Applications of High-Performance Computing in Finance

Stathis Tompaidis

University of Texas at Austin

www.mccombs.utexas.edu/faculty/stathis.tompaidis

SAMSI — Workshop on Financial Mathematics, Statistics and Econometrics

September 21, 2005

Computationally Demanding Problems

- Monte Carlo simulation problems
  - Derivative pricing
  - Portfolio risk management
  - Markov Chain Monte Carlo simulation
- Dynamic Programming
  - Optimal asset allocation
  - Optimal stopping
- Other problems; e.g., solving general equilibrium models

This talk will focus on parallel programming resources/algorithms and explore their limits. Even though parallel programming can not address the curse of dimensionality, it allows us to extend well understood methods to more complicated problems.

Monte Carlo Simulation

- Monte Carlo simulation is based on the central limit theorem.
- Given independent, identically distributed variables X_i, the sum $\frac{1}{N}\sum_{i=1}^{N} X_i$ is approximately distributed as a normal random variable with mean equal to $E(X_i)$, and standard deviation proportional to $\frac{1}{\sqrt{N}}$.
- To estimate $E(X)$ one can simply generate N outcomes drawn from the distribution of X and average them.
- In Finance, many quantities of interest are expressed as expected values; e.g., option prices.

Derivatives of European Option Prices

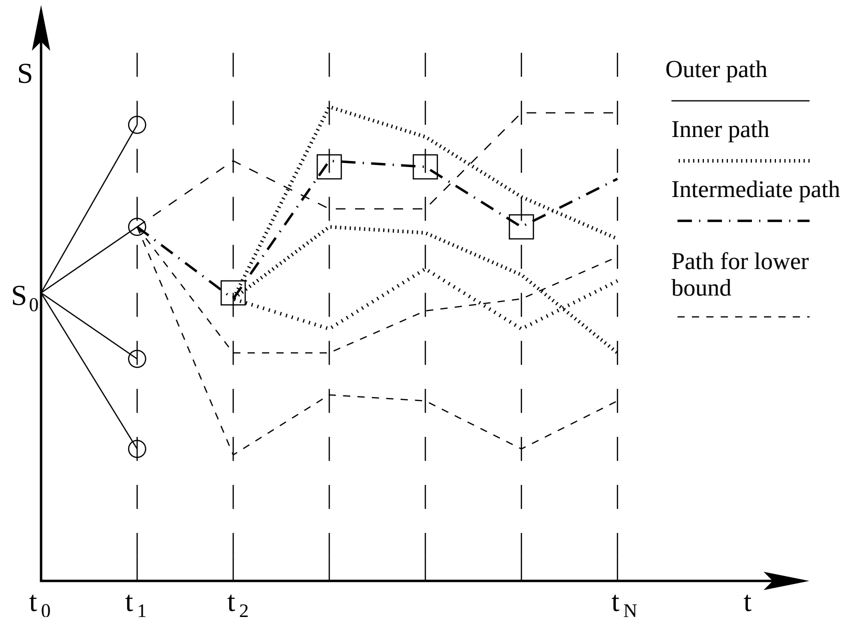
For European options, one can calculate derivatives of the option price as a weighted expected value (Broadie, Glasserman 96):

$$
\begin{aligned}
\Delta_i(x) = \left.\frac{dQ_0}{dx_i}\right|_{S(0)=x} &= \frac{d}{dx_i}\mathbb{E}_0\left[e^{-\int_0^T r(t)dt}h(S(T))\right] \\
&= \frac{d}{dx_i}\left[\int e^{-\int_0^T r(t)dt}h(S(T))g(S(T),x)dS(T)\right] \\
&= \int e^{-\int_0^T r(t)dt}h(S(T))\frac{\partial g(S(T),x)}{\partial x_i}dS(T) \\
&= \int e^{-\int_0^T r(t)dt}h(S(T))\frac{\partial \ln g(S(T),x)}{\partial x_i}g(S(T),x)dS(T) \\
&= \mathbb{E}_0\left[e^{-\int_0^T r(t)dt}h(S(T))\frac{\partial \ln g(S(T),x)}{\partial x_i}\right]
\end{aligned}
$$

where Q_0 is the European option price, h is the payoff, $g(S(T),x)$ the transition density of reaching $S(T)$, given $S(0) = x$. Thus, derivatives of European options can be calculated by Monte Carlo simulation.

Derivatives of Bermudan Option Prices

One can also calculate bounds on Bermudan option prices using an idea based on a dual representation of option prices, and a numerical approximation based on a doubly-nested loop (Davis, Karatzas 94, Rogers 02, Haugh, Kogan 04, Andersen, Broadie 04). It is possible to combine the two methods in a triply-nested Monte-Carlo simulation and achieve bounds on the derivatives of Bermudan options (Kaniel, T, Zemlianov 05).



Parallelizing Monte-Carlo

Monte-Carlo simulations are straightforward to parallelize.

- Partitioning the load is simple
- Communication between processors is very limited
- Each processor generates independent draws and reports means and standard deviations to a central processor
- Care should be used in the “random” number generation between different processors
- We have implemented this process for calculating bounds on derivatives of Bermudan option prices and verified that the speed of the algorithm scales linearly with the number of processors.

Dynamic Programming

In dynamic programming, we solve an optimal control problem by maximizing an objective function. The method consists of solving problems in a nested fashion, where the solution of one problem becomes an input for the next problem. A simple example of dynamic programming is a multi-player game, where the players play only once, in a certain order and each player tries to optimize an objective function. The solution is to find the optimal strategy of the last player taking into account the positions of the earlier players, and then find the optimal strategy of each player prior to the last one in order, taking into account the optimal strategy of the following players.

In Finance, dynamic programming is used in pricing American options and in optimal asset allocation problems.

Example: Tax Management Strategies with Multiple Risky Assets

(Gallmeyer, Kaniel, T 05)

Setup: we consider the case of an investor that maximizes utility from consumption and bequest over his lifetime, and whose investment opportunity set consists of one riskless and two risky assets. The investor pays taxes on dividends, interest, and realized capital gains.

Motivation: taxation is a significant determinant of asset allocation decisions. When multiple risky assets are available, the investor may face conflicting goals of diversification (to reduce risk) and the avoidance of trading (to avoid realizing capital gains). Without capital gains taxes, an investor that has access to risky assets will hold an index. This is not true with capital gains taxes.

Computational Complexity

- Four State variables:
  - Initial asset allocation for each risky asset (2)
  - Initial basis for each risky asset (2)
- Three Choice variables
  - Optimal asset allocation for each risky asset (2)
  - Consumption (1)
- Time

Given that, at each time, the problem can be solved independently for each initial allocation/basis, we solve in parallel, by partitioning the state-space grid at each time between several processors. Communication is limited to reporting the optimal allocations/consumption to a central processor, and receiving information from the central processor, for each time, of the value of the objective function at all the points of the state-space grid.

Systems available at UT-Austin

- longhorn
  - IBM Power4 System, running AIX
  - 224 processors: 224, 1.3 GHz each
- tejas
  - 64 processors, 1.0 GHz each
- Other resources
  - lonestar
    - Cray-Dell PowerEdge Xeon Cluster, 1024 processors, 3.06 GHz each, running Linux
    - Point-to-point bandwidth: 250 MB/sec
  - roundup
    - 1297 processors, heterogeneous in terms of speed/OS.
    - Uses United Devices software (SETI project)

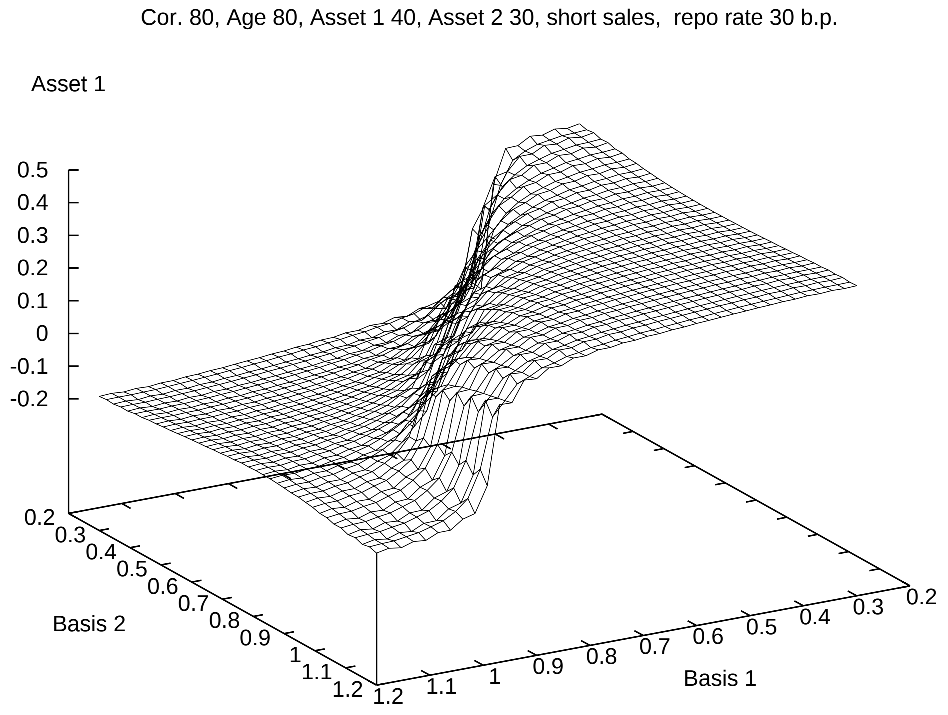
Performance for the Optimal Asset Allocation Problems with Capital Gains Taxes

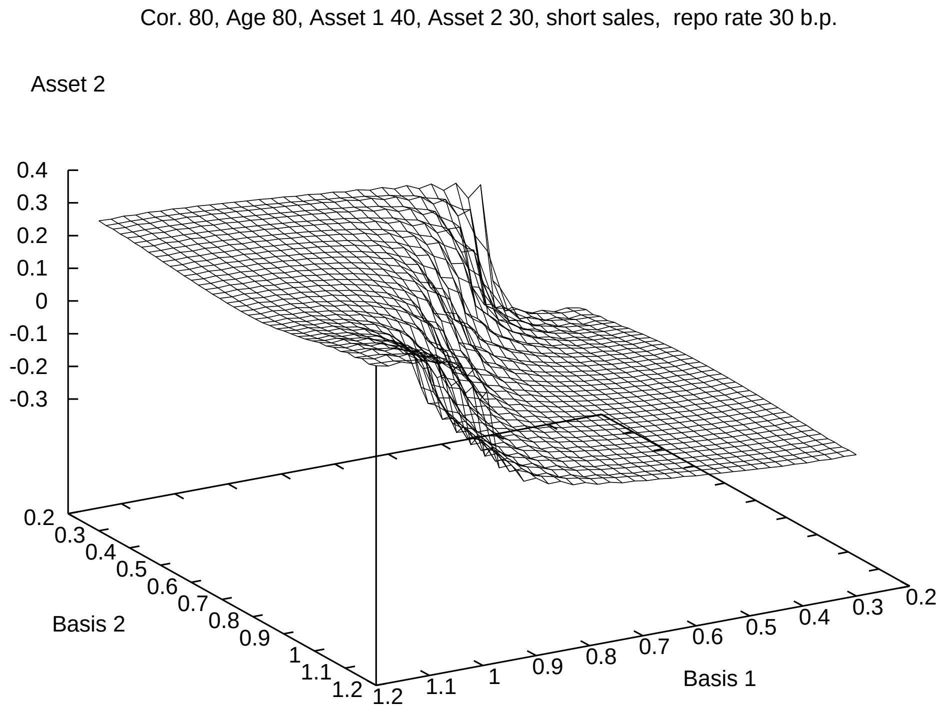
longhorn		tejas	
Processors	Nodes/Proc/Min	Processors	Nodes/Proc/Min
16	1289	14	1003
16	1125	14	966
16	1291	16	1000
16	1310	16	1001
16	1326	16	969
16	1305	16	964
16	1308	16	959
16	1334	16	979
16	1255	16	969
16	1313	16	971
48	1310	20	963
64	1142	24	960
64	1132	32	946

Slight decrease in performance with an increase in the number of processors — evidence of increasing communication needs.

Optimal Asset Allocations

We are able to identify situations where the investor, ex-ante, prefers a non-diversified position to minimize the realization of capital gains in the future.





Conclusions

- Parallelization allows to study complex problems in Finance
- Monte-Carlo simulation and Dynamic Programming can be easily parallelized, since tasks are largely independent and can be partitioned among different processors, and the need for communication is limited.
- Even though parallelization allows for the study of complex problems, the curse of dimensionality is not overcome.