中国科学院超级计算中心
Supercomputing Center of Chinese Academy of Sciences

DCABES2009

Large-scale Parallel Simulation of High-dimensional American Option Pricing

Hongxu Chang
Supercomputing Center, Chinese Academy of Sciences

Wuhan, October 17, 2009

Abstract

- ✓ High-dimensional American option pricing is computational challenging in both theory and practice.
- ✓ Using stochastic mesh method combined with performance enhancement policy by bias reduction to solve this practical problem in classic Black-Scholes framework.
- ✓ Effectively parallelize this algorithm, use MPI for implementation and execute large-scale numerical experiments on DeepComp7000.
- ✓ Numerical study of parallel simulation.

Outline

1、 Background & Motivation

2、 Stochastic Mesh Method

3、 Parallelization of Stochastic Mesh Method

4、 Numerical Experiment

5、 Numerical Study

6、 Conclusions

Background & Motivation

- Emerging Computational Finance Research
- Exotic/Complicated Derivatives Pricing

  High-dimensional American Option, CDO, etc
- Computational Demands of Financial Industry

  Pricing, Risk Management, Algo Trading, etc
- Development of HPC Technique

  Multi-core, Cluster, GPU, etc

Background & Motivation

Monte Carlo Simulation-based method is popular in derivatives pricing and can easily dispose of high-dimensional/path-dependent problems.

- The convergence rate independent of the dimensionality of the specific problem on hands.
- Quite flexible to different option payoff types.
- Implementation easily.

Large-scale parallel Monte Carlo simulation can efficiently overcome the computational challenges in high-dimensional American option pricing.

Background & Motivation

Problem Formulation

$S_t = (S_t^1, S_t^2, \cdots, S_t^n)$ is a Markov process on R^n with fixed initial value S_0

$$dS_t^i = (r - \delta_i) S_t^i dt + \sigma_i S_t^i dW_t^i \,,\ i = 1, 2, \cdots, n$$

$$\mathrm{cov}(W_t^i, W_t^j) = \rho_{ij} t \quad \Sigma_{ij} = \rho_{ij} \sigma_i \sigma_j \,, \Sigma = (\Sigma_{ij})_{n \times n} = CC^T$$

$$S_t^i = S_0^i \exp\{(r - \delta_i - \frac{1}{2}\sigma_i^2) t + Z_t^i\} \qquad Z_t = C \cdot RNum$$

the transition density function of S_{t+1} given S_t

$$f(S_t, S_{t+1}) = (2\pi\Delta t)^{-\frac{n}{2}} \, | C^{-1} | \, (\prod^{n} S_{t+1}^i)^{-1} \exp\{-\frac{\| \Theta(S_t, S_{t+1}) \|^2}{2\Delta t}\}$$

$$\Theta(S_t, S_{t+1}) = \left[\log \frac{S_{t+1}^i}{S_t^i} + (\frac{1}{2} \sum_{1 \le l \le n} C_{il}^2 - r + \delta_i) \Delta t \right]_{1 \le i \le n}$$

Background & Motivation

Problem Formulation

$h(t, S_t)$ the payoff function of high-dimensional American option at exercise date t

max-option on n assets $h(t, S_t) = \max\{\max\{S_t^1, S_t^2, \cdots, S_t^n\} - K, 0\}$

geometric average option on n assets $h(t, S_t) = \max\{(S_t^1 S_t^2 \cdots S_t^n)^{\frac{1}{n}} - K, 0\}$

American option allows exercising at any time before the maturity.

$$\begin{cases} P(M, S_M) = h(M, S_M) \\ P(t, S_t) = \max\{h(t, S_t), C(t, S_t)\} \qquad t = M-1, \cdots, 1, 0 \end{cases}$$

$$C(t, u) = e^{-r\Delta t} E(P(t+1, S_{t+1}) \mid S_t = u)$$

Background & Motivation

Longstaff and Schwartz(2001): regression-based method

basis function choosing/lower bound

Broadie and Glasserman:

random tree method(1997), computational prohibitive

stochastic mesh method(1997, 2004), effectively

Avramidis and Hyden(1999):

performance enhancement policies

Avramidis and Matzinger(2004):

convergence of stochastic mesh method in theory

Background & Motivation

Haugh and Kogan(2004), Roger(2002) :

the duality approach

Andersion and Broadie(2004):

the primal-dual algorithm

Inanez and Zapatero(2004):

estimating the optimal exercise frontier

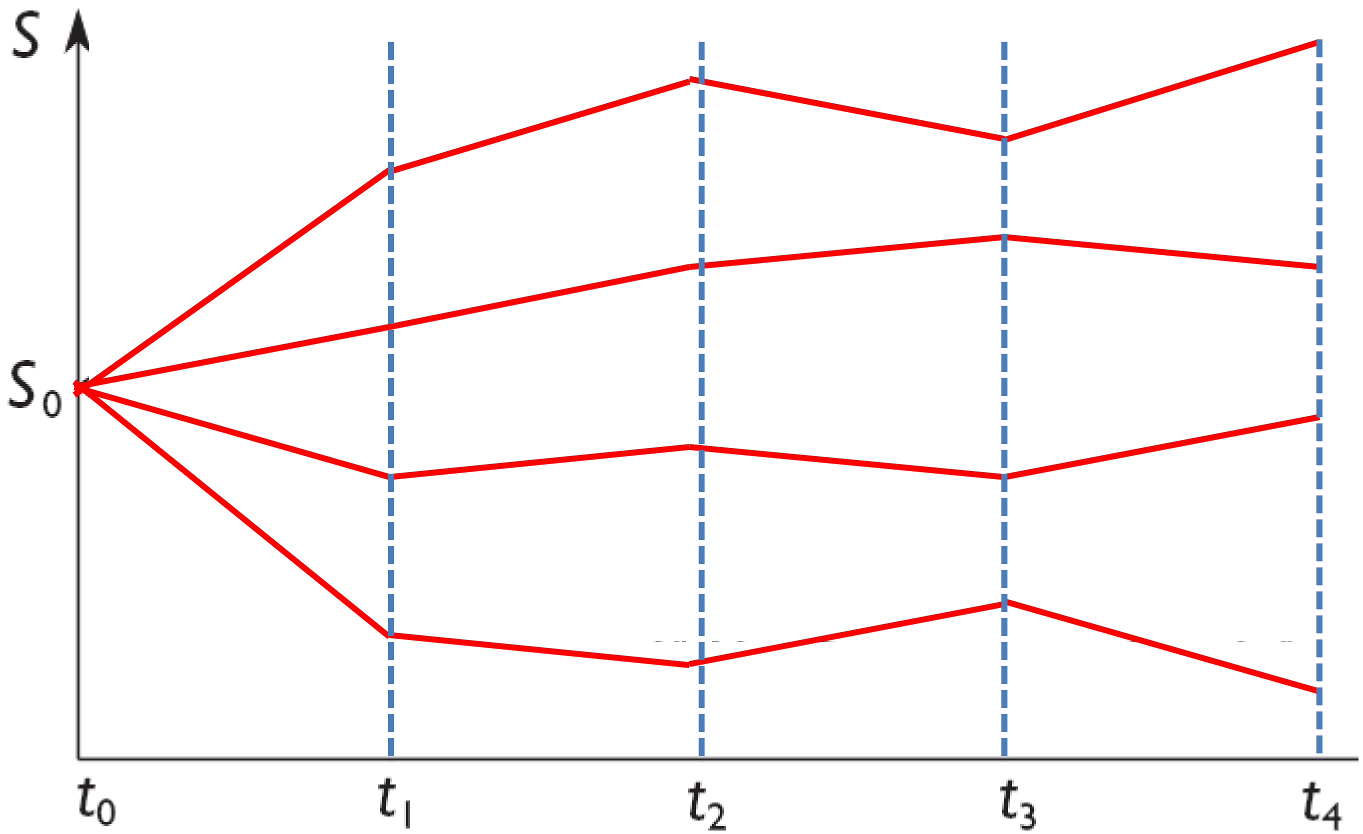
Kargin(2005):

lattice-based method/nonlinear high-dimensional interpolation

……

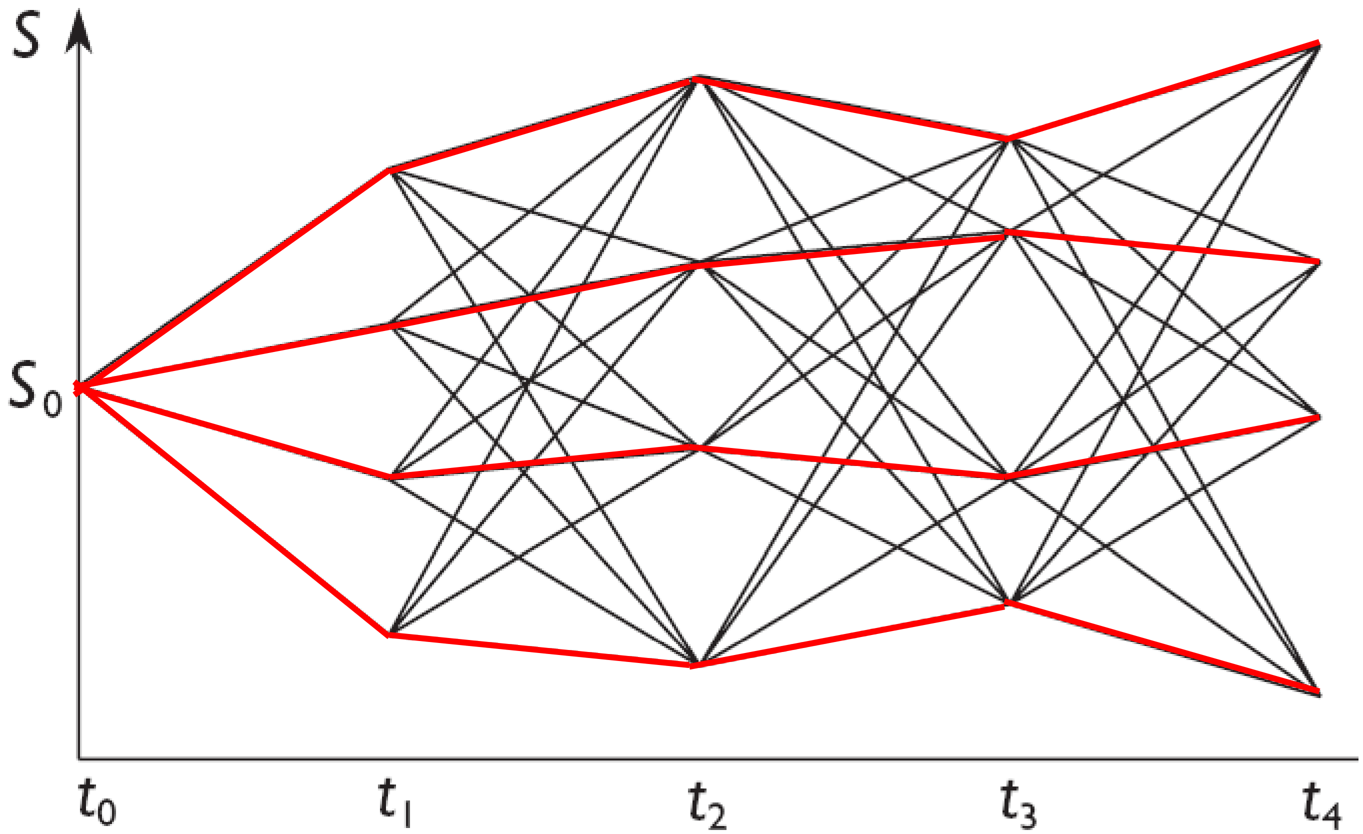
Stochastic Mesh Method

Simulate the underlying price……



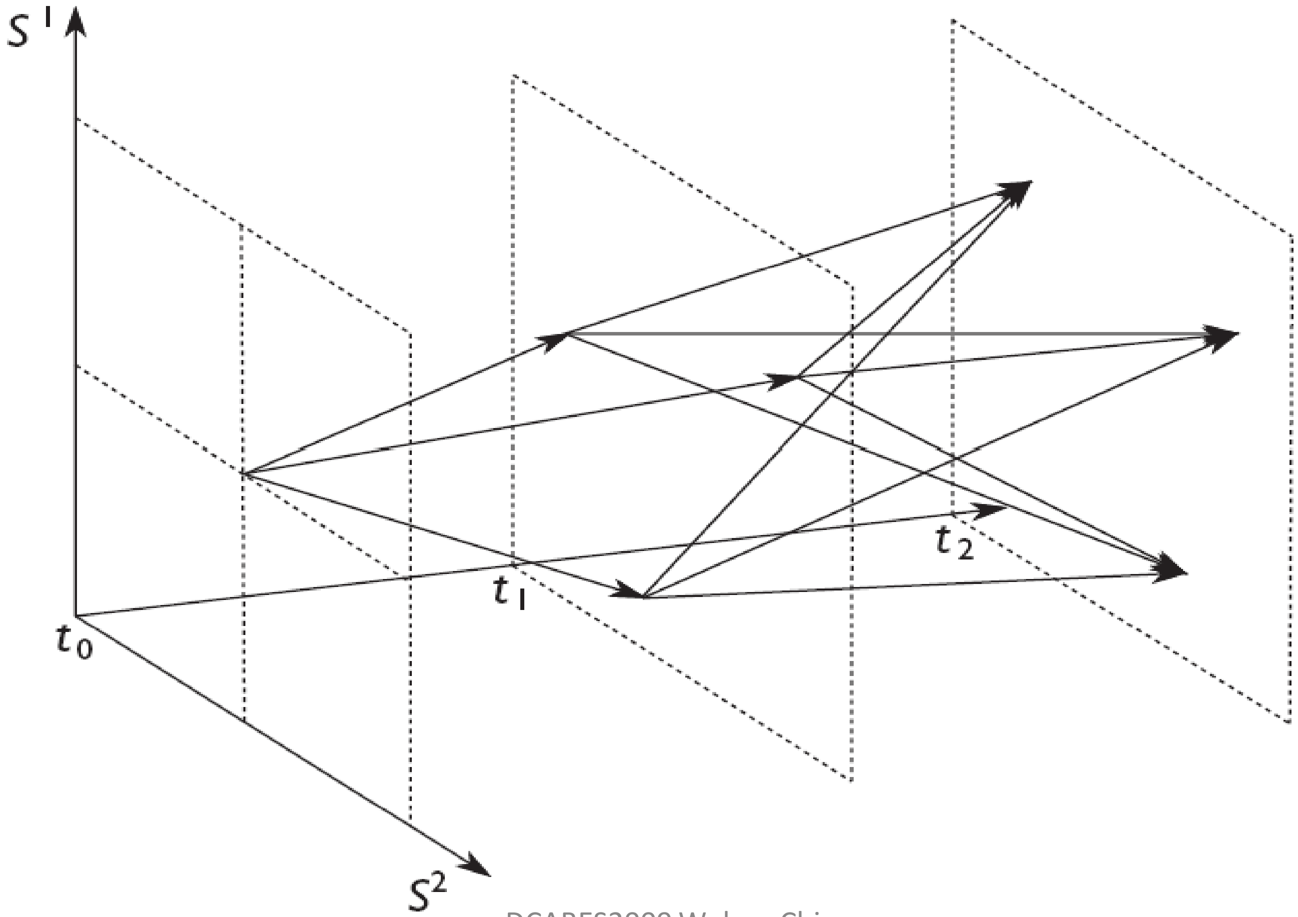
Stochastic Mesh Method

……and construct the stochastic mesh



Stochastic Mesh Method

High-dimensional case



Stochastic Mesh Method

High-biased estimator

$$\begin{cases} \hat{P}_H(M, S_M(i)) = h(M, S_M(i)) \\ \hat{P}_H(t, S_t(i)) = \max\{h(t, S_t(i)), \hat{C}(t, S_t(i))\} \end{cases}$$

$$\hat{C}(t, S_t(i)) = e^{-r\Delta t} \frac{1}{b} \sum_{j=1}^{b} \boxed{\hat{P}_H(t+1, S_{t+1}(j))}\, \underline{w(t, S_t(i), S_{t+1}(j))}$$

$$w(t, S_t(i), S_{t+1}(j)) = \frac{f(S_t(i), S_{t+1}(j))}{\frac{1}{b}\sum_{k=1}^{b} f(S_t(k), S_{t+1}(j))}$$

b: mesh size

Stochastic Mesh Method

Low-biased estimator(== multi-asset European option pricing)

forward simulate $nbMC$ paths of assets dynamics respectively

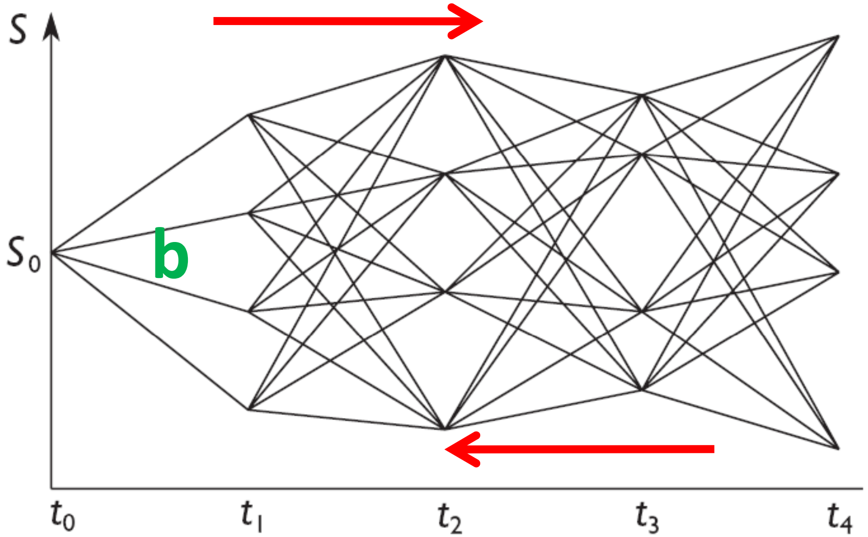
derive the option value by using $\boxed{\text{the suboptimal exercise policy}}$

Replicate the mesh for N times

average the high-biased and low-biased estimators of option value respectively,
construct a 1−α (α =5%, 10%, …) level confidence interval of the option value,
point estimate

Stochastic Mesh Method

Step1. mesh generation;
Step2. stochastic dynamic programming;
Step3. mesh replication.



Improved Stochastic Mesh Method

Drawback of Standard Stochastic Mesh Method: existence of the estimated bias

$\hat{P}(t, S_t(i))$ **the estimated value of the true option value**

$\hat{P}_{HA}(t, S_t(i))$ **the high-biased average estimator**

$\hat{P}_{LA}(t, S_t(i))$ **the low-biased average estimator**

$$\begin{cases} \hat{P}_{HA}(M, S_T(i)) = h(M, S_M(i)) \\ \hat{P}_{HA}(t, S_t(i)) = \max\{h(t, S_t(i)), e^{-r\Delta t} \frac{1}{b} \sum_{j=1}^{b} \hat{P}(t+1, S_{t+1}(j)) w(t, S_t(i), S_t(j))\} \end{cases}$$

b:mesh size

Improved Stochastic Mesh Method

set $B = \{1, 2, \cdots, b\}$, $B_j = B - \{j\}$

The estimate of the continuation value only using the mesh points in *I* at exercise date *t*

$$\hat{C}(t, S_t(i), I) = e^{-r\Delta t} \frac{1}{|I|} \sum_{j \in I} \hat{P}(t+1, S_{t+1}(j)) w(t, S_t(i), S_{t+1}(j))$$

$$\hat{P}_L(t, S_t(i), I) = \begin{cases} h(t, S_t(i)) & \text{if } h(t, S_t(i)) \geq \hat{C}(t, S_t(i), I) \\ \hat{C}(t, S_t(i), I^c) & \text{otherwise} \end{cases}$$

$$\begin{cases} \hat{P}_{LA}(M, S_M(i)) = h(M, S_M(i)) \\ \hat{P}_{LA}(t, S_t(i)) = \frac{1}{b} \sum_{j=1}^{b} \hat{P}_L(t, S_t(i), B_j) \end{cases}$$

b:mesh size

Improved Stochastic Mesh Method

$$\begin{cases} \hat{P}(M, S_M(i)) = h(M, S_M(i)) \\ \hat{P}(t, S_t(i)) = \dfrac{1}{2}(\hat{P}_{HA}(t, S_t(i)) + \hat{P}_{LA}(t, S_t(i))) \end{cases}$$

Replicate the mesh for N times, average the estimated option values, and then a better point estimate of the true option value can be given.

Random Number Generation

Simulate the underlying price……

$$S_t^i = S_0^i \exp\{(r - \delta_i - \frac{1}{2}\sigma_i^2)t + Z_t^i\} \qquad Z_t = C \cdot \underline{RNum}$$

Monte Carlo: pseudorandom series

Quasi Monte Carlo: low-discrepancy series (Sobol/Halton, etc)

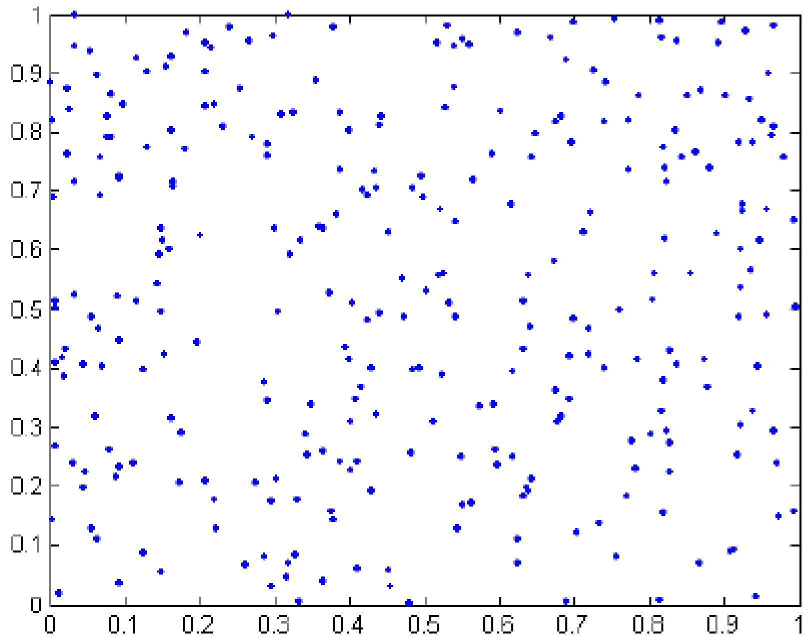
Inverse method:

Uniform Distribution → Standard Normal Distribution

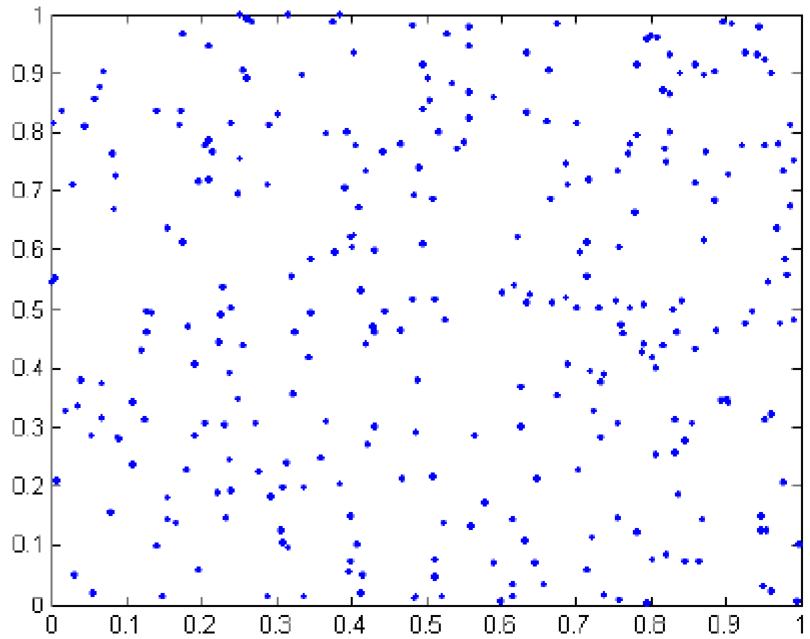
Random Number Generation

Pseudorandom series, 6-dim, 300 $O(1/\sqrt{N})$

Dim-2 Dim-3



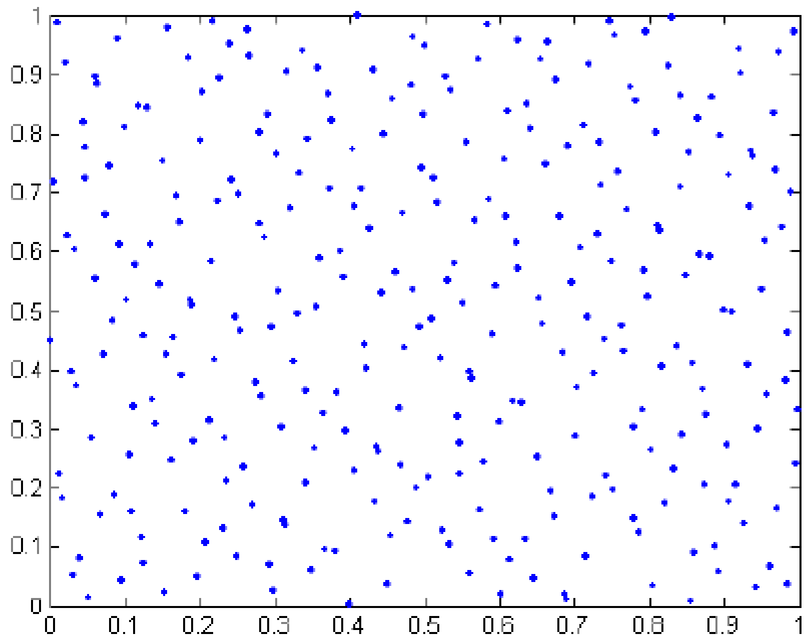
Dim-3 Dim-4



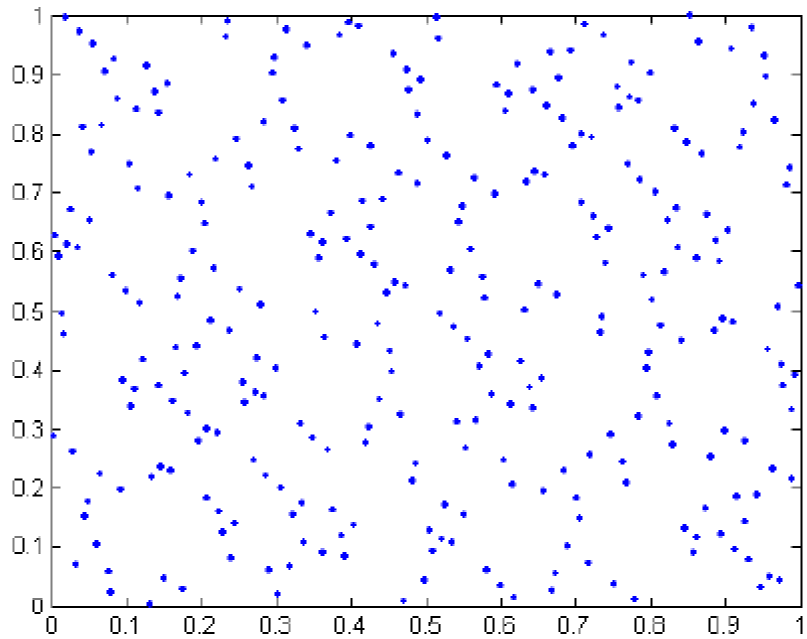
Random Number Generation

Low-discrepancy series, 6-dim, 300 $O(\log^m N / N)$

Dim-2 Dim-3



Dim-3 Dim-4

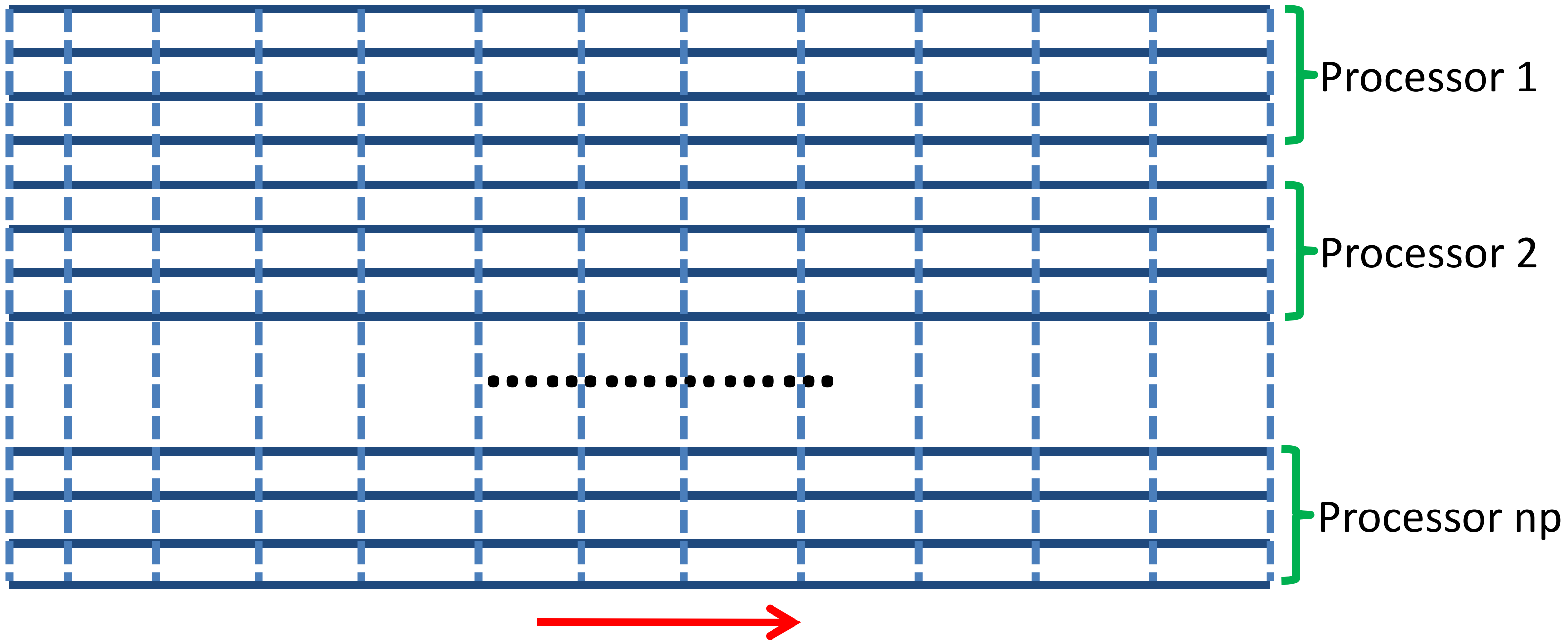


Parallelization

Suppose we use np processors, where $np = 2^x$, $x = 1, 2, \cdots$.

Construct the **whole** mesh on **each** processor respectively, so that each one has the full information of the mesh and the communication is avoid.

Split the mesh by **row**.



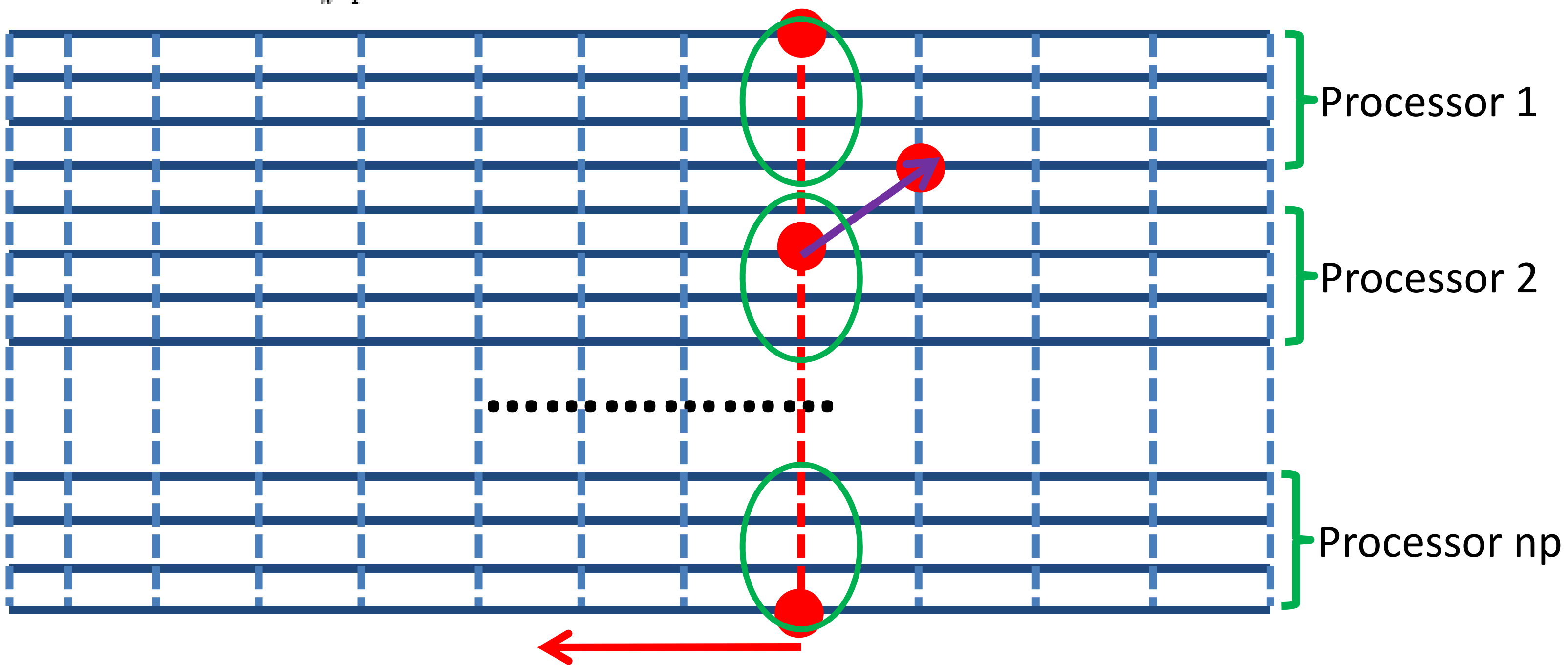
Parallelization

Weights calculation

$$w(t, S_t(i), S_{t+1}(j)) = \frac{f(S_t(i), S_{t+1}(j))}{\frac{1}{b}\sum_{k=1}^{b} f(S_t(k), S_{t+1}(j))}$$

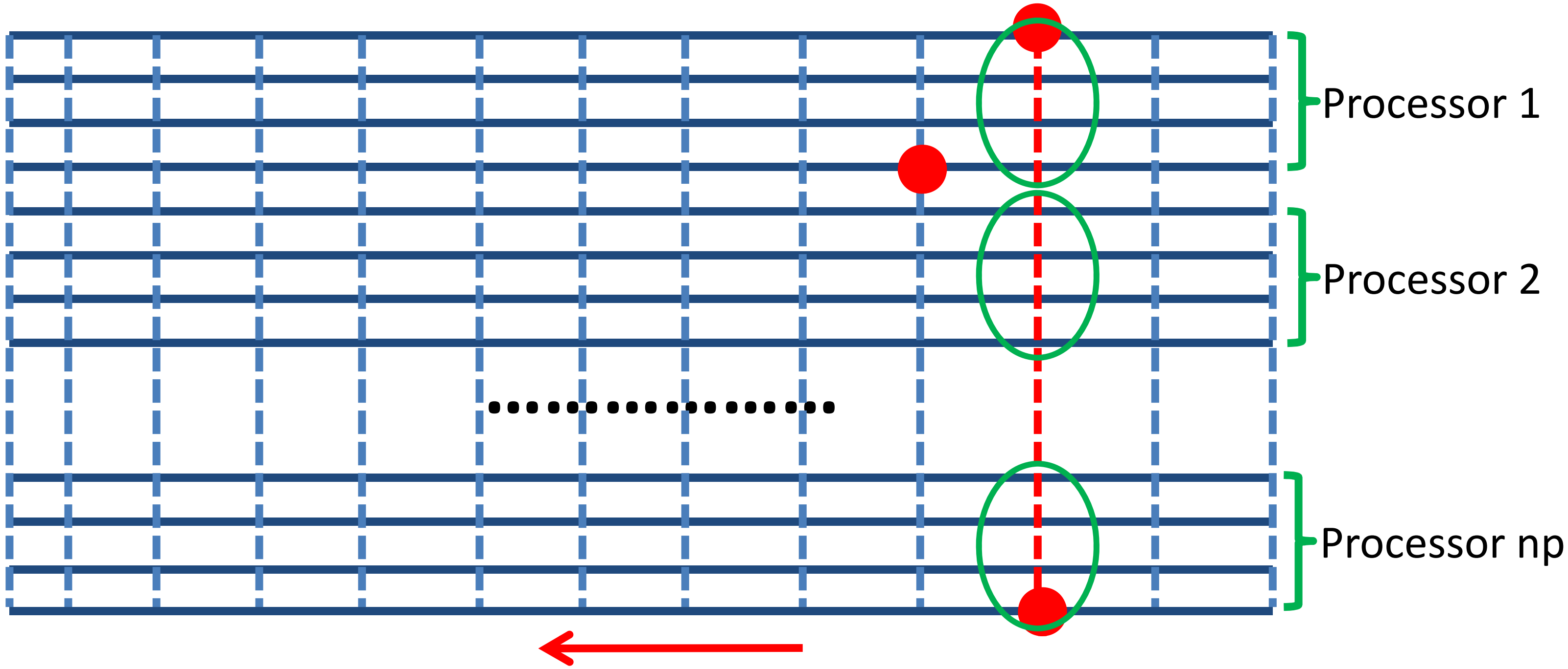
Processor p:

$$\sum_{k=pb\cdot p+1}^{pb\cdot(p+1)} f(S_t(k), S_{t+1}(j)) \qquad pb = \frac{b}{np}$$



Parallelization

$$
\begin{cases}
\hat{P}_{HA}(M, S_T(i)) = h(M, S_M(i)) \\
\hat{P}_{HA}(t, S_t(i)) = \max\{h(t, S_t(i)), e^{-r\Delta t}\dfrac{1}{b}\sum_{j=1}^{b}\boxed{\hat{P}(t+1, S_{t+1}(j))}\, w(t, S_t(i), S_t(j))\}
\end{cases}
$$



Parallelization-Implementation

Programming Language: C

Used MPI Functions:

MPI_Allreduce()/MPI_Bcast()/MPI_Datatype/……

Random Number: Sobol Sequences(low-discrepancy)

Numerical Experiment

Computational Platform: DeepComp 7000

- Supercomputing Center, CAS
- 106.5 Teraflops, 19th@Top500(Nov, 2008)
- Blade Nodes(Cluster) + SGI Nodes

System parameters of "Cluster" and "SGI"

parameter	**Cluster**	**SGI**
Operating System	Red Hat Enterprise Linux Server release 5.1	SUSE Linux 10SP2
Linux Kernel	2.6.18-53.e15	2.6.16.60-0.21.default
Complier MPI Library	Intel C/C++ Compiler 11.0.081 Intel MPI 3.2.011	Intel C/C++ Compiler 10.1.008 SGI MPT 1.2
Processor	Intel Xeon E5450, Quad-core, 3.00Ghz	Intel Itanium2 9140M, Dual-core, 1.66Ghz

Numerical Experiment

the symmetric case

$S_0^i = S$, $\delta_i = \delta_j$, $\sigma_i = \sigma_j$, $\rho_{ij} = \rho$, $i, j = 1, 2, \cdots, n$, $i \neq j$

(1) 5-Dim Case: American max call option on five assets

$n = 5$, $S = 100.00$, $K = 100.00$, $r = 0.05$, $\delta_i = 0.10$, $\sigma_i = 0.20$, $T = 3$, $i = 1, 2, \cdots, n$

(2) 7-Dim Case: American geometric average option on seven assets

$n = 7$, $S = 100.00$, $K = 100.00$, $r = 0.03$, $\delta_i = 0.05$, $\sigma_i = 0.40$, $T = 1$, $i = 1, 2, \cdots, n$

Use speedup to measure the performance of parallel algorithm and estimated bias to study the performance of stochastic mesh method.

- The speedup is the ratio of the serial computational time to the parallel consumed time using many processors;
- The estimated bias is calculated via “true” value subtracting point estimated value.

Numerical Study- Ⅰ

ρ = 0.0 in both cases; M = 3 in 5-Dim test case, while in 7-Dim test case M = 10.

In each test case, let the mesh size b equal to 1024, 2048 and 4096 respectively and replicate the mesh for 50 times. Then, execute the codes in both computing environments for each level of mesh size, using 1, 2, 4, 8, 16, 32 and 64 processors respectively.

Compute the “true” values of both options by setting b = 32768 , N = 50 and running the codes on “Cluster” using 128 CPUs. The computations cost 1003.86 and 4957.28 seconds respectively. The “true” values of both options are 25.224569 and 3.325708, with variances 0.000203 and 0.000079 respectively. The corresponding European options are 23.051029 and 2.419403 respectively, based on 100,000,000 Monte Carlo simulations.

np

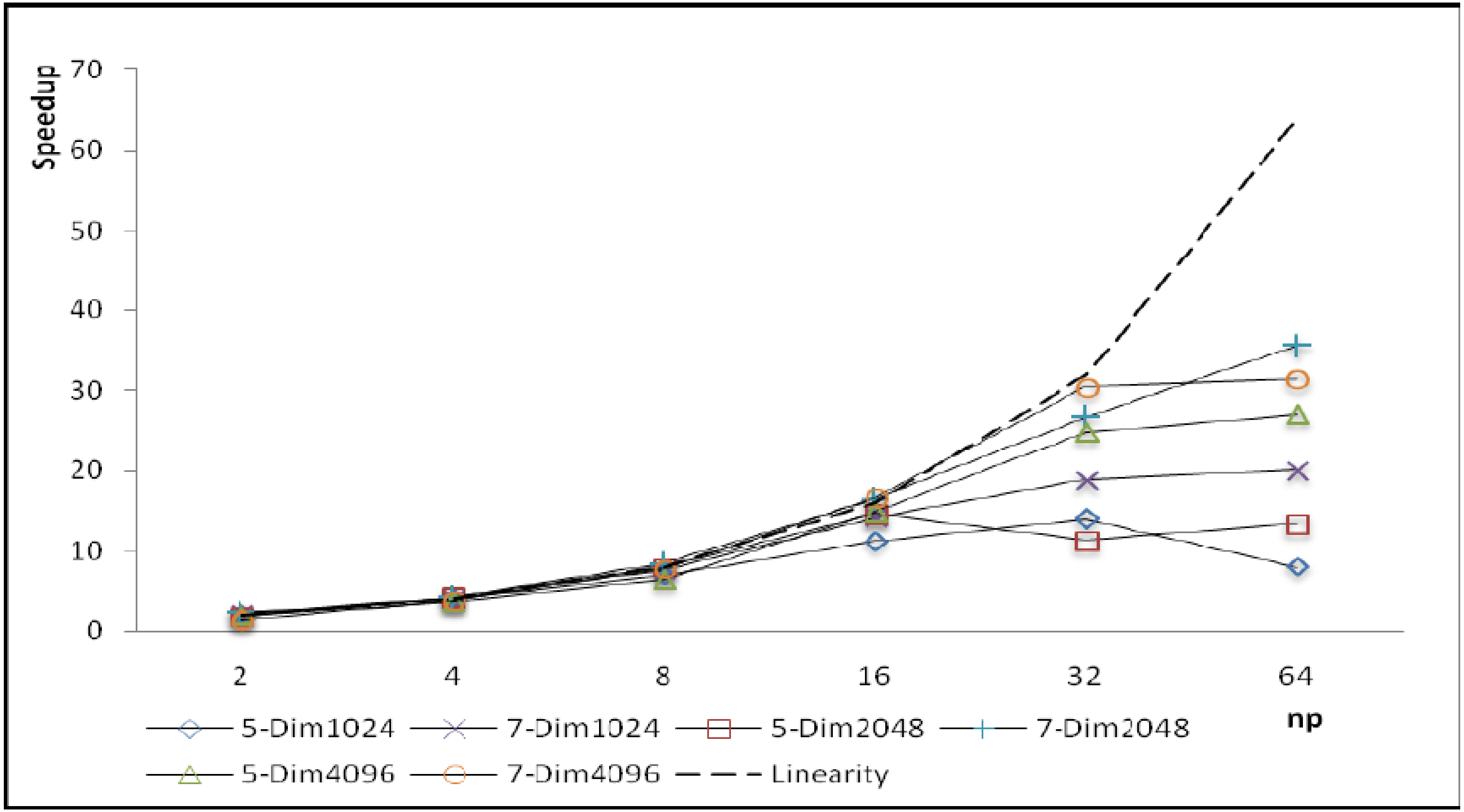
Numerical Study

Results in the standard 5-Dim and 7-Dim settings

n	b	Point Estimator	Estimated Bias	np						
				1	2	4	8	16	32	64
5	1024	25.028061 (0.068107)	0.196508	86.00	41.65	21.62	12.49	7.67	6.12	10.70
				26.00	12.35	6.15	3.35	2.67	2.13	2.31
	2048	25.073362 (0.013985)	0.151207	371.00	200.61	89.74	47.03	25.33	32.74	27.73
				185.00	95.63	46.76	24.54	10.79	5.30	6.13
	4096	25.103284 (0.005256)	0.121285	1490.00	780.85	416.41	233.47	100.27	60.10	54.99
				686.00	351.06	210.95	146.18	72.37	26.30	14.03
7	1024	3.179396 (0.002966)	0.146312	629.00	287.72	157.23	83.10	44.26	33.50	31.40
				181.00	88.19	67.40	26.82	18.66	10.62	16.52
	2048	3.261448 (0.001806)	0.06426	2624.00	1240.38	646.91	315.88	160.91	98.14	73.86
				980.00	525.83	349.72	168.27	63.58	31.85	27.24
	4096	3.299665 (0.001100)	0.026043	10581.00	7731.02	2851.46	1363.94	636.94	348.06	337.50
				4147.00	2271.78	1140.39	732.55	417.04	151.71	85.46

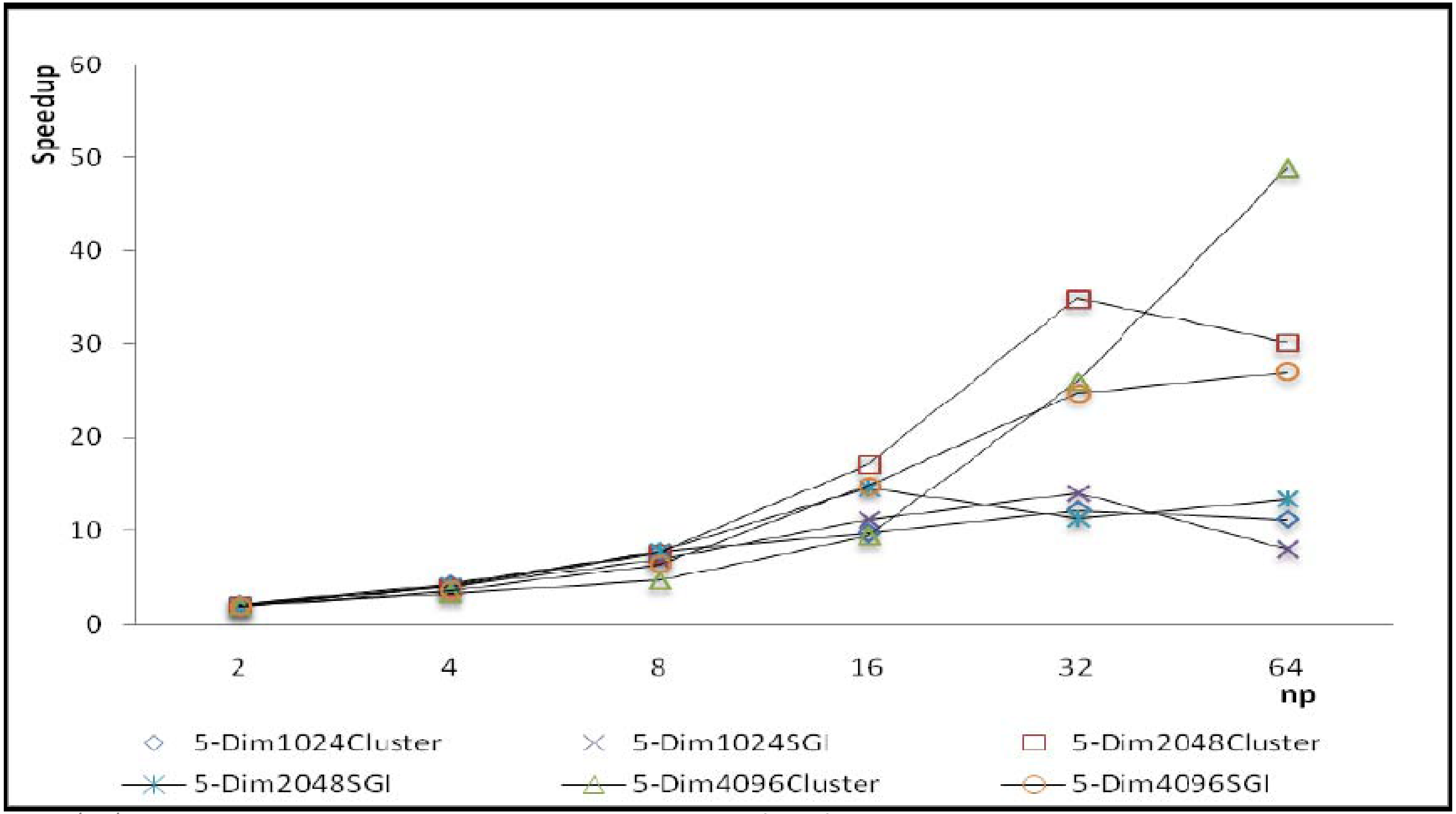
Numerical Study

Speedups of test cases in the standard settings on "SGI"



Numerical Study

Speedups of test cases on “SGI” and “Cluster”



Numerical Study-Ⅱ

5-Dim setting: **ρ = 0.5 and ρ = 0.7** respectively.

Set the mesh size to 4096 and replicate the mesh for 50 times.

compute the "true" value on "Cluster" using 128 CPUs, by setting b = 32768 , N = 50 . The "true" values are 18.647219 and 15.594348, with variance 0.000137 and 0.000143 respectively. The values of corresponding European options are 16.9774342 and 14.06876539 respectively, based on 100,000,000 Monte Carlo simulations.

ρ	Point Estimator	Estimated Bias	np						
			1	2	4	8	16	32	64
0.5	18.560954 (0.004301)	0.086265	1470.00	796.14	397.01	206.54	103.53	57.14	57.18
			665.00	387.24	199.67	134.95	70.66	26.85	14.65
0.7	15.517858 (0.002454)	0.076490	1470.00	780.71	412.58	240.62	116.23	66.78	44.43
			686.00	359.75	190.79	162.26	66.91	26.76	14.41

Numerical Study-III

7-Dim case with **different numbers of exercise dates**.

Set the number of exercise opportunities equal to 10, 20, 30, 40 and 50 respectively.

In each test case, b=4096, N=50.

Numerical Study

Results in the 7-Dim case with different exercise dates

Exercise Dates	Point Estimator	*np*						
		1	2	4	8	16	32	64
10	3.299665	10581.00	7731.02	2851.46	1363.94	636.94	348.06	337.50
	(0.001100)	4147.00	2271.78	1140.39	732.55	417.04	151.71	85.46
20	3.089466	-	11368.46	6078.62	3040.96	1379.79	742.55	538.96
	(0.000984)	8776.00	4780.91	2605.75	1781.82	956.36	374.65	183.78
30	2.833293	-	16301.30	10349.91	4650.35	2521.30	1237.52	981.32
	(0.000654)	14109.00	7753.29	3972.82	2926.89	1509.14	655.24	287.44
40	2.619965	-	-	12189.84	5963.58	2966.40	1672.48	1066.51
	(0.000614)	17980.00	9910.18	5526.19	3888.15	1619.10	655.05	371.75
50	2.446534	-	-	15246.06	7959.88	3649.26	1954.20	1316.79
	(0.000532)	-	12574.81	6527.18	4510.93	2015.47	998.87	510.66

Note: The "-" in the table represents that the consumed time exceeds 6 hours.

Numerical Study- Ⅳ

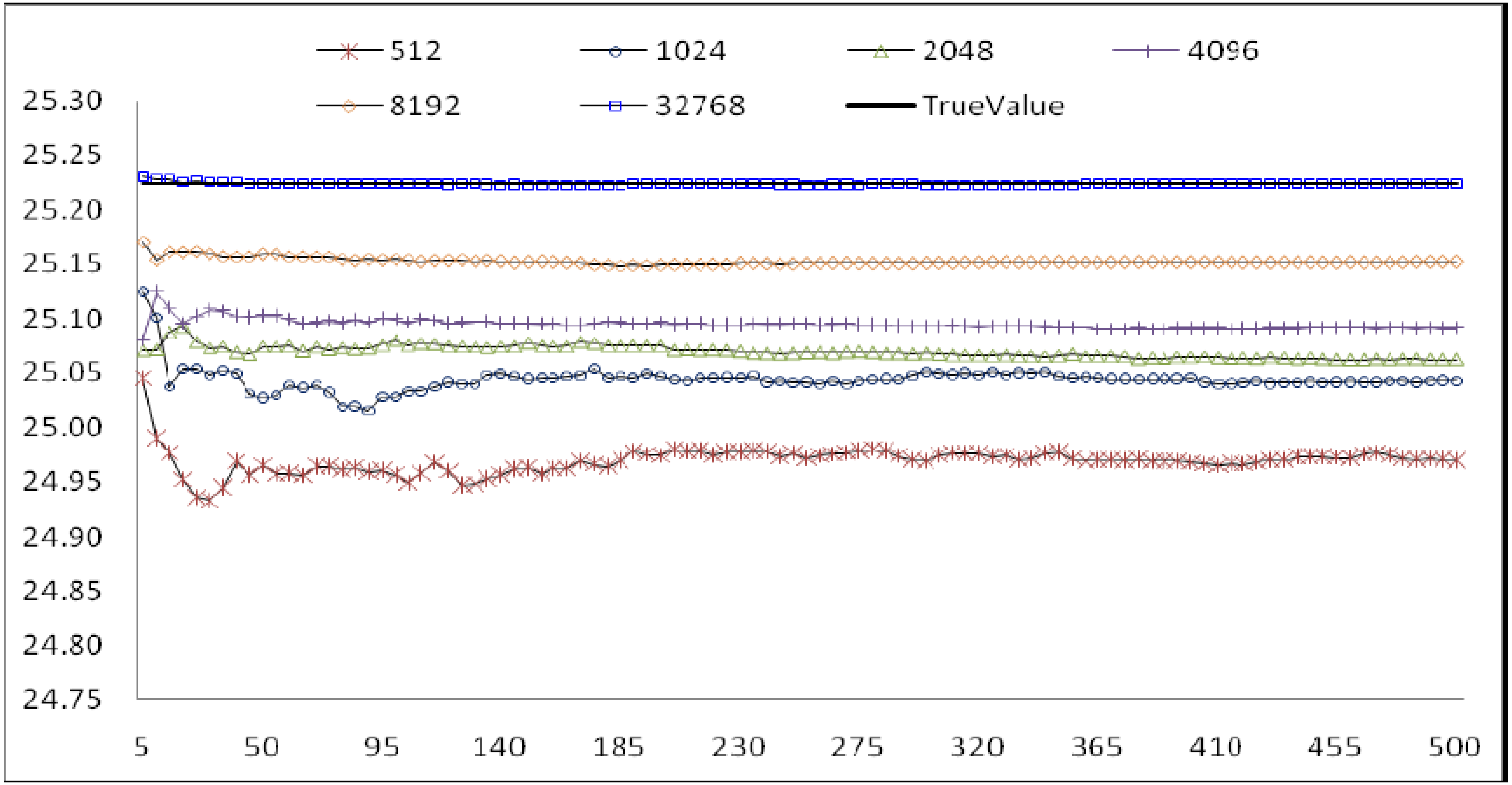
Observe the influence of **the number of mesh replication N** on estimated option values.

Use test cases with mesh size b = 512, 1024, 2048, 4096, 8192 and 32768 in 5-Dim setting.

Let N vary from 5 to 500 with step 5.

Numerical Study

Estimated option values with N from 5 to 500



Conclusions

- ✓ The parallel performance is much better for cases with large-scale simulated parameters and has good scalability in different parallel environments.

- ✓ The approach with bias reduction policy using smaller mesh size might **underestimate** the true option value.

Conclusions

- ✓ Our results can motivate the further performance enhancement of stochastic mesh method and then enable the practitioners to embrace this effective method.
- ✓ Parallel computing is an effective and efficient tool for computational finance research.

中国科学院超级计算中心
Supercomputing Center of Chinese Academy of Sciences

Thank you~~

Q & A, please