Porting the SEE-GRID EMMIL application into EDGeS infrastructure

Attila Csaba Marosi
Ákos Balasko
MTA SZTAKI
13-17. Kende u. Budapest 1111.
Hungary
atisu@sztaki.hu

Lívia Kacsukné Bruckner
International Business School - Budapest
2-4 Tárogató út Budapest 1021
Hungary
lkacsuk@ibs-b.hu

1. Introduction

Procurement has a critical role in the operation of manufacturing companies from both quality and cost aspects. While quality must be ensured by careful supplier evaluation process, cost reduction is only possible by increasing the competition among qualified suppliers. Business-to-Business e-commerce offers cost saving opportunities for buyers and improved market access for suppliers. While companies are already utilising the current well established forms of e-marketplaces, this is still an evolving field of research and the emerging new models can offer even more significant cost savings to participants. For manufacturers the ultimate cost of procurement is the total landed cost of the purchased parts and raw materials, i.e. the cost of commodities plus the cost of transportation. Electronic procurement marketplaces deal with the transportation only after the goods and their suppliers have been selected, which does not allow direct optimisation on the landed cost. In order to show a new approach to this problem a novel e-marketplace model called EMMIL (E-Marketplace Model Integrated with Logistics) has been developed by one of the authors (Kacsukné and Cselényi 2005). The model is built upon a composite auction where transportation providers place their offers step by step during the negotiation of the buyer and suppliers. The marketplace engine combines the proposals and selects the optimal combination for buyer.

The complexity of the auction algorithm and the composite bid structure make the optimisation task difficult. The solution known today requires lengthy computation that may take several hours to run on a single computer. Problems that require computing power beyond a single processor system can be executed on clusters and grid systems. The first parallel implementation of EMMIL was carried out within the framework of the EU SEE-GRID-2 project when EMMIL was adapted to the gLite based SEE-GRID. (Kacsuk et al. 2007.) Based on the promising results of the first implementation EMMIL has been also made available on desktop grids as one of the experimental applications of EDGeS infrastructure. In this article we describe how the SEE-GRID application has been ported to the DG infrastructure. In order to understand the special requirements of the porting process first we introduce the essentials of theoretical background briefly then describe the key steps of the SEE-GRID implementation.

2. The EMMIL procurement marketplace

Participants of the EMMIL procurement marketplace are the following: a single buyer looking for a set of tangible products in specified quantities, many suppliers offering certain quantities of the required goods and many third party logistics providers (3PLs) that undertake transportation. The products are assumed to be homogenous from the transportation point of view which means they belong to the same transportation class and are packed uniformly. We assume that unit load bulks will not be broken thus required quantities will be integer numbers The marketplace operates a composite reverse auction with discrete rounds of sealed bidding that alternate between sellers and 3PLs. Bidders have to go below the relevant best offers of the previous round. 3PLs in EMMIL can bid in a special structure of fix and variable costs. The fix part covers basically the costs of moving the vehicle from starting point (a supplier) to the end buyer. The variable cost is proportional to the volume transported where extra costs of handling can be accounted for.

2.1. Overview of the composite auction algorithm

1. The buyer issues a request for quote identifying the products, the quantities, highest price limits and the location where the goods should be delivered to. The buyer also sets the maximum number of suppliers (S) that can be awarded a contract in parallel.
2. Time limits are set for suppliers' bidding period, 3PLs' bidding period and for the total auction time
3. Suppliers' bid on prices of the required products (without quantity discounts)
4. The marketplace engine forwards the bids to the 3PLs
5. 3PLs' bid on fix and variable cost of transportation from suppliers who placed bids for the products.
6. The marketplace engine runs the optimisation algorithm combining the bids of suppliers and the bids of 3PLs in order to find an optimal solution with minimal total cost.
7. The bidding cycle is continued at step 3. until the auction time is over.
8. The suppliers and 3PLs who won in the last optimisation round will be the winners of the auction.

2.2. The optimisation problem

In this section we outline the optimization problem in an intuitive way without giving the actual formulas. This problem is a combination of two allocation problems, one for the products and another for the logistical services. With the optimization we have to answer the following questions

- What quantities of each product should be purchased from each supplier?
- Which 3PLs should transport the goods from the individual suppliers?

To answer the first question we introduced NxM integer variables, where N is the number of products and M is the number of suppliers present in the market. These variables denote the unknown quantities to be purchased product-wise and supplier-wise. For the second question we need LxM binary variables indicating whether a given 3PL is selected for transporting goods from each supplier. (L is the number of 3PLs taking part in the auction). The objective function will consist of the sum of the cost of the purchased goods and the cost of transportation. In the cost of transportation we have to multiply quantity variables and selection variables, thus the objective function will not be linear. Furthermore, this function will be neither smooth nor continuous because the special cost structure of the 3PLs. In order to solve the optimization problem a linearization algorithm was introduced, resulting in a piece-wise linear objective function with binary and integer variables based on Eso et al. 2001. The resulted mixed integer linear programming (MILP) problem could be easily solved using any standard commercial MILP packages – in theory. Our simulation experiences showed that the segmentation resulted in a big model size with a high number of integer variables which led to very long solution times using standard commercial MILP packages. Even in a small size case where we specified 20 kinds of products, 11 sellers and 11 logistics providers it was possible to set the parameters (prices and required quantities) so that in unlucky cases we ended up with computation times of 8-10 hours on an up-to date PC. Obviously such a long execution time inhibits the required number of tests as well as the creation of a real auction system.

The remedy for the problem can be the use of parallel and distributed systems like clusters and grid systems. To create a distributed solution we decided to exploit the constraint for the maximum number of suppliers (S) set by the buyer at the beginning of the auction. The basic idea was to subdivide the MILP problem into a number of similar problems but with only a few variables (determined by constraint S) that can be solved in a short time and can be done in parallel. The optimal total costs of the different solutions then could be compared and the minimal cost solution could be selected.

2.3. The parallel optimization algorithm

1. Piecewise linearization is performed and the coefficients of the MILP problem are calculated.
2. Suppliers are pre-filtered, worst suppliers are eliminated, the best U of them is kept only
3. Parameters for all S-ad combination of the best U sellers are generated for parallel processing.
4. All the MILP problems are set up and solved in parallel
5. The individual optimums are compared and the lowest one is selected as optimal solution.

3. Implementing EMMIL application on SEE-GRID infrastructure

The SEE-GRID application of EMMIL was implemented with the help of P-GRADE portal of MTA SZTAKI. The workflow is based on the most common type of parallel executions in grid infrastructure; the same core program should be executed on various input data and parameterization. It means that the different instances of the core program can be separated from each other, and can be executed in parallel. This type of parallelization, called Parameter Sweep or Parameter Study (PS), is one of the most supported parallel-execution possibilities of P-GRADE Portal, therefore the workflow was developed as a PS application in P-GRADE Portal (see Figure. 1). The first stage of solving the optimization problem is to generate the sub-problems for all the allowed combinations of suppliers (note that buyer can limit the number of the favored suppliers). This task is performed by a specific job called “Generator”. This type of jobs in P-GRADE Portal is used for generating a set of input files for the parallel jobs according to the parameterization of the problem. In the next stage the portal executes the core program, a Mixed-Integer Linear Programming Problem solver called lp_solve[1], in as many instances as many input files were generated by the Generator. This is the parallel part of the application. Finally a specific job called “Collector” will start after all parallel jobs were finished. This job will save the results, compare them to each other and find the solution with the lowest total cost.

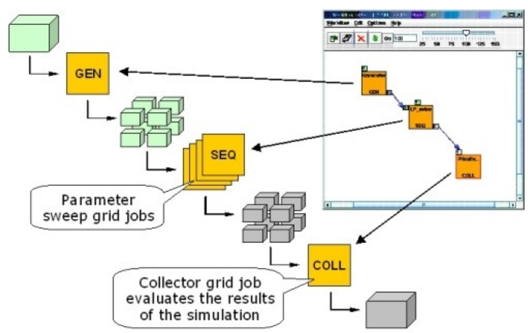

Figure 1: Workflow of the EMMIL application

Execution of EMMIL is made user-friendly with the help of a new portlet developed in P-GRADE Portal 2.5 to make bid-taking and auction-creation easier. The main stages of the business process are illustrated by Figure 2. As it is shown, parameterization for one execution consists of many separated sub-parts. First of all, the buyer has to create an auction by specifying the required products along with the quantities and setting all the parameters as described in 2.1. Then the suppliers can select this auction, and place bids for the products. After the suppliers’ bidding time has expired, third party logistics can place bids for the transportation from the different suppliers to the buyer (see Figure 3).

[1] **lp_solve** is an open source program that is free under the GNU Lesser General Public License. (Berkelaar et al. 2004.)

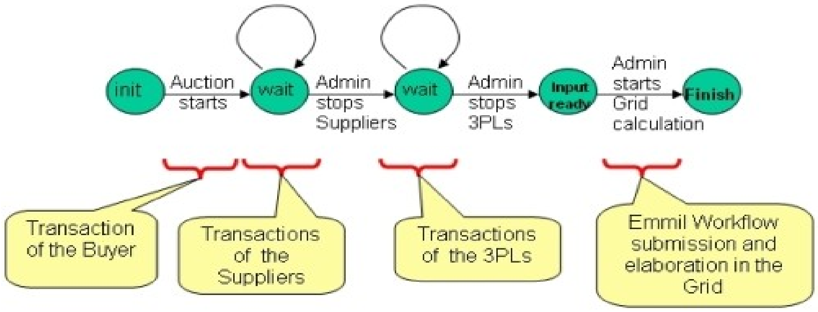

Figure 2: Process of the auctions

When the time set for 3PLs' bidding is over, the input parameterization will be passed to the Generator job as an input file, and the workflow will be submitted into the grid as a PS application of P-GRADE Portal. When the workflow finishes, all users can view a report about winners of the completed auction round. The auction can be continued with the next round according to 2.1. or closed as required.

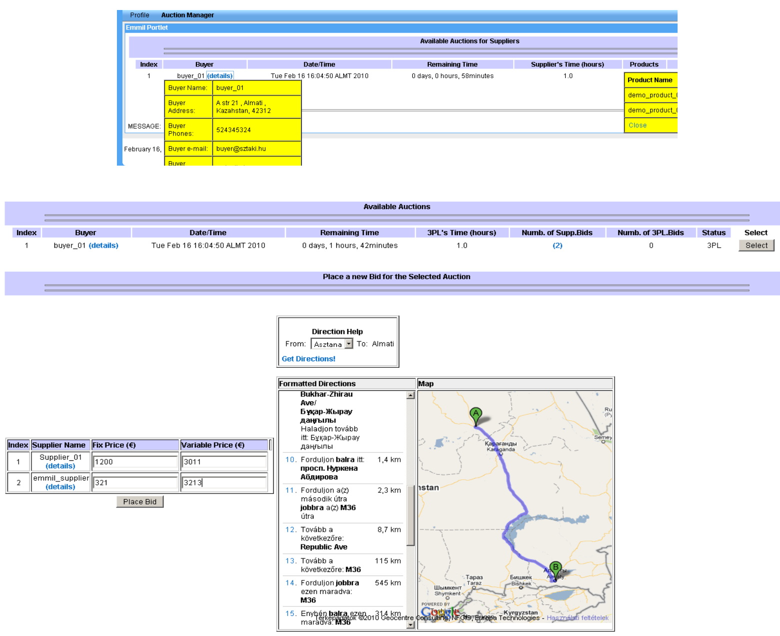

Figure 3: Bid-taking interface for Third Party Logistics

4. Porting EMMIL to the EDGeS Infrastructure

Within the framework of the EDGeS project we have investigated how to adapt EMMIL for a desktop grid infrastructure. After implementing EMMIL on the SZTAKI Desktop Grid, the next step was to investigate how the EMMIL parameter sweep jobs can be transferred from the SEE-GRID environment to the BOINC-based desktop grid systems of the EDGeS infrastructure. The original EMMIL application was ported to EGEE as a Parameter Study Workflow developed in P-GRADE Portal. It means that parallelization is implemented by using a Generator job in the workflow. Then the parallelized core-program runs with a lot of instances. Only this core-program was ported to EDGeS, but since Desktop Grids utilize diverse resources part of the task was to provide a Windows version of EMMIL.

When all of the instances are finished and generated the outputs, another program, the Collector job, runs on EGEE too, collects the results of the core-programs, and aggregates the final result. This means that only the core-application required modification, other ones run on EGEE as they did before the porting. The connections between the Generator and Collector running on EGEE and the BOINC-based parallelized core-program are solved by the EGEE→DG bridge ("EDGeS Bridge Services" on Figure 4) developed by EDGeS. The core-program runs with multiple data, so in the DG's point of view, the core-program is a sequential program. A newly developed tool called GenWrapper[2] (Marosi et al. 2009) was used to port the core-program. GenWrapper is a tool to make applications run on Desktop Grids without modification to the source code, for example resolve all filenames used by the real program to BOINC generated ones. There is no modification required within the source-code of the real program, but an additional wrapper algorithm must be downloaded to the clients to execute the application. On the other hand, BOINC requires a program to validate the results arrived from different clients that executed EMMIL with the same input file. This type of redundancy is used to ensure that the returned results do not contain any errors (e.g. caused by malfunctioning hardware). BOINC provides a set of validators for the most common type of results (e.g. comparison of text files, binary comparison of files, etc.), but the existing validators could not be used in case of EMMIL, because the result files may be different, depending on the Operating System, where the core-program has been executed. The linear programming problem solver models large real number differently on Windows and Linux systems. That's why an own specific validator had to be developed for EMMIL.

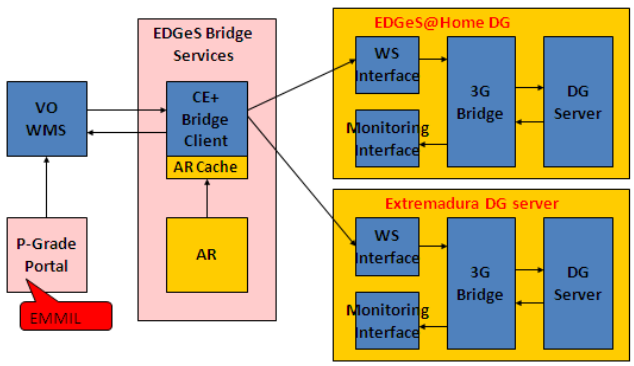

Figure 4: The EDGeS EGEE→DG infrastructure

Currently the EDGeS@Home BOINC project[3] and Extremadura Red Tecnologica Eductativa BOINC projects supports the EMMIL application.

5. Results and conclusions

The primary result of porting the EMMIL application is that the execution of EMMIL workflow becomes much faster, due to the utilisation of BOINC resources. Since the most time consuming part of the workflow can also be scheduled to BOINC from now on and since (public version of) BOINC projects usually provide much larger amount of resources than gLite based, the EMMIL workflow progresses much faster than before thanks to the EDGeS infrastructure as well.

The second benefit of the porting project is the experience we gained from it. This project proved that the toolkit we developed in the frame of EDGeS can efficiently support the porting process and the seamless interoperation between EGEE and BOINC infrastructures; thus other EGEE applications can be easily ported to desktop grids as well. In addition, the validator developed for EMMIL is also an asset for grid users since it can be further enhanced to support a wide range of applications using lp_solve on the grid.

6. Acknowledgements

The work presented in this paper was supported by the EDGeS FP7 project funded by the European Commission under contract number INFSO-RI-211727.

[2] http://genwrapper.sourceforge.net/
[3] http://home.edges-grid.eu/home/

7. References

Berkelaar, M. Kjell Eikland, Peter Notebaert (2004) ”lpsolve manual” http://lpsolve.sourceforge.net/5.5/

Eso, M. S. Ghosh, J. Kalagnanam, and L. Ladanyi, (2001) “Bid evaluation in procurement auctions with piece-wise linear supply curves”, IBMResearch, Yorktown Heights, NJ, USA, Research Report RC 22219.

Kacsuk,P, Kacsukné B. L., Hermann, G., Balasko,A (2007): Simulation of EMMIL E-Marketplace Model in the P-GRADE Grid Portal, Proceedings of ESM'2007 International Conference, Malta, pp. 569-573.

Kacsukné Bruckner, L. – Cselényi, J. (2005).: “Principles and Algorithms of EMMIL Marketplaces”, IEEE Conference on E-Commerce München 19.-22. July 2005. 426-430.

Marosi, A. Cs., Balaton, Z., Kacsuk, P. (2009): GenWrapper: A Generic Wrapper for Running Legacy Applications on Desktop Grids. 3rd Workshop on Desktop Grids and Volunteer Computing Systems (PCGrid 2009), 2009 May, Rome, Italy