A N-dimensional Stochastic Control Algorithm for Electricity Asset Management on PC cluster and Blue Gene Supercomputer

Stéphane Vialle, Xavier Warin, Patrick Mercier

A N-dimensional Stochastic Control Algorithm for Electricity Asset Management on PC cluster and Blue Gene Supercomputer

Stéphane Vialle[1], Xavier Warin[2], and Patrick Mercier[1]

[1] SUPELEC, IMS research group, 2 rue Edouard Belin, 57070 Metz, France,
Stephane.Vialle@supelec.fr Patrick.Mercier@supelec.fr
[2] EDF - R&D, OSIRIS group, 1 Avenue Charles de Gaulle, 92141 Clamart, France,
Xavier.Warin@edf.fr

Abstract. Management of French electricity production to control cost while satisfying demand, leads to solve a stochastic optimization problem where the main sources of uncertainty are the demand load, the electricity and fuel market prices, the hydraulicity, and the availability of the thermal production assets. A stochastic dynamic programming method is an interesting solution, but is both CPU and memory consuming. It requires parallelization to achieve speedup and size up, and to deal with a big number of stocks (N) and a big number of uncertainty factors.
This paper introduces a distribution of a N-dimension stochastic dynamic programming application, on PC clusters and IBM Blue Gene/L supercomputer. It has needed to parallelize input and output file accesses from thousands of processors, to load balance a N-dimension cube of data and computation evolving at each time step, and to compute Monte-Carlo simulations requiring data spread in many separate files managed by different processors. Finally, a successful experiment of a 7-stock problem using up to 8192 processors validates this distribution strategy.

1 Main objectives and challenges

EDF (*Electricité de France*) is an electricity producer that needs to manage its production assets in order to satisfy the consumer demand. It leads to a stochastic optimization problem where the main sources of uncertainty are the demand load, the electricity and fuel market prices, the hydraulicity, and the availability of the thermal production assets. The goal of the optimization is to save energetic resources and money by managing the production cost while satisfying demand. The main levers of this optimization are the commands that can be used to manage the different stocks involved in the problem and the thermal production levels. The necessity to deal with a lot of stocks (mainly water stocks used by hydraulic power plant, and fuel stocks) gives a very big optimization problem. Due to the non convexity of the problem, the stochastic dynamic programming method is a method of choice used to find a solution. Unfortunately, it is unable to deal with a big number of stocks and a big number of uncertainty factors. The goal of this work is to be able to deal with at least three or four uncertainty factors, and at least six or seven stocks in optimization while being able to efficiently use in simulation the commands calculated.

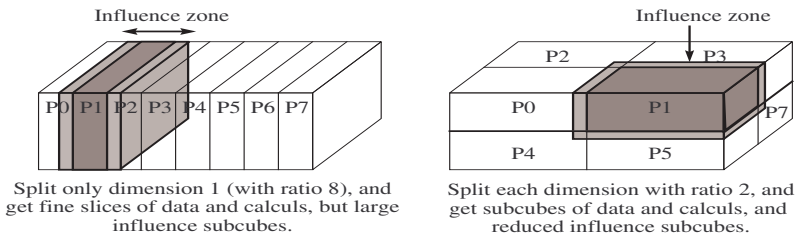

Fig. 1. Small example of a *cubic* split of the N-Cube of data and computation

The entire application is composed of three different parts and parallel algorithms. The first part consists in *reading many input files* and to store data in each processor memory, using an efficient parallel file system (like on Blue Gene supercomputer) or a classical file system (like NFS on our PC cluster). The second part of the application is the *optimization* part that computes the optimal decision to be taken depending on the state (levels of stocks and realizations of aleas) by a backward time recursion. The amount of data and computation can change at each time step of this recursion, so it is mandatory to redistribute the new computations and their required input data at the beginning of each time step to ensure a good load balancing. Moreover, processors generate a lot of intermediate results (the so called Bellman values) to store concurrently on disks at each step. The third part is devoted to the *simulation* process. Some Monte Carlo scenarios for aleas are calculated and some trajectories of stocks are computed using the optimal commands calculated in the first part. It looks like a loop of embarrassingly parallel computations, but these computations need data spread in all the intermediate result files stored in the second part and require many communications. Finally, applications results have to be saved in files from each processor at each time step, in order to derive some risk indicators.

2 Parallelization strategy

In the first part of the application, processor 0 opens and reads input files, and broadcasts data to all other processors. This solution appeared a little bit longer on small number of processors on Blue Gene supercomputer (20s in place of 19s) but allows large PC clusters to succeed to upload input data using classical file systems.

In the second part, we split a N-dimension cube of data and associated calculations (the *N-cube*) on an hypercube of $P = 2^k$ processors. Due to the nature of data and associated computations, some dimensions of the N-cube can not be split. Our splitting algorithm supports this constraint, and cuts each divisible dimension of the N-cube in order to compute the map of a load balanced data and calculation distribution at t_i. Then a routage plan is computed and executed to bring back on each processor the required t_{i+1} input data to achieve calculations of t_i data (this is a backward time step loop). So, each processor stores its *influence N-subcube* of t_{i+1} input data, and to minimize the size of the *influence N-subcubes* and to achieve great size up, our splitting algorithm

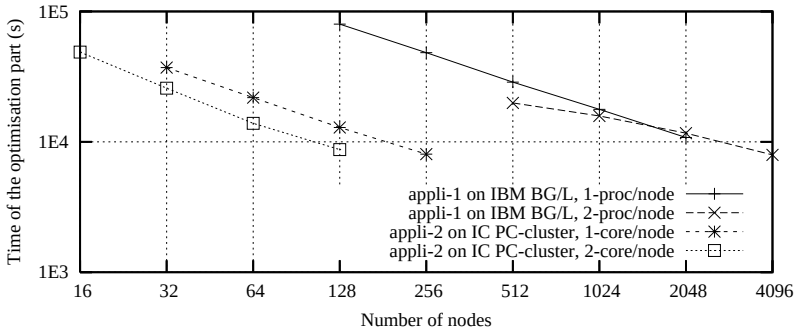

Fig. 2. Execution time of two 7-dimension stochastic control applications

attempts to create *cubic* sub-cubes of data and calculations on each processor (in place of *flat* sub-cubes), see figure 1. This is a generalization of the one-dimension algorithm we designed and validated on a gas storage valuation [3, 2]. At each step, each processor appends its intermediate results in four files, from t_n to t_0, and $4.P$ files written concurrently store all results of this second part of the application. On a Blue Gene supercomputer all intermediate results are stored on a remote disk across an efficient file system at each step, while on a PC cluster each processor stores its files on its local disk, and can overlap these file accesses with its communications and computations using multithreading.

The third part of the application is composed of set of N_s forward Monte-Carlo simulations, and each processor computes N_s/P simulations. These MC simulations are independent computations, but require intermediate results stored in files by the second part of the application. In order to minimize the number of file accesses, the implementation is a time step loop, and at each time step t_i each processor reads a file and loads data of step t_{i+1}. Then processors enter a second loop, and process one of their N_s/P simulations. However, to progress from t_i to t_{i+1} each simulation requires data not always located on the processor. So, processors compute data required to process one of their simulations, exchange data, process one of their simulations, and repeat these operations for each of their N_s/P simulations.

3 Current achievements

We have successfully implemented the first and second part of the application in C++ with MPI library, on a 256-PC cluster of SUPELEC (built by CARRI Systems company) and on the IBM Blue Gene/L supercomputer of EDF R&D. The algorithm of the last part has been designed and is under implementation. Our first experiments show that we succeed to achieve speedup and size up both on PC clusters up to 256 processors and on IBM Blue Gene/L up to 8192 processors. Figure 2 shows the execution time decreases regularly when processing two different 7-energetic-stock applications, running the same 7-dimension stochastic control algorithm (we used a modelization of a stock of water and 6 stocks of monthly peak and off-peak future products [5]). It is impossible to run these ap-

plications in sequential mode on one processor (lack of memory and prohibitive execution time), but parallel executions have succeeded.

Execution time of the input file reading and optimization parts of *application-2* was close to 8000s on 256 nodes of our PC cluster using one core per node, and on 128 nodes using 2 cores per node. Straight and approximately parallel curves on the left of figure 2 show the *application-2* speedups regularly on our PC cluster up to 256 nodes, and show we can efficiently use 2 cores per node. *Application-1* speedups regularly on Blue Gene/L supercomputer using one processor per node, and achieves an execution time close to 10800s on 2048 processors. Using two processors per node leads to better execution time up to 1024 nodes, but leads to worst time on 2048 nodes. When the numbers of nodes and communications increase it seems better to devote one processor per node to manage communications. Unfortunately it has not been possible to run *application-1* on 4096 nodes using only one processor per node (due to the load of the machine), but we succeeded to run *application-1* on 4096 nodes using 8192 processors and figure 2 shows the execution has decreased again to reach 8000s.

4 Perspectives

Next development steps will consist mainly in implementing the third part of the application and processing higher number of stocks on the new 32000 processor Blue Gene/P of EDF R&D. From a parallel computing point of view, we will further examine where the calculation time is now spent when we use thousands of processors and we will track optimizations on Blue Gene very large architecture. From an applicative point of view, we will extend our application so that it will be able to deal with others stocks appearing in electricity management asset problem. The methodology developed will bring some reference calculation that will help us to derive some simplified versions to use in production.

Acknowledgment: This research is part of the ANR-CICG GCPMF project, and is supported both by ANR and by Region Lorraine.

References

1. F. Longstaff and E. Schwartz. Valuing american options by simulation : A simple least-squares. *Review of Financial Studies*, 14(1), Spring 2001.
2. C. Makassikis, S. Vialle, and X. Warin. Large scale distribution of stochastic control algorithms for financial applications. In *The First International Workshop on Parallel and Distributed Computing in Finance*, Miami, USA, April 2008. To appear.
3. C. Makassikis, X. Warin, and S. Vialle. Distribution of a stochastic control algorithm applied to gas storage valuation. In *The 7th IEEE International Symposium on Signal Processing and Information Technology*, Cairo, Egypt, 2007.
4. J.N. Tsitsiklis and B. Van Roy. Optimal stopping of markov processes: Hilbert spaces theory, approximations algorithms and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control*, October 1999.
5. X. Warin and W. Van-Ackooij. Electricity asset management with future hedging. Technical Report H-R33-2006-04103-FR, EDF, 2008. to appear.