II.1 Distributed Computing and Asynchronism

Permanent Staff: A. Leader, Didier El Baz (CR-HDR); B. Members, Moussa Elkihel (MCF), Jean-Michel Enjalbert (MCF);

Objectives, research topics and positioning

1) Scientific and technological environment
The evolution of Computer Science during the recent years has changed our vision of computing, programming and using machines. The size and complexity of applications and problems to treat have led to a quest of high performance that has imposed the concept of parallelism in computer architectures. This phenomenon can be seen almost everywhere, i.e. from supercomputers to desktop and laptops. Now, the architectures of processors are based on the use of many cores and specialized units like GPU are even used for general purpose parallel computing. Simultaneously, networks technologies have known an important rise, parallel and distributed computing have converged and concepts such as clusters or peer to peer networks have emerged.

2) Open questions
The lack of centre and global clock together with a priori unbounded communication delays (i.e. asynchronous model), massive parallelism and heterogeneity lead to interesting problems. We note that technical challenges in distributed computing are not limited to the problems quoted above (one could quote also cooperation in presence of faults and topology changes and many more challenges). More generally, these technical challenges can be identified as part of broader “Grand Challenges” like:
   a) Easiness of programming and using massively parallel or distributed architectures
   b) Quest of efficient parallel or distributed methods.
Some computer scientists like Rodney Brooks and Michel Raynal consider that those “Grand Challenges” are among the most important ones in Computer Science and that the future of this science depends greatly on our capacity to address these important questions. These circumstances are very propitious to research works led by scientific researchers of our team who have worked for a long time on parallel and distributed computing.

3) Goals and research topics
The main goals of the Distributed Computing and Asynchronism team, DCA for short, are to facilitate the use and programming of parallel and distributed computing systems at a very large scale and to propose new algorithms that will permit one to use more efficiently parallel and distributed architectures. The activity of DCA spans a domain of research that goes from the design and analysis of distributed algorithms (e.g. see [ACL308]), communication studies (see [ACTI681]) and load balancing techniques analysis (see [ACL2]), to the design of environments for distributed computing (see [ACTI80] and [INV49]). The team DCA works particularly on high performance distributed computing on general purpose architectures and dedicated architectures; e.g. DCA deals with the solution on supercomputers or peer to peer networks of large scale numerical simulation problems like boundary value problems (see [ACL308]) and difficult integer programming problems (see [ACL2]).

The team DCA works on scalability and heterogeneity (see [INV49]); it concentrates in particular on asynchronism (e.g. see [ACL502]), a key concept to understand the very nature of distributed computing and address efficiency issues. As a matter of fact, asynchronous algorithms are well suited to all type of networks; moreover, one cannot reasonably think at synchronizing a large number of processors or processes in a massively distributed system. It turns out also that the use of asynchronous algorithms permits one to obtain better efficiency when parallel tasks are unbalanced or when convergence is monotone. Finally, asynchronous schemes of computing permit one to obtain some fault tolerance since one shows that asynchronous algorithms can converge even when some data are lost. It is important to note that asynchronism is about to become one of the main topics in Computer Science. This topic can be found in various areas of research which are the numerous facets of the same concept which consists in letting entities go at their own pace; e.g. one finds this concept in circuits, communications, processes and algorithms. For example, one speaks of asynchronous circuits when circuits have no global clock. One speaks also of asynchronous communication in message passing systems when communications aspects are not connected with synchronization aspects e.g. Remote Memory Access, RMA, or situations where there is no shake hand like protocol; one advantage being the possibility to overlap communication and computation. One also speaks about wait free processes, threads and asynchronous coprocessors. Finally, one speaks about asynchronous iterations or asynchronous iterative algorithms when components
of the iterate vector are updated without any order nor synchronization. Let us conclude by saying that these last schemes of computation are becoming more popular nowadays and that research works on communication libraries like Open MPI are pursued in University of Tennessee and other universities, in order to facilitate the use of asynchronous algorithms. Users of massively parallel architectures, like Christian Engelmann at Oak Ridge National Laboratory, recommend also the use of asynchronous algorithms for efficiency and fault tolerance purpose. DCA deals also with self organization, another key concept in distributed computing. Far beyond self optimization, self organization is an important property in distributed decision making that permits one to obtain efficient behavior in hazardous situations or in the presence of faults and to insure everlastingness of applications. One can find for example this property in human or animal societies like swarms where several strategies are developed in order to insure a certain level of efficiency of the group; without such a property one can hardly think at effectiveness of entities in the system and efficiency of the whole system in general (it turns out also that any system that cannot adapt itself to an evolution context is finally condemned to dysfunction and extinction). Note finally that non synchronicity and self organization are somehow related to cooperation which is central in the domain of parallel and distributed computing, where collaborating entities share total work.

4) Situation of the team
The DCA team was founded ex nihilo in October 2007 according to the advices of the previous Evaluation Committee of LAAS-CNRS. Researchers in our team are affiliated to two important schools in France: first of all, a school dealing with scientific computing and in particular parallel iterative methods for PDE, led by Jean-Claude Miellou; secondly, a school in Combinatorial Optimization and Computer Science led by Gérard Plateau. As previously said, the situation of our team is quite unique if one thinks at our research works which are conducted at the intersection of Computer Science, Applied Mathematics, Control Theory and Numerical Analysis. This excellent situation has permitted us to establish fruitful collaborations with researchers in this country and abroad; those collaborations are detailed in the last Section of our report. In particular, we have worked and published with almost all research teams in Computer Science, Applied Mathematics and Numerical Analysis in the world that have dealt with parallel or distributed asynchronous algorithms and we have delivered many reports on scientific papers that have been submitted to major scientific journals like Journal of Parallel and Distributed Computing, IEEE Transaction on Parallel and Distributed Systems, Parallel Computing and SIAM Journals as well as projects submitted to ANR.

It is worth noting that there is no team in LAAS-CNRS doing research work specifically on this topic and that there are relatively few teams in this country that are developing research work in this domain. Parallel and distributed computing is widely studied in the world. Nevertheless, except for few teams in Universities like MIT and Temple University, Philadelphia, that have produced research works at the intersection of Computer Science and Applied Mathematics, the bulk or research works is in Computer Science, i.e. programming models, programming environments, fault tolerance and security or performance study.

Highlights and Major Achievements

1) Highlights
Our results during the period mainly deal with the design and analysis of parallel and distributed algorithms. Our results concern two main topics: the solution of large scale systems of equations mainly derived from boundary value problems via parallel or distributed asynchronous iterative algorithms and sequential or distributed solution of difficult integer programming problems.

1.1) Distributed asynchronous iterative algorithms
We have been working on this topic for a while. We have considered in particular mathematical modeling and convergence analysis (see [ACL6]), stopping criteria and study of round off errors (see [ACL56] and [ACL502]) and study of performance (see [ACL308]). We have contributed to the design and study of a new class of parallel or distributed iterative algorithms, i.e. Asynchronous Algorithms with Order Intervals, AAOI for short (see [ACL6]). The AAOI class is a general class of parallel or distributed iterative algorithms whereby several processors can perform computations, i.e. updating phases, at their own pace, without any order nor synchronization. One of the main features of this new class of algorithms is that the current value of the components of the iterate vector, i.e. a partial update which is not necessarily labeled by an iteration number can be used in the computation (see Figure 1), whereas only values labeled by an iteration number are used with the classical asynchronous iterative model. This new feature is particularly important in the case of convergence under partial ordering since it permits one to obtain better efficiency of parallel/distributed algorithms. This feature is also attractive when using two stages iterative methods or bloc iterative methods since it is not necessary to wait for the
completion of an iteration to get updates. As a consequence, the AAOI model fits new communication mechanisms like RMA and introduces more flexibility in the way data can be used in computations. Convergence results in a contraction context have been established for AAOI in [ACL6].

Several theoretical stopping criteria in relationship with forward, backward or absolute errors have been proposed for perturbed asynchronous fixed point methods in the presence of round-off errors (see [ACL56] and [ACL502]). The last case is particularly interesting since all tests are made within the same macro-iteration. We have studied distributed termination detection in [ACTN5]. In particular, we have proposed a method based on activity graph and messages acknowledgement that tends to minimize the number of exchanged messages.

Issues related to the implementation of asynchronous algorithms on message passing architectures have been studied in details in [ACTN61] with emphasize on communications aspects. Efficiency of synchronous and asynchronous iterative schemes of computation has been studied for nonlinear convection diffusion problems in [ACL308]. In this last paper, computational results on IBM SP4 machines with more than one hundred processors have been analyzed in details for different communication frequencies (see Figure 2) showing the interest of asynchronous algorithms with high communication frequencies.

We have also worked on easiness of use of distributed architectures. Several generic solvers have been proposed for problems linked both to computing and networking in distributed peer to peer applications. These solutions concern initiation and termination of computation, task transparency and routing. A first series of computational results with application to shortest path problems solved via auction algorithms are presented in [ACTN80]. A self adaptive protocol for peer to peer high performance computing has also been proposed in [INV49]. The self adaptive protocol which combines micro protocols is developed in the framework of ANR Project CIP that deals with the design of a decentralized environment for distributed peer to peer computing (see also next Section).

![Figure 1: data exchanges in AAOI, updates exchanges (boldface) and partial updates exchanges (others)](image1)

1.2) Distributed methods for integer programming

Many integer programming problems are very difficult to solve; one speaks of NP-complete problems. As a consequence, the solution of these problems is a great challenge for parallel/distributed computing. Nevertheless, most parallel/distributed solvers are derived from sequential solvers and the attractiveness of a parallel method mainly depends on the sequential method it is derived from. This is why DCA does also some research work on sequential methods for integer programming problems before deriving efficient parallel or distributed algorithms, as we shall see in the sequel.

DCA has concentrated on knapsack problems, e.g. multidimensional knapsack problems and knapsack sharing problems that arise in several practical contexts like cargo loading, capital budgeting, processor allocation and cutting stock problems. We have dealt with complexity issues. We have concentrated in particular on the origin of difficulty and the generation of difficult instances (see [ACTN241] and [TH147]).

DCA has produced a series of works on the solution of multidimensional knapsack problems. Several heuristics that outperform well known heuristics in the literature in time and quality of solution have been proposed in [ACTN178] and [EXT1]. Exact cooperatives methods that combine dynamic programming and branch and bound have been studied in [ACT1063] and [ACT189]. Research works have also been done on domain reduction techniques. These studies permit one to precondition problems. We have considered in particular constraint rotation methods (see [ACL551]). Let us note finally that a method based on Lagrangian relaxation has also been proposed for the solution of...
combinatorial problems with equality constraints (see [ACTN170]).

A parallel dynamic programming algorithm based on lists method with pseudo polynomial complexity has been proposed for knapsack problems (see [ACL2]). Data exchange management plays a prominent part in this parallel method. Several load balancing techniques have been studied in order to obtain efficient parallel methods (see [ACL358]).

2) Major Achievements

We would like to put forward the set of results on asynchronous iterative algorithms quoted in the previous subsection since these results concern all the aspects related to this topic i.e. convergence analysis (see [ACL6]), stopping criteria (see [ACL56] and [ACL50]), implementation and performance analysis (see [ACL308]).

We emphasize also on the organization in LAAS-CNRS of the 16th international conference on Parallel Distributed and network-based Processing, PDP, February 13-15, 2008 (see http://www.pdp2008.org/).

Dr Didier El Baz was Chair of the Organizing Committee and Chair of the Program Committee (see [DO12]). This successful conference has received 140 papers from 24 countries; this represents a 40% increase of papers as compared with previous years; 40% of papers have been selected as regular papers for publication in the Proceedings published by IEEE CPS and 120 researchers attended the conference.

We have also organized a CUDA tutorial with the help of NVIDIA USA and NVIDIA France (see http://www.pdp2008.org/nvidia.html). This tutorial was also very successful with 40 participants from 8 countries.

Dr Didier El Baz was also Chair of the Program Committee of PDP Weimar, Germany, February 18-20, 2009 (see [DO20] and http://www.pdp2009.org/).

Dr Didier El Baz has also organized Special Session: Integer Programming and Industrial Applications at international conference on Computers and Industrial Engineering, CIE39, Troyes, July 6-8, 2009 (see http://www.utt.fr/cie39/SSSS.htm).

Significant projects and collaborations

1) Significant projects

DCA is the global coordinator of project ANR CIP, ANR-07-CIST-011, that deals with high performance peer to peer distributed computing (see http://www.laas.fr/CIS-CIP/). CIP was funded by ANR in 2007 in the framework of the so-called “Calcul Intensif et Simulation”, CIS, Call for Proposal. The partners are: LAAS-CNRS, IRIT-ENSEEIHT, University of Picardie, University of Franche-Comté and Euromedtextile. The goal of this project is to develop a decentralized environment for high performance peer to peer distributed computing that allows efficient direct communication between peers (see [INV49]). This is an example on how we plan to address the easiness of use grand challenge. The environment is based on a self adaptive communication protocol whereby choice of communication depends on the combination of application level decisions like choice of computation scheme, e.g. synchronous versus asynchronous schemes and network level elements of context like topology, e.g. adherence to a given cluster or nearness of peers. The self adaptive protocol makes wide use of micro protocols (see Figure 3).

Market solutions that have followed the precursor SETI program are rather grid or global computing solutions (that are not totally decentralized) than actual peer to peer solutions. Peer to peer networks seem to be an attractive approach for massively distributed computing. As a consequence, there is a real economic stake at developing decentralized environments for peer to peer distributed computing applications. Peer to peer computing that permits one to use large sets of machines which are often unemployed appears also as a pragmatic “Green Computing” solution.

Aside from the development of a self adaptive communication protocol and decentralized environment for peer to peer computing, the project deals also with large scale peer to peer distributed computing simulation and performance prediction, it develops also a series of distributed solvers in financial mathematics, process engineering and logistics.

Figure 3: architecture of the self adaptive protocol

DCA is also partner of ANR ROBO 06 project Smart Surface that deals with the design of a distributed system for micro entities moving and positioning (see http://www.smartsurface.cnrs.fr/). Smart Surface aims at designing a surface based on MEMS
technologies. The partners are Femto-ST (coordinator), InESS, LAAS-CNRS, LIFC and LIMMS. The main originality of the smart surface project is to propose a completely distributed solution whereby micro parts are moved via the collaboration of a set of entities; each entity integrates sensors, a processor and actuators (see Figure 4). The Smart Surface project is an example of high performance distributed computing on a dedicated architecture. DCA designs distributed algorithms for state acquisition and pattern recognition. DCA deals also with convergence study, stopping criteria definition, algorithm validation and performance analysis.

Figure 4: the distributed smart surface

2) Collaborations
Among the collaborations during the period, we would like to quote the long term collaboration with Jean-Claude Miellou from LCS Besançon and Pierre Spiteri from IRIT-ENSEEIHT. This collaboration on asynchronous iterations with order intervals is detailed in the Section “Highlights and Major Achievements” of this report. One can quote also the collaboration with Andreas Frommer from University of Wuppertal, Germany on convergence of asynchronous iterations. DCA is also in contact with Professor Vaidy Sunderam from Emory University, Atlanta; the topic concerns the definition of distributed computing paradigms inside Unibus. DCA works also on peer to peer distributed computing with NICITA, Evelyhe, Australia.
One can also quote the collaboration with Julien Bourgeois from LIFC Montbéliard on ANR projects Smart Surface and CIP. This collaboration was also essential to the success of the international conference Parallel Distributed and network-based Processing, PDP 2008, Toulouse (see [DO12]).
The team DCA has also cofounded working group Knapsack and Optimization, KSO, of GDR RO with Professor Mhand Hifi from University of Picardie (see http://www.laas.fr/KSO/). One can finally quote the collaboration with Gérard Plateau from Paris 13 University on constraint rotation methods (see: Highlights and Major Achievements Section).

Main References