IoT and the Need for High Performance Computing

Didier El Baz
CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ;
elbaz@laas.fr

***Abstract*—The connection between Internet of Things (IoT) and High Performance Computing (HPC) is presented in details. New devices and paradigms for HPC are presented. Several examples related to smart buiding management, smart logistics and smart manufacturing ar e detailed.**

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

I. INTRODUCTION

Internet of Things (IoT) 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.

In this keynote presentation, I concentrate on the combination of IoT and High Performance Computing (HPC) or high speed computing. I present the challenges in IoT / HPC. And give several examples related to smart world applications: smart building management, smart logistics and smart manufacturing.

Section 2 presents new trends in HPC. Section 3 deals Smart World examples were the combination of IoT and HPC is particularly critical. Conclusions are presented in Section 4.

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, GPU computing, Many Integrated Core (MIC) computing, Peer-to-Peer (P2P) computing, Cloud computing (or mixed Volunteer / Cloud computing) and Grid Computing.

A. GPU computing

GPUs are highly parallel, multithreaded, many-core architectures. They are better known for image processing. Nevertheless, NVIDIA introduced in 2006 CUDA (Compute Unified Device Architecture), a technologie that enables users to use a GPU card to address parallel applications.

As shown in Fig. 1, a parallel code on GPU (hereafter named the device), is interleaved with a serial code executed on the CPU (hereafter named the host). The parallel threads are grouped into blocks which are organised in a grid. The grid is launched via a single CUDA program, the so-called kernel. The GPU implementation of the code can be performed via CUDA a parallel programming plarform.

(see Fig. 1).

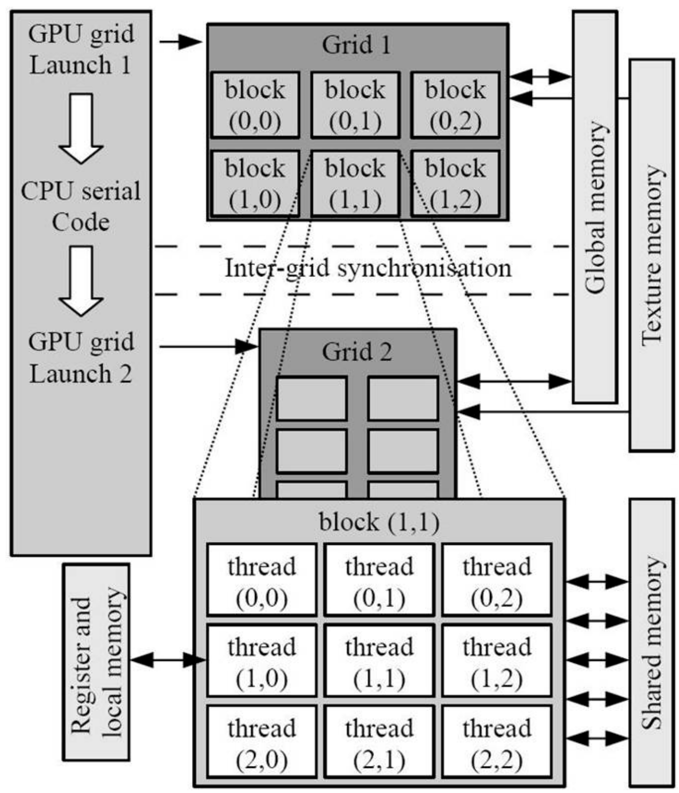

Figure 1. Thread and memory hierarchy in GPU

B. MIC

In 2013, Intel released the Many Integrated Core (MIC) coprocessor : the Intel Xeon Phi. The coprocessor is composed of up to 61, x86 processor cores, interconnected by a highspeed bidirectional ring (see Fig. 2). The architecture of a core is based on 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 [7].

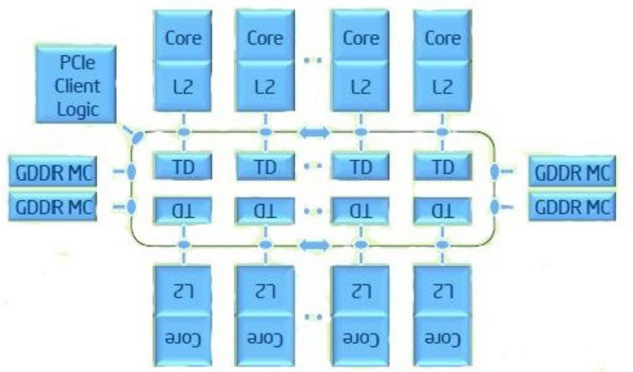

Figure 2. MIC microarchitecture

C. Peer-to-peer computing

(see Fig. 3).

Figure 3. Communication network of the smart surface

We recall that P2PDC was originally designed as a decentralized environment for peer-to-peer High Performance Computing; P2PDC is particularly devoted to task parallel applications. P2PDC is intended in particular to scientists who want to solve numerical simulation 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 [17] (see Fig. 2) 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 first as an extension of the CTP transport protocol [18] based on the CACTUS framework which uses the concept of microprotocols (see [19]). The CTP protocol includes a wide range of micro-protocols including a small set of basic micro-protocols like Transport Driver, Fixed Size or Resize and Checksum that are needed in every configuration and a set of micro-protocols implementing various transport properties like acknowledgements, retransmissions, error correction and congestion control. The P2PSAP communication protocol takes into account Ethernet and Infiniband clusters. Reference is also made to [20] for details on peer-to-peer computing.

D. Cloud computing

III. Smart applications

In this Section, we present three smart world applications with strong connections between Iot and HPC problems.

E. Smart building management

The ADREAM building at LAAS-CNRS, Toulouse, France (see Figure 1) is a typical example of smart building whose management, e;g. air conditioning, light, requires the solution of difficult combinatorial problems.

Figure 1. the Adream building at LAAS-CNRS Toulouse

Adream Building at LAAS-CNRS is an energy autonomous building buildt 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). The

ADREAM building features 6000 sensors of various nature, e.g., temperature sensors, light sensors, motion sensors and cameras. The building features a mobile grid to fix Motion Capture (MoCap) Cameras with IP adresses, light etc. (see Fig. 2).

Figure 2. Inside ADREAM Building

Besides solar panels, Adream Building features also devices like a geothermal exchanger with very low energy and energy storage batteries (see Fig. 3).

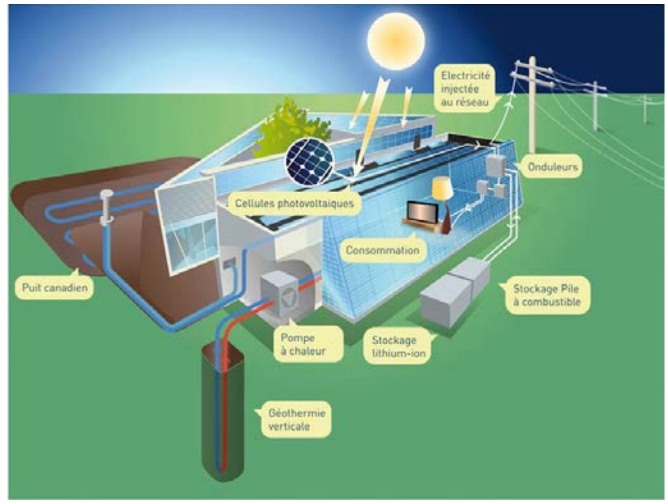

Figure 3. ADREAM Building principle

Peak energy production is given for 100,000 W. Solar energy production is displayed permanently on a monitor at the entrance of the building (see Fig. 4).

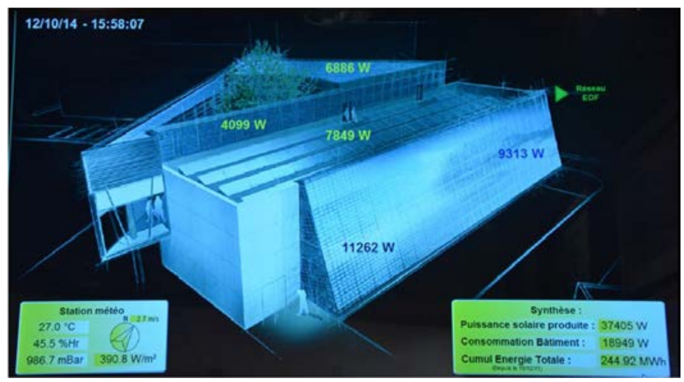

Figure 4. Solar energy production

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. Optimal scheduling of tasks that consume energy like light and air conditioning in a Smart Building 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), some problems of which are even irregular problems (see) and that have low energy consumption. Distributed heterogeneous computing solutions, in particular seem well suited to the nature of this problem. Similarly, new devices like the processor MIC may present interest in order to speed up the solution of these problems. Typically, the problematic of the Smart Building ADREAM address the question of adapvity of machines to complex environments since it deals with a particular type of autonomous systems, i.e., smart environments that are perceptive to human requirements, that manage their own energy and that are equipped with thousands of temperature and light sensors that inject data in real time.

F. Smart Logistics

Logistic applications display also good examples where the combination of IOT and HPC is particularly fruitful and is closely entangled to a smart cities smart world vision.

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 many projects related to smart logistic that have started, we can quote the ALMA project (see) designed and developed at LAAS-CNRS (see Fig. 5).

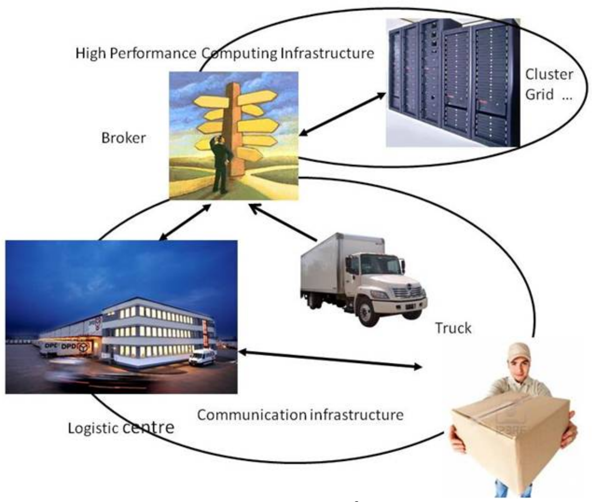

Figure 5. ALMA Infrastructure

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 new High Performance Computing (HPC) infrastructure that makes use of clusters, grids and peer-to-peer 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 [3] to [6]). 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 [7] to [9]); vehicle routing problems are solved via Ant Colony Optimization (ACO) [10]. The approach relies on parallel or distributed computing via several types of architectures, e.g., clusters, grids or peer-to-peer infrastructures since those problems are difficult to solve.

The ALMA logistic application relies on two infrastructures: a communication infrastructure and a HPC infrastructure. Fig. 1 displays the infrastructures of the mobile application ALMA.

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 also the mobile application to ask for a new route. The request for a new route is transmitted to the broker of the HPC infrastructure.

The HPC infrastructure

1) The broker

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 peer-to-peer networks. For a given instance of vehicle routing problem and a given method, the broker selects also a convenient topology and a number of machines. This represents an evolution in comparison with the approach presented in [11].

The main goal of the broker is to select the best computing infrastructure that satisfies the real time constraints of the application. Vehicle routing solution requests are associated with a deadline for result reception so as to limit vehicle idle time since computation time that are too long lead to a blocking of the logistic application. We note that if the infrastructure selection is not convenient, then a suboptimal solution far from the optimum will not be appropriate. The broker chooses a HPC infrastructure that satisfies the time constraint of the application.

Two main phases can be considered for brokering: first, the supervision of available resources, e.g. clusters, grids or peer-to-peer 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 [12] to [14] for previous work on performance prediction of HPC applications on distributed computing infrastructures.

2) The environment for computing

The environment for computing is an extension of P2PDC (see [11]). Reference is also made to [15] and [16] for more details and extensions of P2PDC.

G. 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 limited number of projects related to distributed reconfigurable smart conveyors, the Smart Blocks project [10] 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 [6], [7] have great potentials on many fields and more particularly for manipulating micro parts in many industries like semiconductor industry and micromechanics (see [8], [9]). The centimeter scale modular surface studied 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 (see [2] and [11]). Electro-permanent magnet-based actuators for block motion and sensors are also embedded on each side of a block (see Fig. 1). 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. 6). 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 the bottom of the block. The reader is referred to [] for a complete and detailed presentation of the Smart Blocks project. The Smart Block project is a sequel to the Smart Surface project (see [12] and [13]) 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.

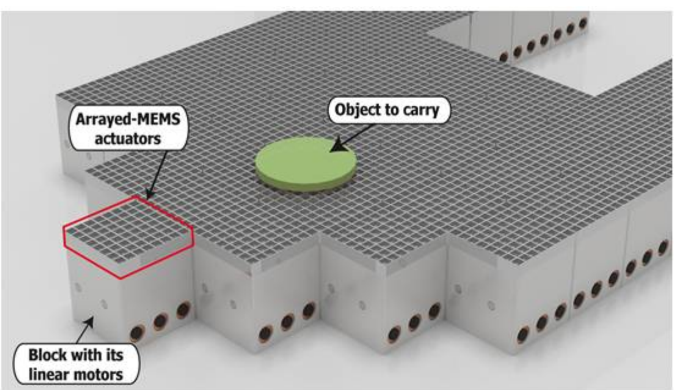

Figure 6. The Smart-Blocks conveyor

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 mimimal 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.

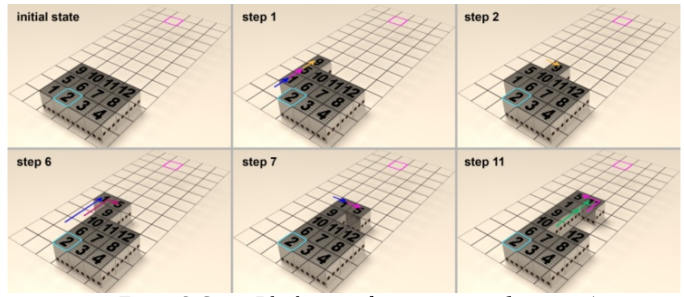

Figure 6. Smart Blocks reconfiguration steps (beginning)

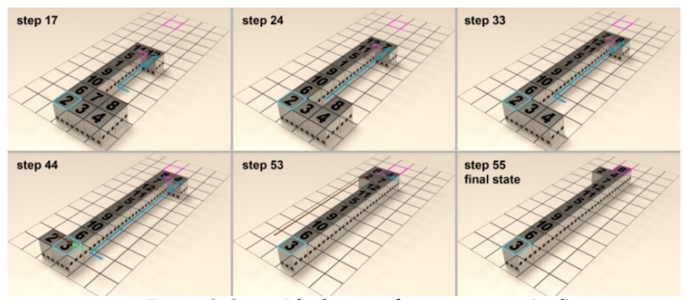

Figure 6. Smart Blocks reconfiguration steps (end)

IV. Conclusions and Perspectives

Beyond IoT there is the inherent complexity of the physical world that leads to many HPC problems. IoT applications can take benefit of new concepts like GPU, MIC, Cloud and P2P computing that revisit distributed and parallel computing.

Acknowledgments

Part of the research works presented in this keynote presentation has been made possible with the support of NVIDIA Corporation with the donation of a Tesla K40 GPU and several Tesla C2050 computing accelerators. The author also gratefully acknowledge support of the Grid5000 experimental testbed, developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

References

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

[2] 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.

[3] K. Boutoustous, E. Dedu, J. Bourgeois, “A framework to calibrate a MEMS sensor network,” Proc. of the International Conference on Ubiquitous Intelligence and Computing, Brisbane Australia, July 2009, Lecture Notes in Computer Science, Springer, Berlin, 2009, pp. 136-149.

[4] Smart Suface, ANR PSIRob 2006 project, http://www.smartsurface.cnrs.fr/

[5] Smart Surface at LAAS-CNRS: http://www.laas.fr/SMART SURFACE/

[6] F. Robert, Discrete Iterations, a Metric Study, Springer Series in Computational Mathematics Springer Verlag, Berlin, 1986.

[7] D. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation, Numerical Methods, Prentice Hall, Englewood Cliffs 1989.

[8] A. Radid, Itérations Booléennes et sur des ensembles de cardinal fini, Ph. D. thesis of the University of Franche-Comté, 2000.

[9] E. W. Dijkstra and C. S. Scholten, “Termination detection for diffusing computation,” Information Processing Letters, Vol. 11, 1980, pp. 1-4.

[10] D.Bertsekas, J. Tsitsiklis, “Parallel and distributed iterative algorithms: a selective survey,” Automatica, Vol. 25, 1991, pp. 3-21.

[11] D. El Baz, “An efficient termination method for asynchronous iterative algorithms on message passing architecture,” Proc. of the International Conference on Parallel and Distributed Computing Systems, Dijon, Vol. 1, 1996, pp. 1-7.

[12] D. El Baz, “A method of terminating asynchronous iterative algorithms on message passing systems,” Parallel Algorithms and Application, Vol. 9, 1996, pp. 153-158.

[13] K. Boutoustous, E. Dedu, J. Bourgeois, “An exhaustive comparison framework for distributed shape differentiation in a MEMS sensor actuator array,” Proc. International Symposium on Parallel and Distributed Computing (ISPDC), IEEE Computer Society Press, Kraków, Poland, 2008, pp. 429-433.

[14] H. Ishida et al. “Recognition of low resolution characters by a generative learning method,” Proc. of the 1st International Workshop on Camera-Based Document Analysis and Recognition, pp. 45-51.

[15] N. Le Fort-Piat et al. “Smart Surface based on autonomous distributed micro robotic systems for robust and adaptive micromanipulation,” ANR Smart Surface Project Proposal, 2006.

[16] K. F. Bohringer, V. Bhatt, B. R. Donald and K. Y. Goldberg. “Algorithms for sensorless manipulation using a vibrating surface,” Algorithmica, 26(3-4), 2000, pp. 389–429.

[17] H. Fujita. “Group work of microactuators,” In International Advanced Robot Program Workshop on Micromachine Technologies and Systems, Tokyo, Japan, October 1993, pp. 24–31.

[18] L. Matignon, G. Laurent, N. Le Fort-Piat, “Design of semi-decentralized control laws for distributed-air-jet micromanipulators by reinforcement learning,” IROS, 2009, pp. 3277-3283.

[19] K.S.J. Pister, R. Fearing, and R. T. Howe. “A planar air levitated electrostatic actuator system,” In IEEE Micro Electro Mechanical Systems. An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots, 1990, pp. 61–71.

[20] Y.-A. Chapuis, L. Zhou, Y. Fukuta, Y. Mita, and H. Fujita, “FPGA-based decentralized control of arrayed MEMS for micro robotic application,” IEEE Transactions on Industrial Electronics, 54 (4), August 2007, pp.1926-1936.

[21] Y.-A. Chapuis, L. Zhou, H. Fujita, and Y. Hervé, “Multi-domain simulation using VHDL-AMS for distributed MEMS in functional environment: Case of a 2D air-jet micromanipulator,” Sensors and Actuators A: Physical, 148, November 2008, pp. 224–238.

[22] S. Tabbone, L. Wendling, and J.-P. Salmon. “A new shape descriptor defined on the Radon transform,” Computer Vision and Image Understanding, 102 (1), April 2006, pp.42-51.

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

[24] 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.

[25] K. Boutoustous, E. Dedu, J. Bourgeois, “A framework to calibrate a MEMS sensor network,” Proc. of the International Conference on Ubiquitous Intelligence and Computing, Brisbane Australia, July 2009, Lecture Notes in Computer Science, Springer, Berlin, 2009, pp. 136-149.

[26] Smart Suface, ANR PSIRob 2006 project, http://www.smartsurface.cnrs.fr/

[27] Smart Surface at LAAS-CNRS: http://www.laas.fr/SMART SURFACE/

[28] F. Robert, Discrete Iterations, a Metric Study, Springer Series in Computational Mathematics Springer Verlag, Berlin, 1986.

[29] D. Bertsekas, J. Tsitsiklis, Parallel and Distributed Computation, Numerical Methods, Prentice Hall, Englewood Cliffs 1989.

[30] A. Radid, Itérations Booléennes et sur des ensembles de cardinal fini, Ph. D. thesis of the University of Franche-Comté, 2000.

[31] E. W. Dijkstra and C. S. Scholten, “Termination detection for diffusing computation,” Information Processing Letters, Vol. 11, 1980, pp. 1-4.

[32] D.Bertsekas, J. Tsitsiklis, “Parallel and distributed iterative algorithms: a selective survey,” Automatica, Vol. 25, 1991, pp. 3-21.

[33] D. El Baz, “An efficient termination method for asynchronous iterative algorithms on message passing architecture,” Proc. of the International Conference on Parallel and Distributed Computing Systems, Dijon, Vol. 1, 1996, pp. 1-7.

[34] D. El Baz, “A method of terminating asynchronous iterative algorithms on message passing systems,” Parallel Algorithms and Application, Vol. 9, 1996, pp. 153-158.

[35] K. Boutoustous, E. Dedu, J. Bourgeois, “An exhaustive comparison framework for distributed shape differentiation in a MEMS sensor actuator array,” Proc. International Symposium on Parallel and Distributed Computing (ISPDC), IEEE Computer Society Press, Kraków, Poland, 2008, pp. 429-433.

[36] H. Ishida et al. “Recognition of low resolution characters by a generative learning method,” Proc. of the 1st International Workshop on Camera-Based Document Analysis and Recognition, pp. 45-51.

[37] N. Le Fort-Piat et al. “Smart Surface based on autonomous distributed micro robotic systems for robust and adaptive micromanipulation,” ANR Smart Surface Project Proposal, 2006.

[38] K. F. Bohringer, V. Bhatt, B. R. Donald and K. Y. Goldberg. “Algorithms for sensorless manipulation using a vibrating surface,” Algorithmica, 26(3-4), 2000, pp. 389–429.

[39] H. Fujita. “Group work of microactuators,” In International Advanced Robot Program Workshop on Micromachine Technologies and Systems, Tokyo, Japan, October 1993, pp. 24–31.
[40] L. Matignon, G. Laurent, N. Le Fort-Piat, “Design of semi-decentralized control laws for distributed-air-jet micromanipulators by reinforcement learning,” IROS, 2009, pp. 3277-3283.
[41] K.S.J. Pister, R. Fearing, and R. T. Howe. “A planar air levitated electrostatic actuator system,” In IEEE Micro Electro Mechanical Systems. An Investigation of Micro Structures, Sensors, Actuators, Machines and Robots, 1990, pp. 61–71.
[42] Y.-A. Chapuis, L. Zhou, Y. Fukuta, Y. Mita, and H. Fujita, “FPGA-based decentralized control of arrayed MEMS for micro robotic application,” IEEE Transactions on Industrial Electronics, 54 (4), August 2007, pp.1926-1936.
[43] Y.-A. Chapuis, L. Zhou, H. Fujita, and Y. Hervé, “Multi-domain simulation using VHDL-AMS for distributed MEMS in functional environment: Case of a 2D air-jet micromanipulator,” Sensors and Actuators A: Physical, 148, November 2008, pp. 224–238.
[44] S. Tabbone, L. Wendling, and J.-P. Salmon. “A new shape descriptor defined on the Radon transform,” Computer Vision and Image Understanding, 102 (1), April 2006, pp.42-51.

[45] [1] K. Hwang, G. Fox, J. Dongarra, Distributed and Cloud Computing, from Parallel Processing to the Internet of Things, Morgan Kaufmann, 2012.
[46] [2] NVIDIA, TESLA C2050 and TESLA C2070 Computing processor board. NVIDIA Corporation, 2010.
[47] [3] 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, p. 1778-1787.
[48] [4] V. Boyer, D. El Baz, M. Elkihel, Solving knapsack problems on GPU, Computers and Operations Research, Vol. 39, N° 1, 2012, p. 42-47.
[49] [5] 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, p. 1763-1771.
[50] [6] 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.
[51] [7] T. Rakotoarivelo, M. Ott, I. Seskar, and G. Jourjon, OMF: a control and management framework for networking testbeds," in SOSP Workshop on Real Overlays and Distributed Systems (ROADS '09), Big Sky, USA, Oct. 2009, p. 6.
[52] [8] O. Mehani, G. Jourjon, T. Rakotoarivelo, and M. Ott, An instrumentation framework for the critical task of measurement collection in the future Internet, NICTA, Tech. Rep. 6065, 2012.
[53] [9] G. Jourjon, T. Rakotoarivelo, and M. Ott, A portal to support rigorous experimental methodology in network research, in Proc. of TridentCom, Apr. 2011.
[54]