УДК 681.3.07

L. Globa, T. Kot, A. Luntovskyy, D. Gueter

# Parallelization techniques in the tasks of transaction processing and for CAD of telecommunication networks

**В статье рассмотрены современные технологии распараллеливания вычислительных процессов и организации многопоточной обработки данных. Описана разработанная усовершенствованная технология распараллеливания вычислительных процессов. На конкретном исследовании показано применение данной технологии, а также полученный выигрыш в области применения алгоритмов моделирования высокой степени сложности в системах автоматизированного проектирования телекоммуникационных сетей.**

**Advanced parallelization and multi-threading techniques are discussed, and the developed one is described. The case study shows deployment of the techniques and the obtained gains in the area of high-complexity simulation algorithms in CAD of telecommunication networks.**

## Introduction

Nowadays the networked computing systems (NCS) (including Interned-based ones) have to process considerable amounts of information and transfer huge volumes of different data types. With the development of such networked systems computing facilities for applied tasks in different practical areas, for instance, Computer-Aided Design (CAD), physical phenomena simulation and etc., are not enough. State-of-the-art concepts of multithreading, paralleling, redundancy and replication are to be considered. It is the main reason of parallel computing systems development and high-performance computing usage for raising the modern computing systems efficiency.

## 1. Parallelization techniques and tools: state - of - the-art

Parallel computing organization is a broad labor-consuming task, that's way it is subdivided into next stages [1]:

1. parallel computing systems development;
2. parallel computing efficiency analysis for computing acceleration and degree of all possible computing facilities usage estimation;
3. parallel algorithms development for applied tasks of different practical areas solution;
4. parallel program systems development;
5. software for parallel computing systems development.

Each step of a parallel computing methods organization can be considered as a separate direction. In general there are four steps of parallel algorithms development [2, 3]:

1. computing division on the independent parties;
2. information dependence separation;
3. set of subtasks scaling;
4. subtasks distribution between computing elements.

The following parallel programming technologies can be used for programming complex of parallel performance algorithm development [1, 4 – 6]:

1. Message Passing Interface (MPI) – message passing model [4];
2. Open Multi-Processing (OpenMP) – control parallelism model [5];
3. High Performance Fortran (HPF) – data parallelism model;
4. DVM – data and control parallelism model and others.

## 2. Advanced parallelization and multi-threading

Before the parallelization and multi-threading techniques can be efficiently used and their implementation becomes possible firstly the offered Three-Level Distributing [7] is necessary. It is based on three-level hierarchical model development with these levels further complex analysis and optimization in distributed information environment. Each level provides special tasks analysis and performance:

1. Enterprise business processes (EBP) level considers business model [8] in general and application area business processes in particular.
2. Computing business processes (CBP) level considers the software system business processes: computing, information exchange, etc.
3. Tasks performance level provides separate tasks of computing processes implementation.

Business processes level is to provide:

- the conformity between NCS functionality and enterprises business tasks;
- formalizing the requirements analysis when information processing design;
- raising the probability of their proper implementation.

On the EBP level Business-model and model of EBP and operations are developed. Time and required resources for EBP and operations execution are analyzed and optimized using developed algorithms [8]. The results are used for further business model analysis, using graph analysis methods [9].

On CBP level computing processes graph is optimized and then performed as User Interface(UI) - Web Services (WS) graph.

On tasks performance level CBP is considered as set of separate tasks. Each task is analyzed to be parallelizable. The main analysis criteria are: paralleling appropriateness and availability of necessary computing resources. If paralleling is appropriate, the technologies of parallel programming are chosen and possibility to attach high-optimized multithreaded mathematical operations libraries are considered.

## 3. High-complexity simulation algorithms in CAD of telecommunication

Several areas require algorithms and techniques of high performance computing to reduce computing time and computational power at the same time. Among them the area of high-complexity simulation algorithms in CAD of telecommunication networks is, where wireless topologies with optimal transmitted power, frequencies and propagation environments should be found, transmitted data threads have to be analyzed and so on.

The implementation of high-complexity simulation algorithms in CAD of telecommunication networks is described below.

*Background*

The experience is based on the realization of practical task, called „CANDY - Computer Aided Network Design Utility Framework" [1, 10].

Focuses**:**

High-performance fixed, wireless and mobile communication networks have become more and more important for global corporations, small and medium enterprises, public organizations, and universities. The efficient and economic design of such networks, however, remains a challenge, and existing design tools only provide limited and hardly integrated support. Therefore the CANDY is a much more integrated design methodology with associated tool support.

CANDY is notable for major conceptual and theoretical challenges, as follows:

– An integrated design methodology forming a complete design workflow. It especially focuses on the combination of wired IEEE 802.3-LAN and wireless network design (IEEE 802.11 and 802.16).
– A modern network design markup language (NDML), an XML-based notation that serves as a uniform way of representing all major network elements (active and passive), their detailed technical properties, and their interconnections and related configuration issues. As opposed to existing vendor-specific notations, NDML is based on open standards and enables interoperability and portability of network designs.
– CANDY is a framework with a significant set of design tools. This includes design editors, consistency checks, transformation tools, specific wireless network design tools, and integration of existing simulation environments. As a common notation, NDML once again serves as the common glue for these tools.

*Deployment of High-Performance Computers Aimed at Parallelized Solution of WLAN/WiMAX-Modeling Tasks*

Accurate planning is necessary in order to build cost-effective and efficient wireless networks. The CANDY includes a number of developed tools with the intention to support and simplify network planning. For example, the CANDY Site Finder supports planning of wireless networks (WLAN and WiMAX) by simulating radio propagation. Depending on the used algorithms and the complexity of the considered scenarios, such simulations can take much computing time. Parallel computation can significantly reduce computational time [10]. It is considered how the available resources can be used to speed up simulations of radio propagation in wireless networks. Therefore the available systems are examined and methods for parallel programming are evaluated. Then propagation models are implemented and tested on high performance computers in order to analyze the performance improvement.

*The Dominant Path Prediction* (DPP) algorithm [1,6] provides good results for simulation of outdoor WiMAX networks. Unfortunately, it possesses a high computational complexity. The needed computing time can be reduced significantly by employing parallel programming and virtualized services [9].

*The Message Passing Interface* (MPI) [4] describes the message exchange between parallel processes on distributed computer systems. As a programming interface, it defines operations and their semantics. Different implementations of MPI exist, which are tailored for specifics of the used computer systems: Open MPI, portable implementation of MPI (MPICH2) [11], Hewlett Packard Message Passing Interface library (HP-MPI) [12] and others. All implementations provide a consistent interface, so there is no need to adapt the source code to a specific MPI implementation. Several implementations are specially tailored for dedicated communication networks, where they are performing best. On the other hand, they may not support other systems. It's up to the user to choose a MPI implementation that suits to the underlying system. In general, the operator of a computing center will only provide implementations for the present hardware.

*OpenMP* [5] is another interface for distributed computing. It is being developed since 1997 by different hardware and compiler manufacturers. While MPI uses message exchange at process level for parallelization, OpenMP works on thread level. Therefore, OpenMP defines function calls that instruct the compiler to distribute parts of a program among multiple threads. For example, a for-loop is predestined for being shared among multiple threads and processors respectively. For this purpose OpenMP uses the Fork-Join model [5].

*Case Study: Parallel Implementation for Dominant Path Prediction*

*The SGI Altix 4700 was chosen for experimental task* a powerful large-capacity computer. It works with a distributed memory; therefore it is ideal for applications that were parallelized using OpenMP. Another reason for preferring OpenMP over MPI is the significantly shorter training period that is needed before productive use of the system is possible. OpenMP is employed for development of parallel applications within the CANDY *experimental task* [1, 10]. Initial development with OpenMP can take place sequentially as usual. Parallelization can be implemented at a later stage of development. So often it is possible to use sequential program code as base for development of a parallel application. By contrast, when using MPI, applications have to be re-implemented in large part. Recently Dualcore and Quadcore CPU's found their way into desktop computing. Unfortunately, most applications don't make effective use of multiple cores. OpenMP allows a subsequent parallelization of applications developed with C or C++ with little effort.

The Dominant Path Prediction Algorithm [1, 10, 13] was implemented using OpenMP and executed with different setups on the SGI Altix 4700 with the following characteristics:

– 1024 dual-core Intel Itanium processors and 6,5 TB main memory.
– Communication is possible within the five partitions via the shared memory SAN/NAS.
– Altix is especially recommended for highly parallel and memory-intensive applications.
– MPI/OpenMP/MPI with Java -programs using hundreds of processes achieve excellent performance on the Altix.
– It ranked number 49 in the Top500 list of the fastest computers of the world.

The used scenario with a resolution of 1000x697=697000 points consisted of 64 buildings with 360 walls and two base stations. Table 1 shows the total computing time needed depending on the number of used cores. Fig. 1 gives a graphical representation of the relation between the number of cores and the needed computing time. An exponential approximation of the curve gives:

$$T = K \cdot N^{-\alpha}, K = 8021.2,\ \ \alpha = 0.9462$$

where T – computing time [s], N – required number of cores. The shape of the performance curve is typical for parallel computers and shows the time gained by using multiple cores. As shown by these sample calculations, the DPP is computationally intensive, which makes it a candidate for computation on high-capacity computers. Using 1000x697 simulation points, which means a resolution of 73,4 cm for the given scenario, calculation under single-core conditions takes 2 hours and 23 minutes. Using five cores it is possible to reduce the computing time to less than 30 min. The used scenario is smaller than common WiMAX-cells, which usually have a diameter of above 2 km.

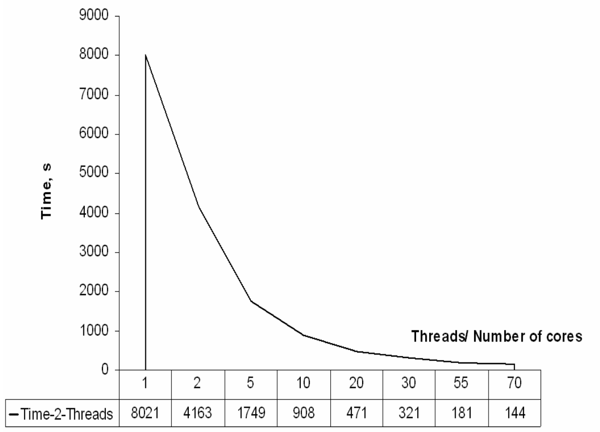


**Fig. 1. Results of high Performance Computing**

**Table 1. Computing Time for DPP with a complex Scenario**

| Number of threads/ cores | Required time [s] |
|---|---|
| 1 | 8021 |
| 2 | 4163 |
| 5 | 1749 |
| 10 | 908 |
| 20 | 471 |
| 30 | 321 |
| 55 | 181 |
| 70 | 144 |

## Conclusions

Existing parallelization techniques are discussed; developed advanced parallelization and multi-threading technique is described.

The case study demonstrates that the offered distributing and parallelization approaches allows shortening time of computing processing in the area of CAD of telecommunication networks.

Despite of evident advantages of existing design tools WLAN and WiMAX-nets there is however a wide field to the achievements in mid-term:

– Increasing of accuracy (e.g. up to some 3dBm);
– Raster step minimization (e.g. down to 15cm x 15cm spot or 2m x 2m by WiMAX-cells);
– Performance improvement (due to deployment of less complex algorithms, computing virtualization and parallel programming);
– Costs optimization.

A significant gain of time by means of parallelization can be achieved for the wireless path propagation simulation out-door method DPP. For the further Cost 231 Walfish Ikegami [13] parallelization is only attractive for high resolutions and large extends of the analyzed scenario. Given the low computing times needed by the Multi Wall, parallelization is dispensable.

Parallelization doesn't relate only to the algorithm itself. Also, different scenarios can be processed in parallel (multi-variant analysis). For example, the individual floors of a building can be simulated in parallel, instead of simulating them sequentially on a single-core machine.

**References**

1. *V. Krukov.* Parallel Programs Development for Computing Clusters and Networks (in Russian), Keldysh Institute of Applied Mathematics RAS (Online 2010): http://parallel.ru/tech/
2. *L.Globa.* CASE Tools for Distributed IT-System Accounting Multithreading", Polish J. of Environ. Stud, Vol. 16, No. 5B (2007), pp. 135-140.
3. *L.Globa,* T.Kot, A.Schill. Business-processes optimization while information systems design, ACS'2008 Proceedings, Poland, 5 p.
4. *MPI* (Online 2010): http://www.mpi-forum.org
5. *OpenMP* (Online 2010): http://openmp.org
6. *A.Luntovskyy,* S.Uhlig, D.Guetter. Cross-Layered Wireless Network Planning and Modeling Methods and Tools, 5th IEEE and ACM ICWMC, June 2009, Leipzig, pp.1417-1421.
7. *Larisa Globa*, Tatiana Kot, Kateryna Iermakova. Three-Level Approach to Information Processing, Proceedings of the conference CADSM'2009, p. 411-412.
8. *Professor Globa Larysa*, Prof. Dr. rer. nat. habil. Dr. h. c. Schill Alexander, Tatiana Kot. Business-processes optimization while information systems design, "Polish J. of Environ. Stud ", Vol. 17, No.4c (2008), p.213-216.
9. *R. Basaker*, T. Saaty, "Finite graphs and networks", 1974, p. 368.
10. *A.Luntovskyy*. Integration Concepts for Computer-Aided Design Tools for Wired And Wireless Local-Area Networks, Monograph: Shaker Verlag Aachen, 2008, 196p.
11. *MICH2* (Online 2010): http://www.mcs.anl.gov /research/projects/mpich2/
12. *HP-MPI* User's Guide (Online 2010): http://docs.hp.com/en/B6060-96022/index.html
13. *A.Luntovskyy,* S.Uhlig, D.Guetter, A.Schill. Integrated Design Tools for Combined LAN. IEEE CriMiCo Conf., Sevastopol, Sept. 2008, pp.360-361.