Massive Computation in Financial Markets

E. Tick

CIS-TR-92-15
July 1992

Abstract

Computational economics is a grand challenge in computer science: to model markets accurately, efficiently, and quickly. Likened to the importance of weather prediction to agriculture, economic predictions are extremely important to businesses, governments, and both private and group investors. The valuation of capital investments and financial instruments is the foundation of free market economies, where debt and equity are traded in great volume. This paper discusses the use of massively parallel computation to analyze financial investments. We discuss the advantages and pitfalls of bringing supercomputers to bear on the problem.

This paper was presented in the *Workshop on Future Directions of Parallel Programming and Architecture*, ICOT, Tokyo, June 1992.

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
UNIVERSITY OF OREGON

1 Introduction

This paper concerns a grand challenge in massively parallel computation: modeling financial markets in a free global economy. Such work has been overshadowed by recent rapid progress in genome mapping in biology and computational chemistry. The hard sciences have always enjoyed being positioned as the main target of supercomputing and parallel processing. This is primarily because of government funding, in addition to the natural mapping of physical models and simulations onto multiprocessors. On the other hand, the field of computational economics, and specifically market modeling, is less mature, and more difficult to map onto regular mathematical models.

With the collapse of communist Europe, the open market conversion of these countries is a major undertaking. The return on Hong Kong to the PRC could likely crystalize a free market movement in China also (which has already slowly begun in Shenzhen). The globalization of markets increases the importance of efficiently allocating resources. Furthermore, there is no doubt that financial intermediaries need all the help they can get concerning long-term investment strategies. Consider the LDC loan defaults, the U.S. thrift and insurance company failures due to heavy investment in junk bonds and real estate, the current Japanese bank losses due to stock depreciation, etc. More astute estimation of risk and return should increase the efficiency of markets, lowering the cost of money and thereby fostering economic growth.

Computational economics is becoming one of the more sophisticated uses of supercomputers [17]. This report may differ somewhat from the suggested topics of the workshop because instead of focusing on the details of what accomplishments and technology we have witnessed over the past decade of growth in parallel processing, I will concentrate on introducing the potentially explosive field of computational market modeling. Furthermore, I will not discuss progress in traditional economic modeling and optimization of commodities markets, because this field is relatively well established (for instance using techniques of linear and integer programming). Instead, I wish to emphasize applications to real markets in debt, equity, and currency. In addition, I will focus on applications that require GFLOPS (10^9 floating-point operations per second), as opposed to qualitative macroeconomic modeling that is also an emerging area (e.g., [9]), which requires less raw power, but provides less exact results.

By "computational market modeling," I mean exploiting supercomputer power to solve a generally diverse collection of problems in financial analysis and market making. These problems include, but are no means limited to:

- <u>Creating secondary market instruments and pricing them.</u> Most popular are certificates of deposit (CDs) in the money market and mortgage backed securities

(MBSs) in real estate. However other securities include include CMOs (collateralized mortgage obligations), STRIPS (Separate Trading of Registered Interest and Principle of Securities), CDs indexed on college tuitions, CARs (automobile debt), and CARDs (credit card debt). Valuating these is complex because of fluctuating interest rates and the "maturity matching problem:" borrowers can prepay their debt at any time. Furthermore, as the instruments get more complex, the number of adjustable parameters explodes the space.

- Pricing stocks and rating bonds. The most exercised class of secondary market instruments are stocks and bonds. These markets are enormous because they are heavily traded "over the counter." Furthermore the national debt is financed with Treasury instruments. Because margins narrow as market competition increases, it becomes increasingly important to accurately price the instruments. With bonds the objective is to better estimate the yield vs. maturity, and "ride the yield curve." Rating bonds, to assess risk (e.g., junk vs. *junque*), has been undertaken by neural networks [8, 21, 4]. Neural nets have also been used extensively to price stocks [15, 14, 24]. Usually these algorithms involve "learning" by back propagation, which requires extensive computational power.

- portfolio optimization. Pricing single instruments is of limited use — to reduce risk, sets of investments, called portfolios, are needed. Examples are MBS pools and groups of stocks and options. Valuation of portfolio risk and return is an embarrassingly parallel application because of the number of constituent instruments that need valuation.

- options and futures. Options are a contract to buy or sell an instrument at or before a future date at a guaranteed price. These can be used to "insure" gains to some minimum loss in an investment, called hedging. Pricing of combinations of options and shares is significantly more complex than pricing the instruments themselves. For example, pricing with the Black-Scholes algorithm was demonstrated in CLP(R) [11]. This tool illustrates the maxim that as investment strategies get more complex, simple number crunching alone is not sufficient — symbolic analysis is also needed.

- foreign currency trading. Calculating exchange rates (i.e., the price of currencies with respect to a baseline) and is critical to currency arbitrage. Neural networks have been demonstrated for this purpose [18, 23]. With multistep prediction (feedback of daily predictions into the network's input), a 20% profit was achieved compared to standard strategies which profit 2–5% [18].

- databases. Financial markets are data intensive and the manipulation of market databases requires great computational power. In 1988 Dow Jones Inc. announced the purchase of a Connection Machine for its database service, and discussed also moving its stock forcasting service to a supercomputer [3, 17].

In the following section I will summarize the results of three of these areas: pricing mortgage-backed securities under uncertain market conditions, pricing stock options, and pricing stocks.

2 Mortage-Backed Security Pricing

A mortgage-backed security (MBS) is a secondary-market instrument created by pooling together many real estate mortgages. The security is a pay-through stake in the pool, and the investors receive cashflows from the payments to the pool by the borrowers, via a lending institution who issues the MBS. By buffering cashflows in this manner, the lending institutions and the investors reduce risk and increase the liquidity of the market. Furthermore, maturity matching is somewhat better accomodated because of the mortgage mix in the pool. Nevertheless, when interest rates drop, as they did in the winter of 1992, overall rates of refinancing increase, and prepayments can still plague the investors.

MBSs are important because they represent a class of complex securities that are increasing in world money markets. The ability of the borrower to prepay the loan is a form of a call option, since the principle is fixed. As Zenios [25] points out, once a single market player can value these securities accurately, all players must do so or lose their shirts. The key point is to value the instrument in *real time*, because otherwise the opportunity to buy/sell the instrument at the market price is gone. In this case, real time may be on the order of *minutes.*

Zenios has done extensive work in parallelizing compute-intensive algorithms to accurately price MBSs [10, 25, 16, 5, 6, 26, 27]. The underlying foundation of his work is how to relate possible interest rate fluctuations (over the life of the security, which can be about 360 periods), to cashflows. The algorithms follow three basic phases:

1. Interest rate paths are generated, a massively data-parallel computation because of the number of periods involved. Effectively a tree is formed, where each branch represents a different rate history.

2. Cashflows are generated among the paths, by modeling prepayment rates. By clever manipulation of the recurrence equations, a *scan multiply* data-parallel instruction can be used to evaluate the flows.

3. Net present values (NPVs) are computed from cashflows. (Actually, *option adjusted spreads* with respect to Treasury yields are calculated, but abstractly the NPVs are sufficient). Again, scan operations can used to exploit data parallelism.

Zenios has demonstrated that a 16,000-processor CM-2a implementation of his tools can analyze 3,000 MBS benchmarks in 40 minutes. Extrapolating, he claims to be able to do the analysis in 5 minutes on a 64,000 processor CM-2.

3 Stock-Option Pricing

The Options Trading Analysis System (OTAS) [11] is an software tool which computes options strategies, such as spreads. Written in CLP(R), a constraint logic programming language on the reals [12, 13], the system can produce symbolic solutions to queries, greatly increasing the flexibility of its use. Its backtracking capability facilitates queries over large data spaces, allowing simple specification of complex searches.

Constraint systems have proved effective in a wide range of areas, ranging from manpower scheduling to circuit testing. Financial analyses are natural applications because they are primarily specified as mathematical models. Classical methods of solution mostly involve mathematical programming, such as linear programming. Constraint systems show two primary advantages over these methods. First, problem specification is easier because a high-level language is used. Second, problem specifications can be extended with heuristics when mathematical models break down. Other notable constraint languages have been implemented, e.g., CHIP [22] and CAL [19].

A constraint system, however, is only as strong as its solver. OTAS, built on a CLP(R) system without the capability to solve nonlinear equations, requires meta-level procedures to approximate these by breaking down more complex constraints. This frustrates the elegance of the paradigm and indicates the necessity to supply powerful engines across a range of domains. Sophisticated systems will require supercomputer speeds to do practical financial analysis. For example, OTAS could solve a query involving nine stocks among 100 choices, producing 45 option positions in 5 minutes on an IBM RTPC. More complex searches over full exchanges (thousands of stocks) will require significantly higher speeds. Concurrent constraint languages (CCLs) [20] promise to supply the framework needed.

4 Stock Pricing

Predicting stock prices has been a popular application for neural networks [15, 14, 24]. Back-propagation teaching techniques require significant computational power. For example, Derr [7] trained a two-hidden-layer feedforward completely-connected network on 625 days of IBM stock prices. To achieve low error, 10,000 epochs (teaching paths through the network) required 4.5 hours on five 80386 Sequent Symmetry processors.

Supercomputer speeds are needed to allow more extensive teaching sets, more complex networks, and more sophisticated teaching algorithms (e.g., [23]). Note that after a network is taught by back-propagation, incremental teaching is not practical. Thus increased supercomputer speeds can permit more frequent reteaching. Because neural network models are inherently concurrent, massively parallel computers can be exploited — parallel architectures and organizations for networks are generating great interest in the research community.

5 Advantages

The value added by supercomputing as opposed to desktop computing, is significant increases in accurate information in a timely manner. The supercomputing required includes both massive parallelism, for data crunching, as well as fast scalar performance, for qualitative analysis. Neural networks, data-parallel algorithms, and constraint systems are a few of the technologies emerging for these applications. These paradigms are only possible because of increased computational power. Moreover it is becoming clear that both number crunching and qualitative reasoning tools are both required, as market complexities outrun analytical models [9].

Interestingly, this exploitation of supercomputing in finance, in contrast to the hard sciences, is inherently a real-time problem, since markets can change rapidly. Information derived from large data flows describing the markets is needed on a daily basis. Also in contrast to the hard sciences, these applications are not experimentally decoupled from their analysis. More accurate modeling will drive the markets in new directions, a result of decreasing margins. Complexity (e.g., of offered securities) will increase, feeding back to require more computational power.

These applications represent a new generation of financial intermediation wherein the expertise of the intermediary is greatly enhanced by computational power. In a free and open market, assuming equal access to such analysis, information bandwidth will increase. In other words, margins will become smaller: pricing will more accurately account for cost. Intermediaries will be driven to offer more sophisticated products with increased utility for their customers. Individuals in the market, seeking to profit from an information imbalance from these tools, will in conjunction create more efficient markets. In particular, better modeling of the affects of currency prices on debt and equity investments will create more efficient global markets.

6 Disadvantages

There are also potential problems with the "computerization" of markets, just as there are concerns with detrimental mutations in genetics. A curious example in the spring of 1992 was an accidental sell order by a New York Stock Exchange (NYSE) clerk of X *shares* instead of $X worth of a portfolio in the closing minutes of the day [1]. Such a large transaction was possible because of computer technology that had no safety checks. The single transaction actually depressed the Dow Jones temporarily. A more damning example is the October 1987 NYSE crash, attributed by some to "programmed trading" run amok. Programmed trading is the software controlled transaction of a significant number of options and shares of different stocks at one time. In an attempt to regulate this, the SEC limits program trades to fall within certain constraints, such as volume and depth. However, not all world exchanges have such constraints, and as the trend of 24-hour trading begins to evolve [2], it will become increasingly difficult to monitor the markets. The emergence of supercomputers instead of desktops driving trades may have additional detrimental affects.

7 Summary

In summary, computational economics is an expanding field joining computer science and economics with the goal of realizing real-time analyses of economic models and markets. In the area of financial markets, current opportunities exist for further developing neural networks, databases, data-parallel risk/return algorithms, and constraint systems. These technologies require supercomputer performance, and moreover promise to exploit massively parallel multiprocessors, to achieve the real-time advantage desired.

Acknowledgements

E. Tick was supported by an NSF Presidential Young Investigator award, with matching funds generously provided by Sequent Computer Corporation.

References

[1] Clerk's Error Spurs Action From Street. *Wall Street Journal.* March 26, 1992.

[2] Tiny Finex Takes Big Step in Round-the-Clock Trading. *Wall Street Journal.* March 20, 1992.

[3] Supercomputers Break Through. *Datamation*, May 1988.

[4] E. Collins, S. Ghosh, and C. Scofield. An Application of a Multiple Neural Network Learing System to Emulation of Mortgage Underwriting Judgements. In *International Joint Conference on Neural Networks*, volume II, pages 459–466. IEEE Computer Society, July 1988.

[5] H. Dahl, A. Meeraus, and S. A. Zenios. Some financial optimization Models: I. Risk Management. *Financial Optimization*, 1992.

[6] H. Dahl, A. Meeraus, and S. A. Zenios. Some financial optimization Models: II. Financial Engineering. *Financial Optimization*, 1992.

[7] C. Derr. personal communication, University of Oregon, May 1992.

[8] S. Dutta and S. Shekhar. Bond Rating: A Non-Conservative Application of Neural Networks. In *International Joint Conference on Neural Networks*, volume II, pages 443–450. IEEE Computer Society, July 1988.

[9] A. M. Farley and K.-P. Lin. Qualitative Reasoning in Economics. *Journal of Economic Dynamics and Control*, 14:465–490, 1990.

[10] J. M. Hutchinson and S. A. Zenios. Financial Simulations on a Massively Parallel Connection Machine. *The International Journal of Supercomputer Applications*, 5(2):27–45, 1991.

[11] T. Huynh and C. Lassez. An Expert Decision-Support System for Option-Based Investment Strategies. *Computers Mathematical Applications*, 20(9/10):1–14, 1990.

[12] J. Jaffar and J.-L. Lassez. Constraint Logic Programming. In *SIGPLAN Symposium on Principles of Programming Languages*, Munich, 1987. ACM Press.

[13] J. Jaffar and S. Michaylov. Methodology and Implementation of a CLP System. In *International Conference on Logic Programming*, pages 196–219. University of Melbourne, MIT Press, May 1987.

[14] K. Kamijo and T. Tanigawa. Stock Price Pattern Recognition — A Recurrent Neural Network Approach. In *International Joint Conference on Neural Networks*. IEEE Computer Society, 1990.

[15] T. Kimoto and K. Asakawa. Stock Market Prediction System with Modular Neural Networks. In *International Joint Conference on Neural Networks*. IEEE Computer Society, 1990.

[16] S. S. Nielsen and S. A. Zenios. A Massively Parallel Algorithm for Nonlinear Stochastic Network Problems. Technical Report 90-09-08, The Wharton School, University of Pennsylvania, September 1991.

[17] P. Penczer. Supercomputers Era Dawns on Wall Street. *Wall Street Computer Review*, November 1989.

[18] A. N. Refenes, M. Azema-Barac, and S. A. Karoussos. Currency Exchange Rate Forecasting by Error Backpropagation. In *Hawaii International Conference on System Sciences*, volume 4, pages 504–515. Kauai, IEEE Computer Society, January 1992.

[19] K. Sakai and A. Aiba. CAL: A Theoretical Background of Constraint Logic Programming and Its Applications. *Journal of Symbolic Computation*, 8:589–603, 1989.

[20] V. A. Saraswat. *Concurrent Constraint Programming.* MIT Press, Cambridge MA, 1990.

[21] A. J. Surkan and J. C. Singleton. Neural Networks as Bond Rating Tools. In *Hawaii International Conference on System Sciences*, volume 4, pages 499–503. Kauai, IEEE Computer Society, January 1992.

[22] P. Van Hentenryck. *Constraint Satisfaction in Logic Programming.* MIT Press, Cambridge MA, 1989.

[23] A. S. Weigend, D. E. Rumelhart, and B. A. Huberman. Generalization by Weight-Elimination with Application to Forecasting. In *Advances in Neural Information Processing Systems 3*, pages 875–882. Morgan Kaufmann Publishers Inc., 1991.

[24] H. White. Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns. In *International Joint Conference on Neural Networks*, volume II, pages 451–458. IEEE Computer Society, July 1988.

[25] S. A. Zenios. Massively Parallel Computations for Financial Planning Under Uncertainty. In J. P. Mesirov, editor, *Very Large Scale Computation in the* 21^{st} *Century.* Society for Industrial and Applied Mathematics, Philadelphia PA, 1991.

[26] S. A. Zenios and P. Kang. Mean-Absolute Deviation Portfolio Optimization for Mortgage-Backed Securities. Technical Report 92-03-04, The Wharton School, University of Pennsylvania, February 1992.

[27] S. A. Zenios and R. A. McKendall. Pricing and Re-Pricing Mortgage-Backed Securitiess using Massively Parallel Computing. Technical Report 91-12-01, The Wharton School, University of Pennsylvania, December 1991.