IoT and the Need for High Performance Computing

Didier El Baz

IoT and the Need for High Performance Computing

Didier El Baz
CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
Université de Toulouse, LAAS, F-31400 Toulouse, France
elbaz@laas.fr

***Abstract*— The connection between Internet of Things (IoT) and High Performance Computing (HPC) is investigated in this keynote presentation. New paradigms and devices for HPC are presented. Several examples related to smart building management, smart logistics and smart manufacturing leading to difficult combinatorial optimization problems are detailed.**

Keywords—Internet of Things; smart building; smart logistics; smart conveyors; combinatorial optimization; High Performance Computing; GPU computing; Peer-to-Peer computing; distributed computing

I. INTRODUCTION

Internet of Things (IoT) [1] is commonly viewed as a network of items embedded with sensors that are connected to the Internet. The items may have embedded intelligence; the intelligence can also be distributed or hosted like in a cloud. A broader view of IoT is a networked connection of everyday objects including computers, sensors, humans, etc. (see [2]).

The number of devices connected via the Internet rapidly grows. One commonly estimates that around six billions humans and almost the same number of devices will be connected to the Internet by 2015. Nevertheless, IoT and the associated concept of smart world give rise to many complex optimization problems. The topic related to the efficient use of embedded, distributed or hosted intelligence in IoT is fundamental in order to address the smart world challenges.

In this keynote presentation, we concentrate on the combination of IoT and High Performance Computing (HPC) or high speed computing. We present the challenges in IoT / HPC. We give several examples of smart world applications, i.e., smart building management, smart logistics and smart manufacturing that lead to difficult combinatorial optimization problems.

Section II presents new trends in HPC including new paradigms and devices. Section III deals Smart World examples where the combination of IoT and HPC is particularly critical. Conclusions are presented in Section IV.

II. NEW TRENDS IN HPC

Recent advances in microprocessors architectures, e.g., the generalization of the concept of parallelism and advances in high bandwidth networks permit one to consider new solutions for HPC like, Graphics Processing Unit (GPU) computing, Many Integrated Core (MIC) computing and more generally heterogeneous computing, Peer-to-Peer (P2P) computing, Cloud computing (or mixed Volunteer / Cloud computing) and Grid Computing.

A. GPU computing

GPUs [2], [3] are highly parallel, multithreaded, many-core architectures. They are better known for image processing. Nevertheless, NVIDIA introduced in 2006 Compute Unified Device Architecture (CUDA), a parallel programming platform and technology that enables users to use GPU accelerators in order to address general purpose parallel applications.

As shown in Fig. 1, a parallel code on GPU (the device), is interleaved with a serial code executed on the CPU (the host). The parallel threads are grouped into blocks which are organized in a grid. The grid is launched via a single CUDA program, the so-called kernel.

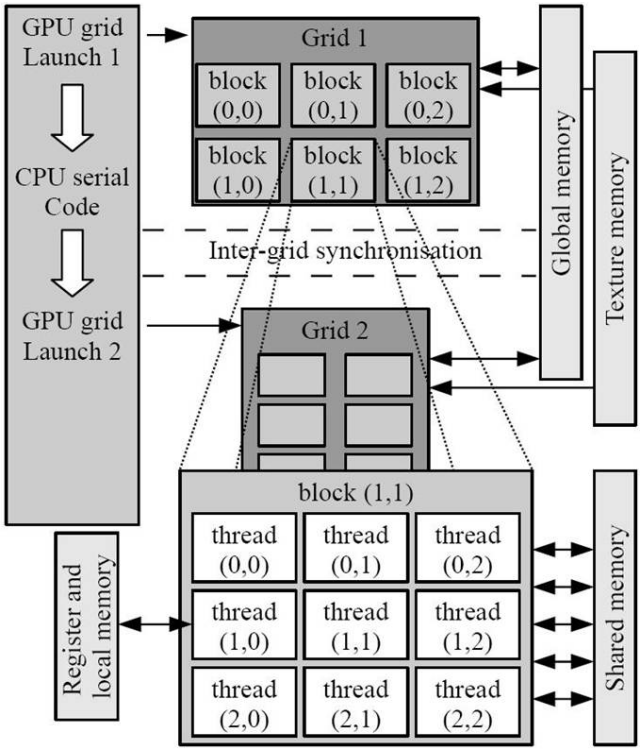

Figure 1. Thread and memory hierarchy in a GPU

978-1-4799-8003-1/14 $31.00 © 2014 IEEE
DOI 10.1109/IIKI.2014.8

GPUs turn out to be widely available, many cores powerful accelerators that are relatively cheap and that require less energy than other computing devices. This last advantage is particularly important for IoT applications. Some GPUs like the brand new Tesla K40 have thousands of CUDA cores and memory bandwidth of 288 Gbytes/s. GPUs can be combined with CPUs in order to build efficient heterogeneous computing platforms. GPUs have been applied with success to many domains in science and engineering including signal processing, linear algebra and numerical simulation [4]. Sometimes dramatic speedups are observed. Complex optimization problems have also been solved via GPUs (see [5], [6], [7], [8] and [9]).

B. MIC

Intel has released in 2013 the Xeon Phi, a Many Integrated Core (MIC) coprocessor for HPC. The coprocessor is composed of up to 61, x86 processor cores, interconnected by a high speed bidirectional ring (see Fig. 2). The architecture of a core is based on the Pentium architecture. Each core can hold four hardware threads (two per clock cycle and per ring's direction).

The Xeon Phi is connected to the CPU via the PCIe connector. The memory controllers and the PCIe client logic provide a direct interface to the GDDR5 memory on the coprocessor and the PCIe bus, respectively. The design of the coprocessor permits one to run existing applications parallelized via OpenMP or MPI; the MIC can be used either in offload mode or native mode (see [10] and [11]). The peak double precision floating point performance of the MIC is 1.2 Tflops which makes it a powerful computing accelerator (see [4]).

C. Peer-to-peer computing

The P2P concept that is very popular for video and file sharing applications has started to be applied with success to HPC applications. Bag of tasks applications have essentially been considered so far. Nevertheless, P2P has been considered recently for parallel and distributed iterative methods with application to optimization and numerical simulation (see [12], [13] and [14]). The P2PDC environment [14] was originally designed as a decentralized environment for P2P HPC.

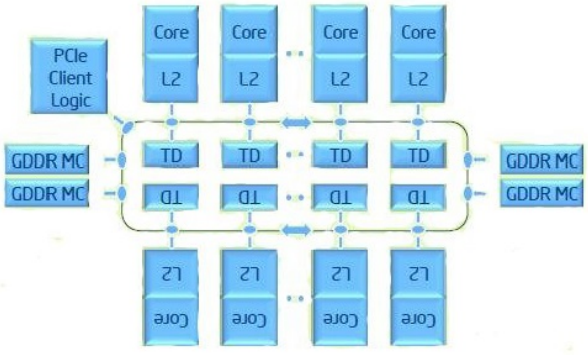

Figure 2. MIC microarchitecture

P2PDC is particularly devoted to task parallel applications. P2PDC is intended in particular to scientists who want to solve optimization problems via distributed iterative methods that lead to frequent direct data exchanges between peers. P2PDC relies on the use of the P2PSAP self-adaptive communication protocol [12] and a reduced set of communication operations (P2Psend, P2Preceive and P2Pwait) in order to facilitate programming. The programmer cares only about the choice of distributed iterative scheme of computation (synchronous or asynchronous) that he wants to be implemented and does not care about the communication mode between any two machines. The programmer has also the possibility to select a hybrid iterative scheme of computation, whereby computations are locally synchronous and asynchronous at the global level.

P2PSAP chooses dynamically the most appropriate communication mode between any two peers according to decision taken at application level like scheme of computation and elements of context like network topology at transport level. In the hybrid case, the communication mode between peers in a group of machines that are close and that present the same characteristics is synchronous and the communication mode between peers in different groups is asynchronous. The decentralized environment P2PDC is based on a hybrid topology manager and a hierarchical task allocation mechanism which make P2PDC scalable.

We note that the P2PSAP communication protocol was designed as an extension of the CTP transport protocol [15] based on the CACTUS framework which uses the concept of microprotocols (see [16]). The P2PSAP communication protocol takes into account Ethernet, Infiniband and Myrinet networks. The reader is referred to [17] for a previous study on P2P computing.

D. Cloud computing

Cloud computing [2] occupies an important position amongst new distributed computing concepts and some high speed computing can reasonably be hosted on the cloud. So far, it is not clear what solution will prevail in the future: either Cloud HPC or HPC in the cloud. Nevertheless, one can merely think about HPC as a service like many others in the cloud.

The cloud has started to be used for HPC applications. There exist now solutions like Amazon Web Service (AWS) cloud including EC2 for HPC, SGI Cyclone cloud and IBM RC2. Hybrid solutions have also been proposed whereby cloud computing is supported by volunteer computing.

III. SMART APPLICATIONS

In this Section, we concentrate on three smart world applications that present strong connections between IoT and HPC problems.

A. Smart building management

The ADREAM building at LAAS-CNRS, Toulouse, France (see Fig. 3) is a typical example of smart building whose management, e.g., air conditioning and light, would require the solution of difficult combinatorial problems, i.e., scheduling problems.

Figure 3. the Adream building at LAAS-CNRS Toulouse

The Adream Building is an energy autonomous building built in the end of 2011 and funded by CNRS, European Community, Regional Council Midi-Pyrénées and Toulouse Métropole. It is a 1700 m² building with 720 m² solar panels on its top and south side (around 150 solar panels).

Figure 4. Inside ADREAM Building

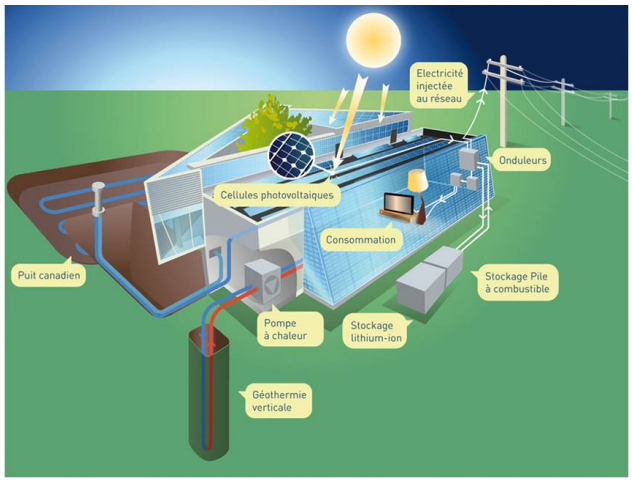

Figure 5. ADREAM building system

The ADREAM building features 6000 sensors of various natures, e.g., temperature sensors, light sensors, motion sensors and cameras. The building features also a mobile grid in order to fix lights, sensors, Motion Capture (MoCap) cameras with IP addresses, etc. (see Fig. 4). Besides solar panels, Adream building features devices like a geothermal exchanger with very low energy and energy storage batteries (see Fig. 5). Peak energy production should be around 100,000 W. Solar energy production is displayed permanently on a monitor at the entrance of the building (see Fig. 6).

Real time management of such a smart building gives raise to many problems like management of data from the many sensors and optimal scheduling of tasks in relationship with heating / air conditioning and light management which is a very difficult optimization problem whose solution demand intensive computation [18]. Optimal scheduling of tasks that consume energy like light and air conditioning is a NP-complete problem. HPC solutions can take great benefit of new devices like GPUs that have been reported to reduce dramatically computing time by factors from 50 up to 150 (see [4] - [9], some problems of which are even irregular problems see [5]) and that require less energy. Distributed heterogeneous computing solutions, in particular seem well suited to the nature of this problem. Similarly, new devices like the MIC coprocessor may present interest in order to speed up the solution of these problems.

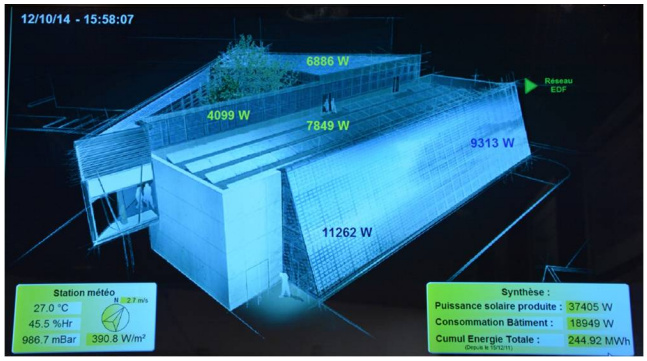

Figure 6. Energy production

Typically, ADREAM Smart Building tends to address the question of adaptivity of machines to complex environments since it deals with a particular type of autonomous systems, i.e., a smart environment that is perceptive to human requirements, that manages its own energy and that is equipped with thousands of temperature, light and motion sensors that inject data in real time.

B. *Smart Logistics*

Logistic applications display also good examples where the combination of IOT and HPC is particularly fruitful.

Logistic operators deliver goods to customers; the optimization of quality of service, e.g., on-time delivery and cost delivery is of major concern in this domain; this necessitates the optimization of truck loading and vehicle routing. The nature of logistic applications is dynamic, e.g., good delivery orders or cancellations may occur at any time; transportation difficulties may also occur at any time. Vicissitudes may be due to vehicle faults, traffic jam or particular weather conditions.

Among the projects related to smart logistic that have started, we can quote the ALMA project (see [19]) designed and developed at LAAS-CNRS (see Fig. 7).

The ALMA project proposes a mobile, real time, IoT-based solution in order to take into account the dynamic nature of logistic problems and to optimize the quality of service. Mobile devices like smart phones are used to report good delivery occurrences and incidents like an engine fault or a traffic jam; they are also used in order to launch computations related to the solution of a resulting routing problem on computing infrastructures in order to cope with incidents in real time. The ALMA project relies on a High Performance Computing (HPC) infrastructure that makes use of clusters, grids and P2P networks via a broker that takes into account computational need and machines availability. The ALMA project relies also on new optimization algorithms for the solution of combined truck loading and vehicle routing problems.

Treatment of vehicle routing problems in conjunction with truck loading has been studied in the literature (see [20] and [21]). The ALMA logistic application concentrates also on dynamic logistic problems whereby dynamicity results from new orders, cancellations as well as traffic incidents that may occur at any time; this leads to extremely difficult problems. Our approach is essentially based on the approximate solution of truck loading problems via strip generation and beam search (see [22] and [23]); vehicle routing problems are solved via Ant Colony Optimization (ACO) [24]. The approach is also based on distributed computing. The ALMA logistic application relies on two infrastructures: a communication infrastructure and a HPC infrastructure. Fig. 7 displays the infrastructures of the mobile application ALMA.

1) The communication infrastructure: Goods to be delivered are identified by tags. When a good is delivered, the transporter scans the tag and transmits the information in real time to the logistic centre with a smart phone connected to the Internet via a 3G connection. The mobile application is based on the existing telecommunication infrastructure. Similarly, the transporter informs the centre in real time of traffic incidents, like road closed and traffic jam. In case of problems, e.g. traffic incidents, the proposed initial route may not be valid. The transporter uses the mobile application to send a request for computation of a new route that is transmitted to the broker of the HPC infrastructure.

2) The HPC infrastructure: The broker is designed in order to select a convenient HPC infrastructure among several available parallel or distributed architectures. These architectures may be clusters, grids or P2P networks. For a given instance of vehicle routing problem and a given method, the broker selects also a convenient topology and number of machines.

The main goal of the broker is to select the best computing infrastructure that satisfies the real time constraints of the application. Vehicle routing requests are associated with a deadline for so as to limit vehicle idle time since computation time that is too long leads to a blocking of the logistic application.

Two phases can be considered for brokering: first, the supervision of available resources, e.g. clusters, grids or P2P networks. Secondly, the prediction of computation time for the considered problem and selected method. We note that these steps can be iterated several times in order to improve prediction. Reference is made to [25] and [26] for previous works on performance prediction of HPC applications on distributed computing infrastructures.

The environment for computing is an extension of P2PDC (see [12] and [13]). Reference is also made to [14] for more details and extensions of P2PDC.

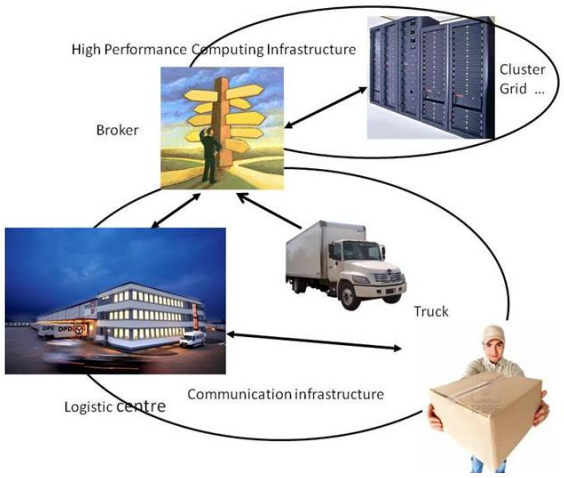

Figure 7. ALMA Infrastructure

C. *Smart Manufacturing*

Reconfigurable conveyors can easily adapt to tasks changes. They require fewer modules than a classic monolithic surface. Reconfigurable conveyors can also cope with faults.

Among a small number of projects related to distributed reconfigurable smart conveyors, the Smart Blocks project aims at designing a centimeter scale self reconfigurable MEMS-based modular surface for safe and fast conveying of fragile micro parts. The Smart Blocks project aims at tackling all related problems so as to increase the efficiency of future production lines. We note that MEMS-based devices with embedded intelligence, also referred to as distributed intelligent MEMS have great potentials on many fields and more particularly for manipulating micro parts in many industries like semiconductor industry and micromechanics (see [27] and [28]).

The centimeter scale modular surface considered in the Smart Block project is composed of few dozens of blocks. A 2D pneumatic MEMS actuator array is embedded on the top of each block in order to move parts. Electro-permanent magnet-based actuators for block motion and sensors are also embedded on each side of a block (see Fig. 8). These features are used to detect neighboring blocks and to move blocks accordingly. Finally processing unit and communications ports are embedded in each block (see Fig. 8). As a consequence, block motion relies on contacts with other blocks and these contacts can occur only on each lateral side of a block, not on the top, nor on the bottom of the block. The reader is referred to [29] for a presentation of the Smart Blocks project. The Smart Block project is a sequel to the Smart Surface project (see [30]) that dealt with a MEMS-based monolithic conveyor which consisted of a distributed array of sensors and air-jet actuators.

The Smart-Blocks project is typical of smart objects with embedded and distributed intelligence that must react very fast in order to reconfigure themselves quickly, i.e. in a high speed distributed context.

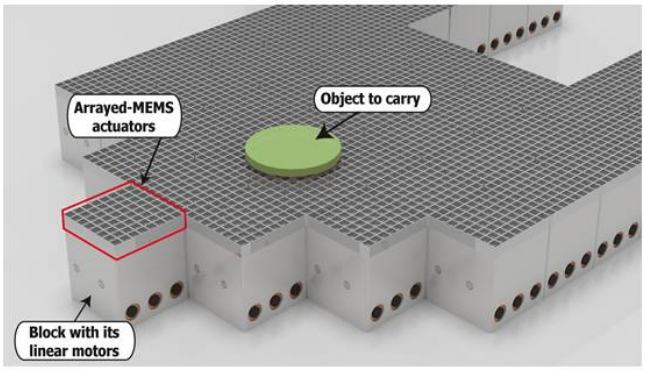

Figure 8. The Smart-Blocks conveyor

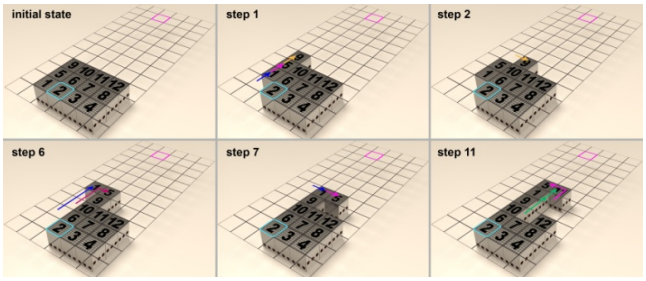

Figure 9. Smart Blocks reconfiguration steps (beginning)

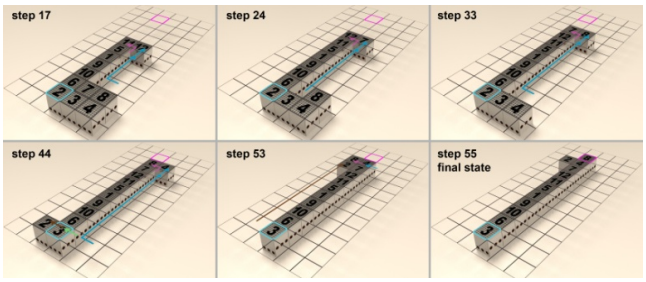

Figure 10. Smart Blocks reconfiguration steps (end)

Blocks cooperate to optimally build the shortest path between the input of parts and their output. A discrete trajectory optimization problem is solved via a distributed algorithm so as to reconfigure the modular surface. In particular, a distributed election of the block that can reach a given position on the surface with a minimum hop count is made; this block raises the next iteration before moving to its final position. The distributed solution is scalable, flexible and optimal. This permits one in particular to quickly set up a modular conveyor with optimal distance between input and output. The shortest path between input and output is computed via a strategy based on minimum hop count which minimizes also the number of block moves in order to build the shortest path. This approach based on distributed asynchronous iterative elections is scalable. The reader is referred to [29] for more details on the distributed election algorithm.

We note that connected blocks with embedded intelligence, i.e. smart blocks, can elegantly morph into a reconfigurable distributed system. This system can also be connected to an Internet Cloud in order to deliver statistics on faulty blocks and number of items correctly conveyed.

IV. Conclusions and Perspectives

In this keynote presentation, we have investigated the link between IoT and HPC. It is particularly important to study this link since many IoT applications feature the inherent complexity of the physical world which leads to HPC problems and in particular to difficult combinatorial optimization problems.

We have considered items with embedded, distributed or hosted intelligence. We have concentrated on three main applications related to the smart world concept, i.e., smart building management, smart logistics and smart

manufacturing. We have seen that IoT applications can take great benefit of new parallel devices like GPU and MIC, and emerging distributed computing concepts like Cloud, volunteer and P2P computing that revisit the field of HPC.

ACKNOWLEDGMENT

Part of the research presented here has been made possible with the support of NVIDIA Corporation with the donation of a Tesla K40 GPU. The author gratefully acknowledges support of the Grid5000 experimental testbed, developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities.

REFERENCES

[1] Huansheng Ning, Unit and Ubiquitous Internet of Things, CRC Press, 2013.

[2] K. Hwang, G. Fox, J. Dongarra, Distributed and Cloud Computing, from Parallel Processing to the Internet of Things, Morgan Kaufmann, 2012.

[3] Randima Fernando, GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics, Addison Wesley Professional, 2004.

[4] Bastien Plazolles, Didier El Baz, Martin Spel and Vincent Rivola, "Comparison between GPU and MIC on balloon envelope drift descent analysis", LAAS report, 2014.

[5] V. Boyer, D. El Baz, "Recent advances on GPU computing in Operations Research", in Proceedings of the 27th IEEE Symposium IPDPSW 2013, Boston USA, 20-24 May 2013, pp. 1778-1787.

[6] V. Boyer, D. El Baz, M. Elkihel, "Solving knapsack problems on GPU", Computers and Operations Research, Vol. 39, N° 1, 2012, pp. 42-47.

[7] M. Lalami, D. El Baz, "GPU implementation of the Branch and bound method for knapsack problems", in Proceedings of the 26th IEEE Symposium IPDPSW 2012 / PCO'12, Shanghai China, 20-25 May 2012, pp. 1763-1771.

[8] M. Lalami, V. Boyer, D. El Baz, "Efficient implementation of the Simplex method on a CPU-GPU system", in Proceedings of the 25th IEEE Symposium IPDPSW 2011 / PCO'11, Anchorage USA, 19-23 May 2011.

[9] M. Lalami, D. El Baz, "Multi GPU implementation of the Simplex algorithm", in Proceedings of the 13th IEEE Conference on High Performance Computing and Communications, Banff Canada, 2-4 September 2011.

[10] J. Jeffers and J. Reinders, Intel Xeon Phi Coprocessor High-Performance, Programming, ser. Morgan Kaufmann. Elsevier Science & Technology Books, 2013.

[11] R. Rahman, Intel Xeon Phi Coprocessor Architecture and Tools: The Guide for Application Developers, 1st ed. Berkeley, CA, USA: Apress, 2013.

[12] D. El Baz, T.T Nguyen, "A self-adaptive communication protocol with application to high performance peer to peer distributed computing", in Proceedings of 18th International Conference on Parallel, Distributed and network based Processing, Pisa, Italy, 17-19 February 2010, pp. 323-333.

[13] T.T. Nguyen, D. El Baz, P. Spiteri, G. Jourjon, M. Chau, "High performance peer-to-peer distributed computing with application to obstacle problem", in Proceedings of HOTP2P in conjunction with the Symposium IEEE IPDPS 2010, Atlanta, USA, 19-23 April 2010.

[14] T. Garcia, M. Chau, T. T. Nguyen, D. El Baz, P. Spiteri, "Asynchronous peer-to-peer distributed computing for financial applications", in Proceedings of the 25th IEEE Symposium IPDPSW 2011 / PDSEC 2011, Anchorage USA, 19-23 May 2011.

[15] G. Wong, M. Hiltunen, and R. Schlichting, "A Configurable and Extensible Transport Protocol," in Proceedings of IEEE INFOCOM, 2001, pp. 319–328.

[16] M. A. Hiltunen, "The CACTUS approach to building configurable middleware services," in Proc. of DSMGC 2000.

[17] G. Jourjon, D. El Baz, "Some solutions for peer to peer global computing", in Proceedings of the 13-th conference on Parallel, Distributed and network-based Processing, PDP 2005, Lugano, Suisse, February 9-11, 2005, IEEE CPS, pp. 49-58.

[18] C. Artigues, S. Demassey, and E. Néron, editors. Resource Constrained Project Scheduling: Models, Algorithms, Extensions and Applications. Control Systems, Robotics and Manufacturing Series. ISTE-WILEY, 2008.

[19] D. El Baz, J. Bourgeois, T. Saadi, A. Bassi, ALMA, "A logistic Mobile Application based on the Internet of Things", 2013 IEEE International Conference on Internet of Things, pp. 707-715, 2013.

[20] M. Gendreau, M. Iori, G. Laporte, S. Martello, "A Tabu search heuristic for the vehicle routing problem with two-dimensional loading constraint", Management Science, Vol. 40, N° 10, 1994, pp. 1276-1290.

[21] K. Ganesh, T. T. Narendran, "CLOVES: a cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up", European Journal of Operational Research, Vol. 178, 2007, pp. 699-717.

[22] M. Hifi, R. MHallah T. Saadi, "Algorithms for the constrained two-staged two-dimensional cutting problem", INFORMS Journal on Computing, Vol.20, 2008, pp.212-221.

[23] M. Hifi, T. Saadi, "A cooperative algorithm for constrained two-staged two-dimensional cutting problems", International Journal of Operational Research, Vol. 9, No.1, 2010, pp. 104-124.

[24] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.

[25] B. Cornea, J. Bourgeois, T. T. Nguyen, D. El Baz, "Performance prediction in a decentralized environment for peer to peer computing", Proc. 25 th IEEE International Parallel & Distributed Processing Symposium and Workshops (IPDPSW), Anchorage USA, May 2011, pp. 1613-1621.

[26] B. Cornea, J. Bourgeois, T.T. Nguyen, D. El Baz, "Scalable performance predictions of distributed peer-to-peer application", Proc. 14th IEEE International Conference on High Performance Computing and Communication, Liverpool U.K, 2012, pp. 193-201.

[27] D. Biegelsen et al. "Airjet paper mover," in SPIE International Symposium on Micromachining and Microfabrication, 4176-11, Sept. 2000.

[28] Y. Fukuta, Y. Chapuis, Y. Mita, H. Fujita, "Design fabrication and control of MEMS-based actuator arrays for air-flow distributed micromanipulation," IEEE Journal of Micro-Electro-Mechanical Systems, Vol. 15 (4), 2006, pp. 912-926.

[29] D. El Baz, B. Piranda, J. Bourgeois, "A distributed algorithm for a reconfigurable modular surface", in Proceedings of the 28th IEEE Symposium IPDPSW 2014 / PCO 2014, Phoenix USA, 19-23 May 2014.

[30] D. El Baz, V. Boyer, J. Bourgeois, E. Dedu, K. Boutoustous, "Distributed part differentiation in a smart surface", Mechatronics, Vol. 22, Issue 5, 2012, pp. 522-530.