kurdyumov Institute of Metal Physics of the National Academy of Sciences, Kiev, Ukraine, gord@imp.kiev.ua
**INRIA, LRI, Université de Paris Sud, Orsay, France, Gilles.Fedak@inria.fr
***LAL, Université de Paris Sud, Orsay, France, lodygens@lal.in2p3.fr

Introduction: The current common tendency in science is to use the scientific software packages with many specialized toolboxes, user-friendly interfaces, where scientists can make research in an interactive manner by means of 4th generation languages (4GL) to reach high efficiency and productivity. Unfortunately, most of such popular software packages do not include functions or services which allow users to take advantage of parallel processing or run as many workers as thee want without huge investments of money and human resources.

The main aims of the work:
- to show that a sequential version of the scientific application coded in 4GL (on the basis of MATLAB-code) could to be easily ported to the parallel version for a distributed computing infrastructure (DCI) as Desktop Grid (DG);
- to test different target desktop grid platforms and test robustness of ported 4GL-based application at production level.

For these purposes we used, as an example, our application *MultiscaleIVideoP* (Multiscale Image and Video Processing), which was coded in 4GL (namely, MATLAB-code) and designed for image and video processing in solid state physics [1,2].

1.Porting the Sequential MATLAB-application to Distributed Computing Infrastructure (DCI)

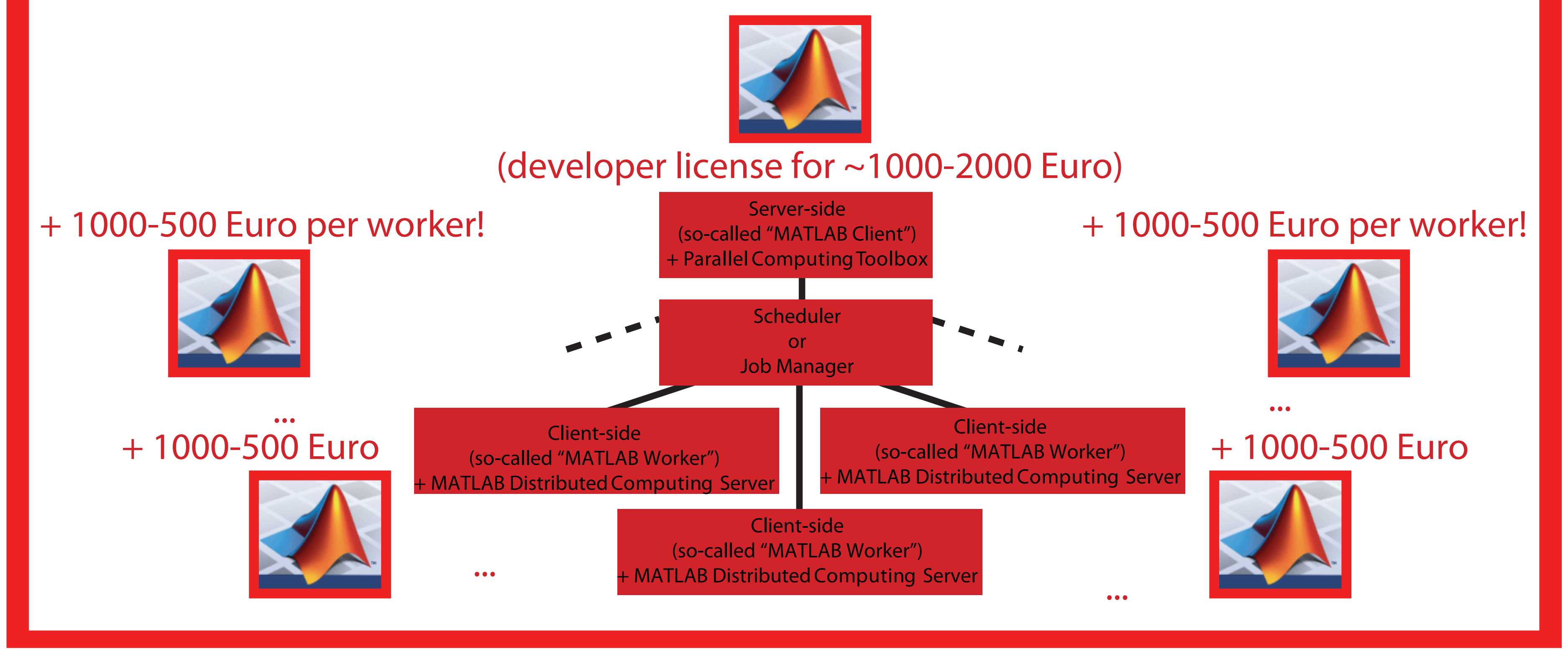
Commercial approach by MATLAB

Minimal pre-requisites (commercial only):
- MATLAB IDE + MATLAB Compiler (developer license for ~1-2 KEuro);
- MATLAB Compiler Runtime (MCR) (in package already);
- MATLAB Parallel Computing Toolbox (~1000 Euro);
- MATLAB Distributed Computing Server (MDCS);

Price of MDCS license depends on the number workers:
8 workers - 7000 Euro,
...
256 workers - 140 000 Euro, ...

Typical obligatory procedures:

1.Port your sequential version of MATLAB application to parallel one.
2.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).
3.Configuring worker nodes ("MATLAB Workers"):
   - install the "mdce" services (on all workers);
   - start the Job Manager andmanually list all workers .
4.Deploy the ported parallel MATLAB-coded application to MDCS server.
5.Run application withmanual housekeeping.

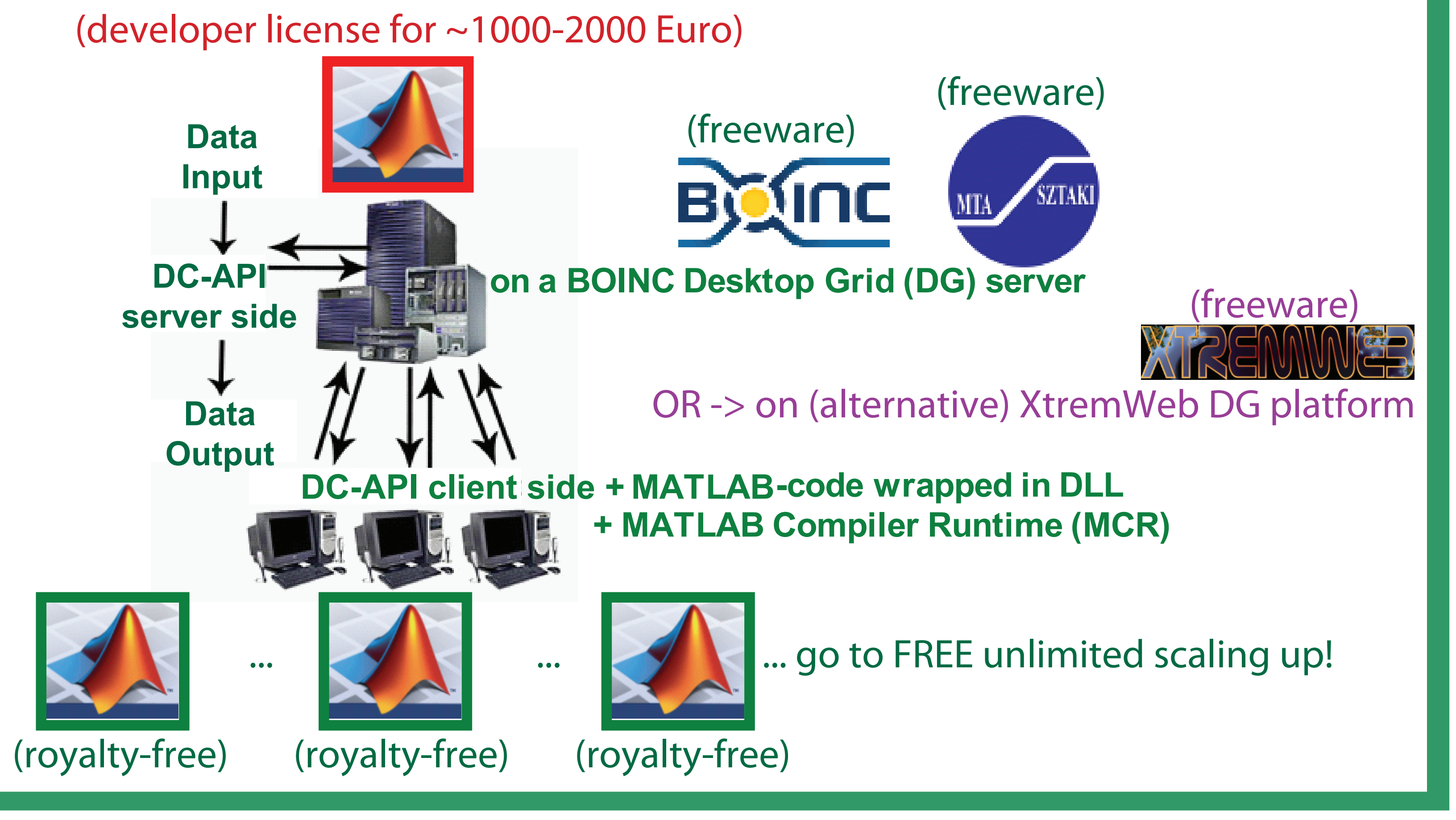

(developer license for ~1000-2000 Euro)
+ 1000-500 Euro per worker!
Server-side
(so-called "MATLAB Client")
+ Parallel Computing Toolbox
+ 1000-500 Euro per worker!
Scheduler
or
Job Manager
...
+ 1000-500 Euro
Client-side
(so-called "MATLAB Worker")
+ MATLAB Distributed Computing Server
Client-side
(so-called "MATLAB Worker")
+ MATLAB Distributed Computing Server
...
+ 1000-500 Euro
...
Client-side
(so-called "MATLAB Worker")
+ MATLAB Distributed Computing Server
...

Our original approach: MATLAB + DC-API + BOINC + XtremWeb

Minimal pre-requisites (commercial and free):
- MATLAB IDE + MATLAB Compiler (developer license for ~1-2 KEuro);
- MATLAB Compiler Runtime (MCR) (if you have MATLAB Compiler, then you already have it);
- Apache Web-server (freeware);
- MySQL database (freeware, GPL);
- BOINC software (freeware, LGPL);
- DC-API libraries for DG by SZTAKI (freeware) [3];
- or (alternative) XtremWeb desktop grid platform (freeware) [4].

Necessary steps to do:

1.Port sequential version of MATLAB application to parallel one.
2.Wrap MATLAB-code in shared library by MATLAB Compiler.
3.Port an application with the shared library to BOINC with DC-API calls.
4.Install BOINC Desktop Grid (DG) with a BOINC DG server.
5.Deploy the ported parallel MATLAB-coded application on the BOINC DG server.
6.Connect BOINC-workers to the DG-server (automatically).
7.Start the application and enjoy automatic jobs, resources, and results management.

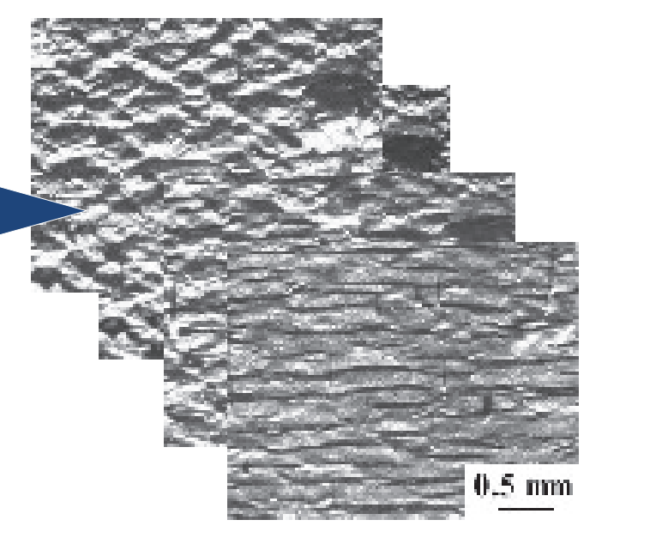

(developer license for ~1000-2000 Euro)
Data
Input
DC-API
server side
Data
Output
(freeware)
BOINC
(freeware)
MTA SZTAKI
on a BOINC Desktop Grid (DG) server
(freeware)
XTREMWEB
OR -> on (alternative) XtremWeb DG platform
DC-API client side + MATLAB-code wrapped in DLL
+ MATLAB Compiler Runtime (MCR)
...
...
... go to FREE unlimited scaling up!
(royalty-free)
(royalty-free)
(royalty-free)

2.Typical Experimental Workflows: Sequential and Parallel at BOINC and XtremWeb DG platforms

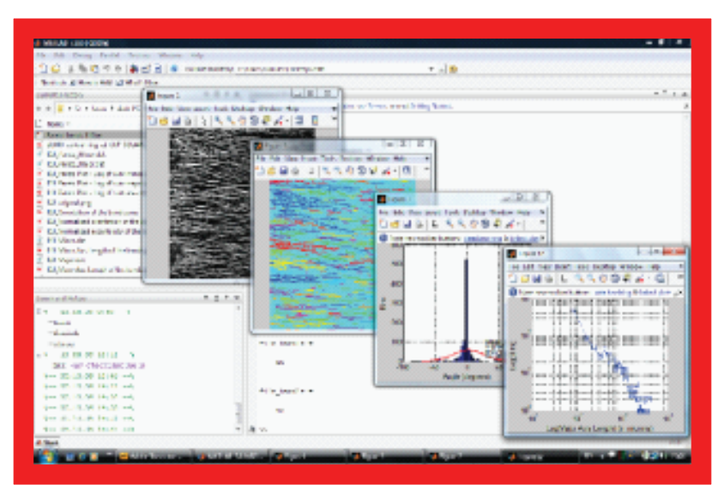
MultiscaleIVideoP with 4GL MATLAB-code was successfully ported and tested on BOINC-based local DG at IMP with Windows-workers[2] and on XtremWeb DG [4] at LAL with Linux-workers; now the work is going on delivering it to EGEE through EDGeS XtremWeb->EGEE Bridge [5].

Experimental machine
for mechanical test of metals

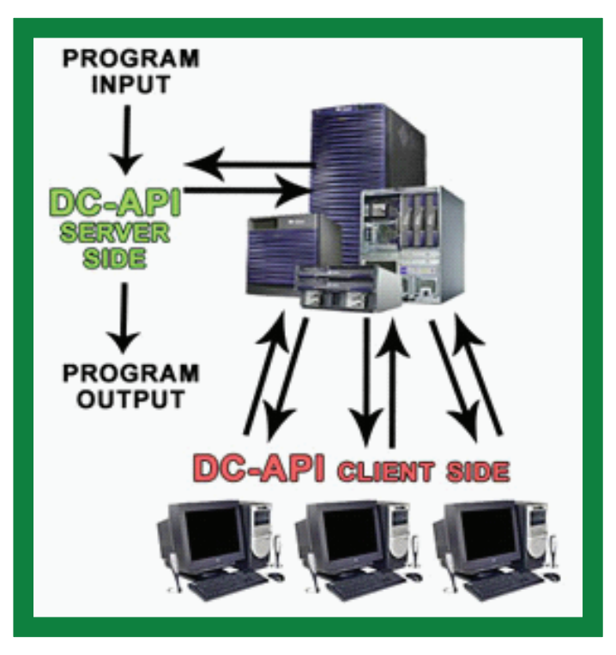
High-Speed
Video Camera
captures
metal surface
evolution

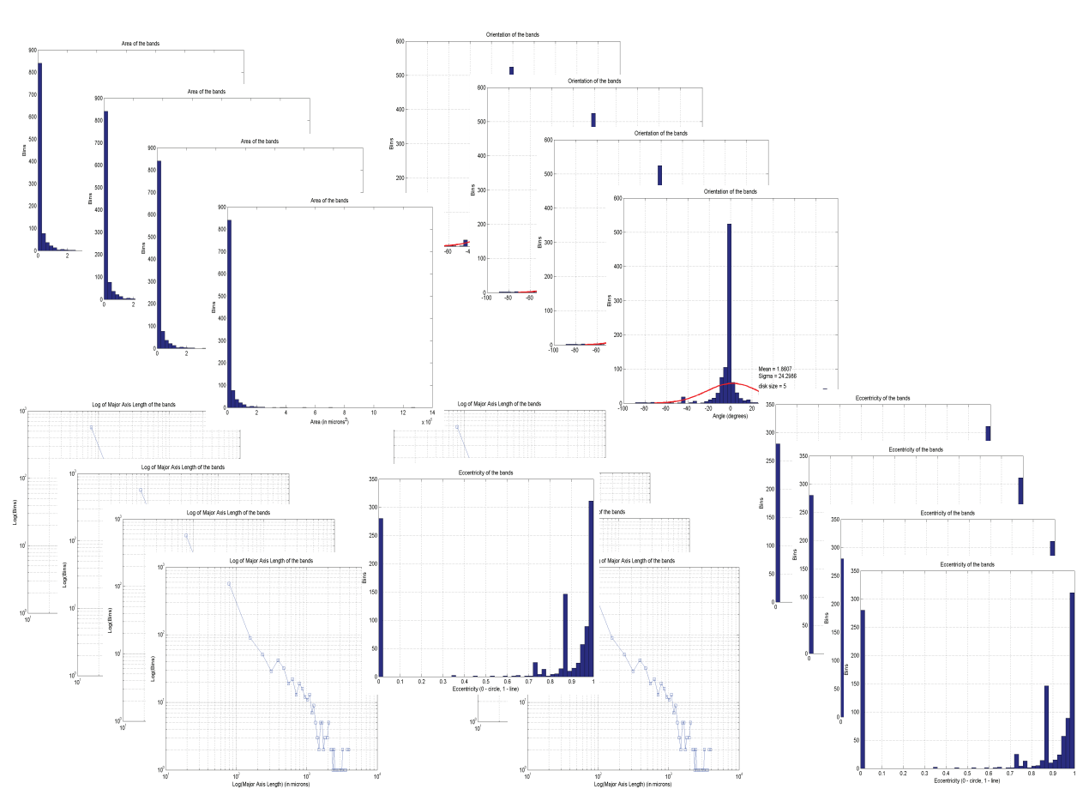

Output is Video File
($>10^5$ frames, >1 Mpixels)

BEFORE: SLOW
Sequential
Processing
in MATLAB IDE



NOW: FAST
Parallel Processing
in Desktop Grid




Scientific results (plots, datasheets)

The main achievement: applicability and feasibility of integration of typical scientific application coded in 4GL environments with DCI on the basis of Desktop Grid for high-performance distributed computing were proved, tested and demonstrated.

The further increase of performance is possible by delivering 4GL-based applications to global computing resources of EGEE on the basis of XtremWeb desktop grid platform [4] and EDGeS XtremWeb->EGEE Bridge [5].

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] Y.Gordienko, E.Zasimchuk, R.Gontareva, Materials Science Forum, vol 567-568, pp.169-172, 2007.
[2] O. Gatsenko, O. Baskova, and Yu.G. Gordienko, Proc. of Cracow Grid Workshop (CGW'09), Cracow, Poland, 2009 , pages 255-263.
[3] P. Kacsuk, J. Kovács, Z. Farkas, A. Cs. Marosi, G. Gombás, Z. Balaton, Journal Of Grid Computing, 2009 Vol 7 , No. 4, 439-461.
[4] F. Cappello, S. Djilali, G. Fedak, T. Herault, F. Magniette, V. Néri, and O. Lodygensky, Future Generation Computer Systems, 21 (2005) 417–437.
[5] E. Urbach, P. Kacsuk, Z. Farkas, G. Fedak, G. Kecskeméti, O. Lodygensky, A. Cs. Marosi, Z. Balaton, Zoltán; G. Caillat, G. Gombás, A. Kornafeld, J. Kovács, H. He, R.Lovas, Journal of Grid Computing, 2009, Vol 7, No. 3, pages 335 -354.