

リスク管理、トレーディング、システム構築
― 金融機関とHPC※技術 ―

2008/7/11「金融ALM／リスク管理モデルの運営と
コンピューティング・グリッド技術の応用」セミナー

※HPC: High Performance Computing ＝ 高速計算技術を意味する学術用語。スーパーコンピューティング。

NUMERICAL TECHNOLOGIES

ニューメリカルテクノロジーズ株式会社　鳥居 秀行

本日のアジェンダ

サブプライム危機の教訓

CITIBANKで何が起きたのか

1. リスク管理とHPC
- サブプライム危機に絡む最新の業界動向を整理
- グリッド化による大規模計算が解決の鍵になる！

BGM Swaptionモデル 1,000,000回モンテカルロがデスクトップPCで1秒未満！

2. トレーディングとHPC
- グリッド化は時代遅れか！？
- 欧米のクオンツも注目する高速化技術

本日実演

3. システム構築とHPC
- 「無駄なシステム投資」と言われないために
- システム予算削減とHPC技術

ベンダーの言いなりにならないで！優秀な理系学生はIT系に就職しない。

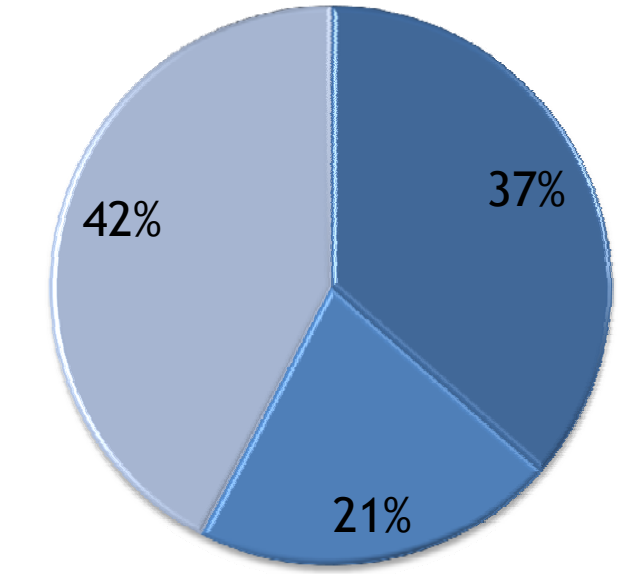
システム部員を学会に派遣せよ！グリッド化だけがHPCではない

付録：東工大との研究成果、メガバンクのALMシミュレーション

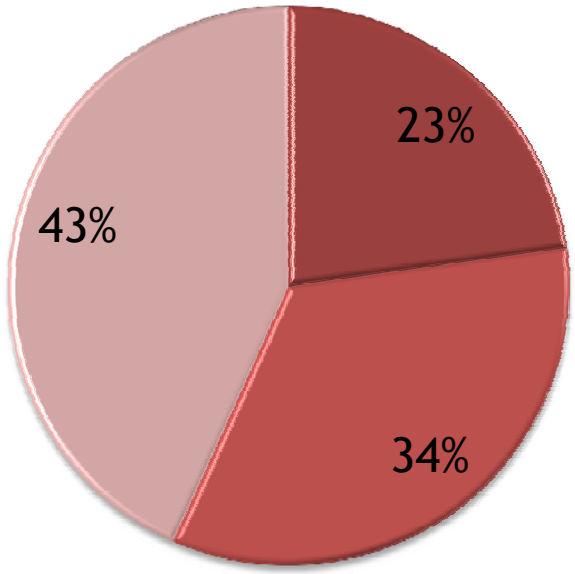
Copyright (C) Numerical Technologies Incorporated

弊社の社会的責任

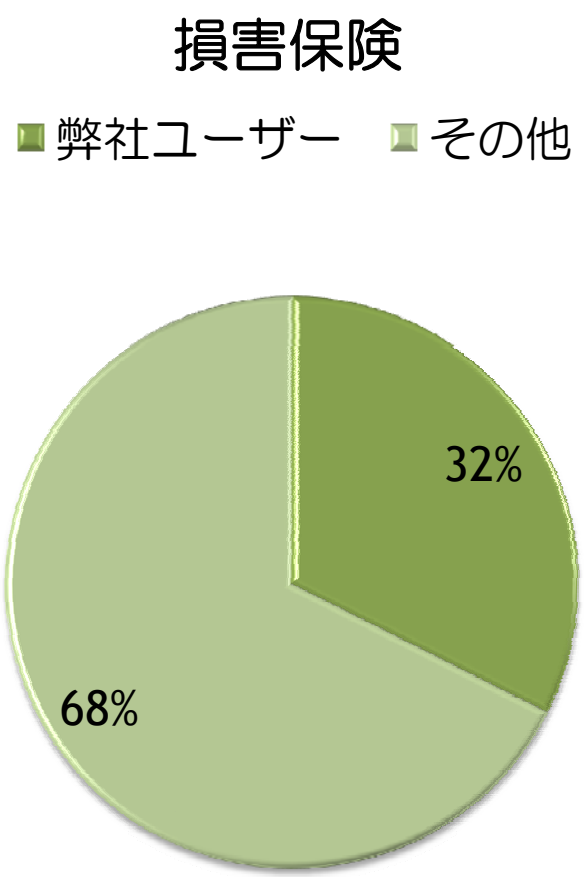
銀行



生命保険



Copyright (C) Numerical Technologies Incorporated



- 上位金融機関の多くが弊社製金融モデルの顧客
  - 日本のGDPに対する弊社ユーザーの総資産シェアは大きい。
- 我が国の金融業界に対して、弊社は相応の責任を負う
  - モデル構築についても節度が求められる。
  - 我々が得たイノベーションを顧客金融機関に還元する必要性。

Copyright (C) Numerical Technologies Incorporated 2008/7/11



1．リスク管理とHPC

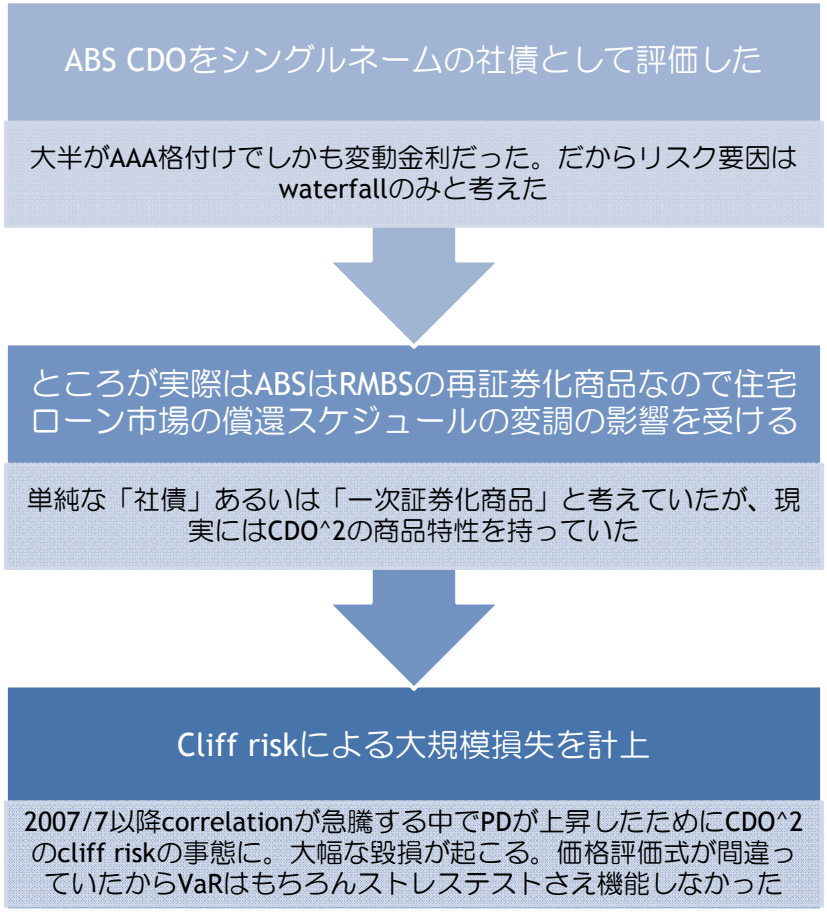
NUMERICAL TECHNOLOGIES

サブプライム危機の教訓（現在進行中）



Copyright (C) Numerical Technologies Incorporated

CITIBANKで何が起きたか



- CITIBANKの今次損失の過半をABS CDOが占める。
- CDOは、FAS157レベル3カテゴリーに属する内部格付けモデルを使い会計上も評価される金融商品。それまで「AAA格付け」と考えてノーマークだった。
- 結果的に正しくVaRを計算しておらず、ストレステストも機能しなかった。

[Source] Eduardo Eppelin/MD, Head of Market, Model & Counterparty Risk Analysis, CITIBANKの講演(2008/7/1)を私がまとめたもの

Copyright (C) Numerical Technologies Incorporated

Death of VaR？

- 米メリルリンチの2007/3QにおけるVaR（タイムホライズン=1day）の最大値は$92mil. 単純に63営業日毎日それだけの損失を出し続けたとすれば$5.8bil. しかし実際は$8.4bil.の損失だったから45%も過少見積もりしていたことになる。仮にルートTルールでVaRを換算したならば1000%以上の乖離幅。
- このようにサブプライム問題発生時における主要欧米金融機関のVaRは、メリルリンチに限らず信頼できないものになっている。
- それでBasel銀行監督委側には現行VaR修正の動きがある。

Copyright (C) Numerical Technologies Incorporated

Baselの反応（現在進行中）

- 証券化・再証券化スキームの見直しは不可避。
  - ABS CDOは事実上死に体。
- VaRの見直し。 Pillar 1関連では信用リスクとイベント性リスクを市場VaRフレームワークに織り込んでキャピタルチャージするアイデアをAIGTB（Accord Implementation Group on the Trading Book）が出している。
  - これに対してISDA他の業界側で対抗案を出す動きがある。
- Pillar 2関連ではストレステストの充実が課題になっている。
- 主要欧米銀に資金調達プレミアムがつく。緊急の問題である流動性リスク対応。市中協議文書を公表（2008年6月17日）。
  - 「健全な流動性リスク管理及びその監督のための諸原則」（原題： Principles for Sound Liquidity Risk Management and Supervision）。

Copyright (C) Numerical Technologies Incorporated

モデルリスク

- クオンツ的に言えば1-factor Gaussian Copulaモデルは死んだも同然。
  - 現在のストラクチャード商品のプライシングにおける主力モデル。
  - 単一のcorrelation parameterを使う。systemicあるいはidiosyncraticなdefault事象に対応できない。
  - 今回の問題はsystemicな事象。マーケットではcorrelationがどんどん上昇するのに対して、モデルのトラッキングエラーは拡大し続けている。
  - それでも業界が納得する代替案は今のところなく、プライスがわからない。
- モノラインによる信用補完問題（Jon Gregory/Barclays Capital）
  - ストラクチャード物の購入側金融機関は「高格付けで高利回りな商品」を求める。それは本来矛盾するから、AAA CDO Trancheにmonolineの保証をつけてleveraged super seniorとして販売してきた。
  - 「AAAに信用補完をつければAAA以上になるのか？」レバレッジ比率は通常10倍以上にもなる。考えてみればmonolineのキャピタルが、原債券の発行額ではなくレバレッジ後のCDOの発行額に応じて希薄化するのだから、ぜんぜん保証になっていない。
  - こうした現実にクオンツも目を瞑ってきたのである。しかもMBSブームの時に（20年前だ）同じことを経験した。だから非難されても当然と言えるだろう。

Copyright (C) Numerical Technologies Incorporated

伝統的なモデルリスク概念

- 2007年まではモデルリスクと言えば、典型的に以下の事例を指していた（と思う）。
  - 1987年：PIと複合ヘッジ戦略の失敗（各社で破綻、ダイナミックヘッジ戦略への過信と社債の価格形成に対する誤解）
  - 1994-5年：UBSのエプシュタインによるスワップヘッジの失敗（長短金利の相関を無視したためで当時「金利モデルの死」と言われた）
  - 1997年：同じくUBSのゴールドシュタインによる一連のミスプライシング（旧UBSの経営破綻の原因となりUBSをスイス銀との合併に追いやった）
- これらはすべて金融機関の内部要因であり、「リスク管理部署はトレーダーが使っているモデルを精査せよ」となって、実際そうしてきた。
- ところが、CITIBANKの事例で見たABS CDOのモデルリスクの件に加えて、新たに外部要因によるモデルリスクが発生…
  （次ページ）

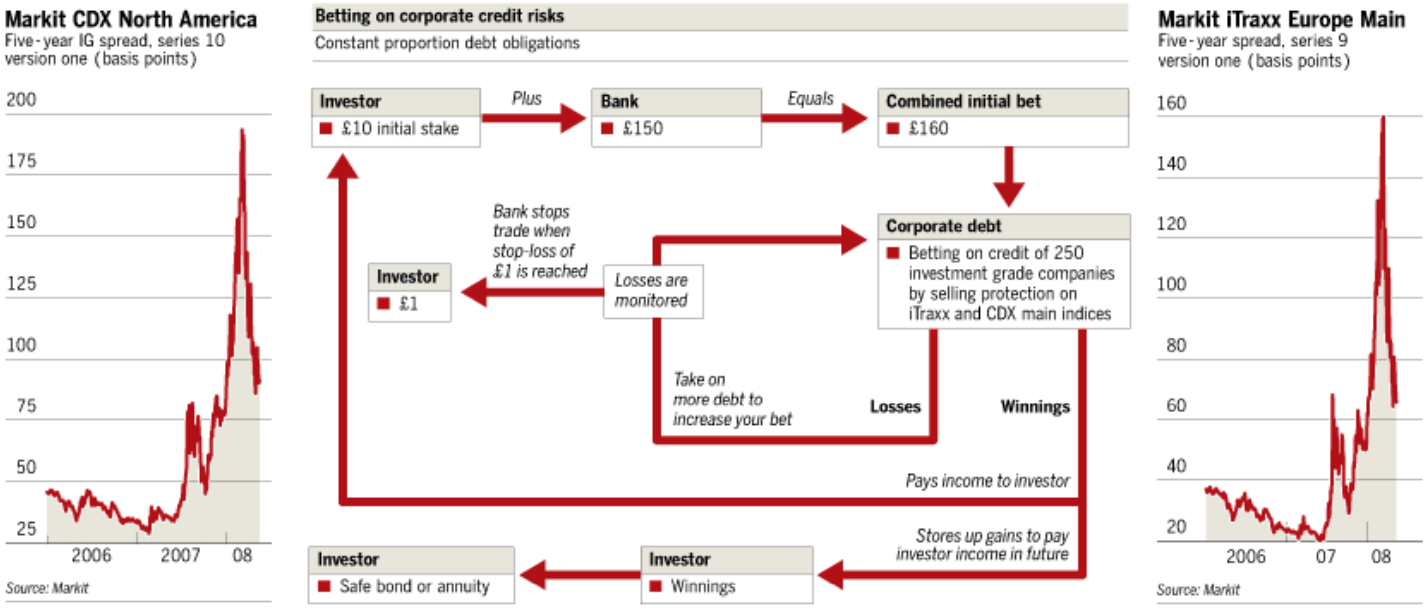
Copyright (C) Numerical Technologies Incorporated

Moody'sのCPDO格付けバグ

- Moody's error gave top ratings to debt products (Source: Financial Times, 2008/5/20)

  Moody's awarded incorrect triple-A ratings to billions of dollars worth of a type of complex debt product due to a bug in its computer models, a Financial Times investigation has discovered.
  …
  The results showed that early CPDOs might lose between 1.5 and 3.5 notches in the Moody's Metric, an internal measure, which equals up to four ratings notches.

- 下図はCPDOの仕組み (Source: 同上)



Copyright (C) Numerical Technologies Incorporated

新しいタイプのモデルリスク

- 世界最大の格付け会社Moody’sがCPDO格付けを最大4ノッチも間違えていた。
  - エンドユーザーが計算結果を常識を持って見ればわかる類の冗談のようなプログラムのバグ。プログラミングの間違いによる被害額としては1962年のマリナー１号事件を軽く抜き去る歴代ワースト１位のバグと言える。
  - しかもCPDOだけではない。より一般的なCDOやABSも…
- 2007年末に始まったMoody‘sとS&PのCDO、ABS大量大幅格下げ。
  - 今やCDOの過半が格下げになっておりAAA格も容赦なし3ノッチ以上も当たり前。そこで改めてMoody’sの公開文書を読めば格付け手法面でUBSのスワップヘッジ事件と同種で相関性をはじめとする見積もりが甘い。いや、むしろ誤っていたと言ってよいだろう。
  - 発行体寄り営業優先の経営方針がいけなかったと言われるが…
  - …それだけではないと思う。格付け会社も今日の投資銀行が組成している高度な数学モデルや複雑なスキームについていけてない。彼らは経験不足な上に賢くもないのだ。
- 格付け機関に外部監査を導入すべきだとする意見は金融業界にもある。またすでに米国議会など政治家が介入を始めている。EU内には欧州独自の格付け機関を作るべきだとする過激な意見もある。
- が、金融業界にとって格付け機関は運命共同体。格付け機関を否定すればBIS規制フレームワークも成り立たない。だから格付け機関も結局は生き残るのだろう。

Copyright (C) Numerical Technologies Incorporated

ストラクチャード物のリスク管理

- 弊社はこれまでCDOに関して次の２つの方法を提案してきた。
  - ①iTraxx、CDX等のインデックス商品として扱う。
  - ②バックアセットを入力してリスク管理モデル側でtranche管理しCDO価格を算出する。
- これに対する運用の実態。
  - 弊社の統合VARシステム（PortfolioBrowser）とALMシステム（Altitude）は①ができる。
  - ②の機能は弊社の信用VARシステム（CreditBrowser）のみが備えている。
  - ただし①に比べて②は遙かに運用が大変であり、バックアセットまで含めたフルバリュエーションによる再評価は計算量が膨大。ハードウェア増強も必要。このため実運用できている金融機関はほとんどない。
- CITIBANKの件を分析した通りであって①は今や不適。
  - そこで弊社ではALMシステム（Altitude）を含めてバックアセットのフルバリュエーション・トランチング機能を計画している。バックアセット評価に要する膨大な計算量への対応も必要だろう。
  - その一方で監督当局による規制動向を注視している。というのは我々が方法②を提案するのはあくまで金融機関による自主的な自衛手段。もし監督当局が方法②を強制したりすれば、現実には下位金融機関や一般顧客がCDOをはじめとするストラクチャード系商品の価格評価をするなど無理だから、ストラクチャード物自体の販売規制と同義である。あるいは90年代のP&G、ギブソン・グリーティング社の訴訟問題と同じ展開を懸念する。現在、そこまで各国の規制当局側が踏み込むことはないと予想して準備中。
  - 日本の金融当局の場合は規制不足よりも過剰規制の方が問題だろう。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

流動性リスク対応

流動性リスク発生の背景

- 主要欧米銀行の高収益性の秘密はハイレバレッジ。これは邦銀が無能なのが低収益の原因と考える日本経済新聞の記者でもない限り疑問はないと思う。恐ろしく預証率・預貸率が高いからアセットの証券化によって資金調達を図るスキームである。
- ところがストラクチャード物の市場が停滞してしまったために資金調達に支障を来した。
- レバレッジが災いして資本不足に。米欧の中央銀行もマーケットを通じて資金支援を継続中。

Baselの反応

- 流動性リスク対応の市中協議文書を公表（ 2008年6月17日）

日本への影響

- 今次危機ははっきり言って関係ない。むしろ資金の出し手であり、リスクの引き受け手。
- 我が国金融当局が国際会議への影響力を失っている現況。欧米相手の規制もとばっちりで国内金融機関まで回ってくる。ここが日本の金融危機の時とは異なる悲しい現実。
- とはいえ前向きに考えるならば、欧米で起きたAAA ABS CDO絡みのモデルリスク問題や、ストレステストについては国内金融機関でも問題であり損失も出ている。対応は急務であろう。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

焦点はストレステスト

- Basel市中協議文書では広範囲なストレステストの実施を求めている。

Principle 10: A bank should conduct stress tests on a regular basis for a variety of institution-specific and market-wide stress scenarios (individually and in combination) to identify sources of potential liquidity strain and to ensure that current exposures remain in accordance with a bank's established liquidity risk tolerance. A bank should use stress test outcomes to adjust its liquidity risk management strategies, policies, and positions and to develop effective contingency plans.

Copyright (C) Numerical Technologies Incorporated

伝統的なストレステスト概念

- これまでストレステストと言えば、Black Monday、北欧通貨危機、アジア通貨危機といった歴史的外的ショックを与えるヒストリカル・ストレステストが中心だった。市中協議文書ではヒストリカル・ストレステストも実施せよとしているが、限界も指摘している。

Principle 10 (Stress test)
… 97. History may serve as one guide when designing stress tests; however, historical events may not prove to be a good predictor of future events. A banker's judgment plays an important role in the design of stress tests. A bank should carefully consider the design of scenarios and the variety of shocks used. A bank should consider short-term and protracted, as well as institution-specific and market-wide, stress scenarios in its stress tests, including: a simultaneous drying up of market liquidity in several previously highly liquid markets; severe constraints in accessing secured and unsecured funding; restrictions on currency convertibility; and severe operational or settlement disruptions affecting one or more payment or settlement systems. Regardless of how strong its current liquidity situation appears to be, a bank should consider the potential impact of severe stress scenarios, and not discount severe scenarios as "implausible".

Copyright (C) Numerical Technologies Incorporated

Cannot predict next problem

- 「次の危機が何であるのか予測などできない」（2008/6/30 Stefan Walter/Secretary General, Basel Committee on Banking Supervision）
- つまりヒストリカル・ストレステストだけをやっていればよいのではない。

Copyright (C) Numerical Technologies Incorporated

パラメトリック・ストレステスト

- むしろパラメトリック・ストレステストを重視していく流れにあるようだ。当然と言えるだろう。
- 同様にヒストリカルVaRへの過信も流れが変わるだろう。

Principle 10 (Stress test)
… 96. In designing stress scenarios, the nature of the bank's business, activities and vulnerabilities should be taken into consideration so that the scenarios incorporate the major funding and market liquidity risks to which the bank is exposed. These include risks associated with its business activities, products (including complex financial instruments and off-balance sheet items) and funding sources. The defined scenarios should allow the bank to evaluate the potential adverse impact these factors can have on its liquidity position.

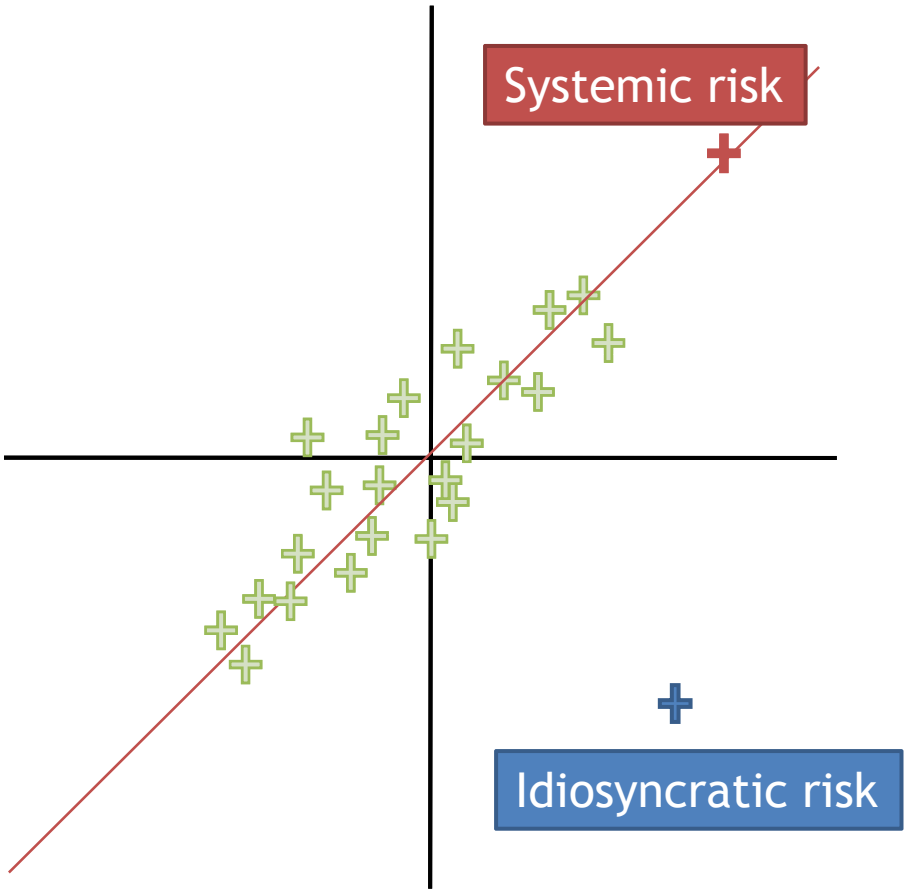
Copyright (C) Numerical Technologies Incorporated

統計的ストレステスト手法

- 統計的手法
  - High CL
    - 説明を要しないと思う。
  - Expected Shortfall（CVaR、tail VaR）の採用
    - サブプライム危機は極端なtailの事象だった。たとえば3σのOTM Putと同じく3σのOTM Callを使ってショートストラングルを作ったとする。従来のVaRならばリスクはゼロ（わずかなベガ・セータ要因は除く）。しかしExpected Shortfallならばリスクを捕捉できる。もちろん問題はシミュレーション回数の増加による計算量の増大。
  - Correlation stress
    - 次ページ

Copyright (C) Numerical Technologies Incorporated

Correlation Stress手法



- 例えば左図のような相関関係があったとして…
- 相関を強調してsystemic riskとする。
- 同様に相関を弱める側に作用させてidiosyncratic riskとする。
- この手法の問題はコンピュータの助けがいること。

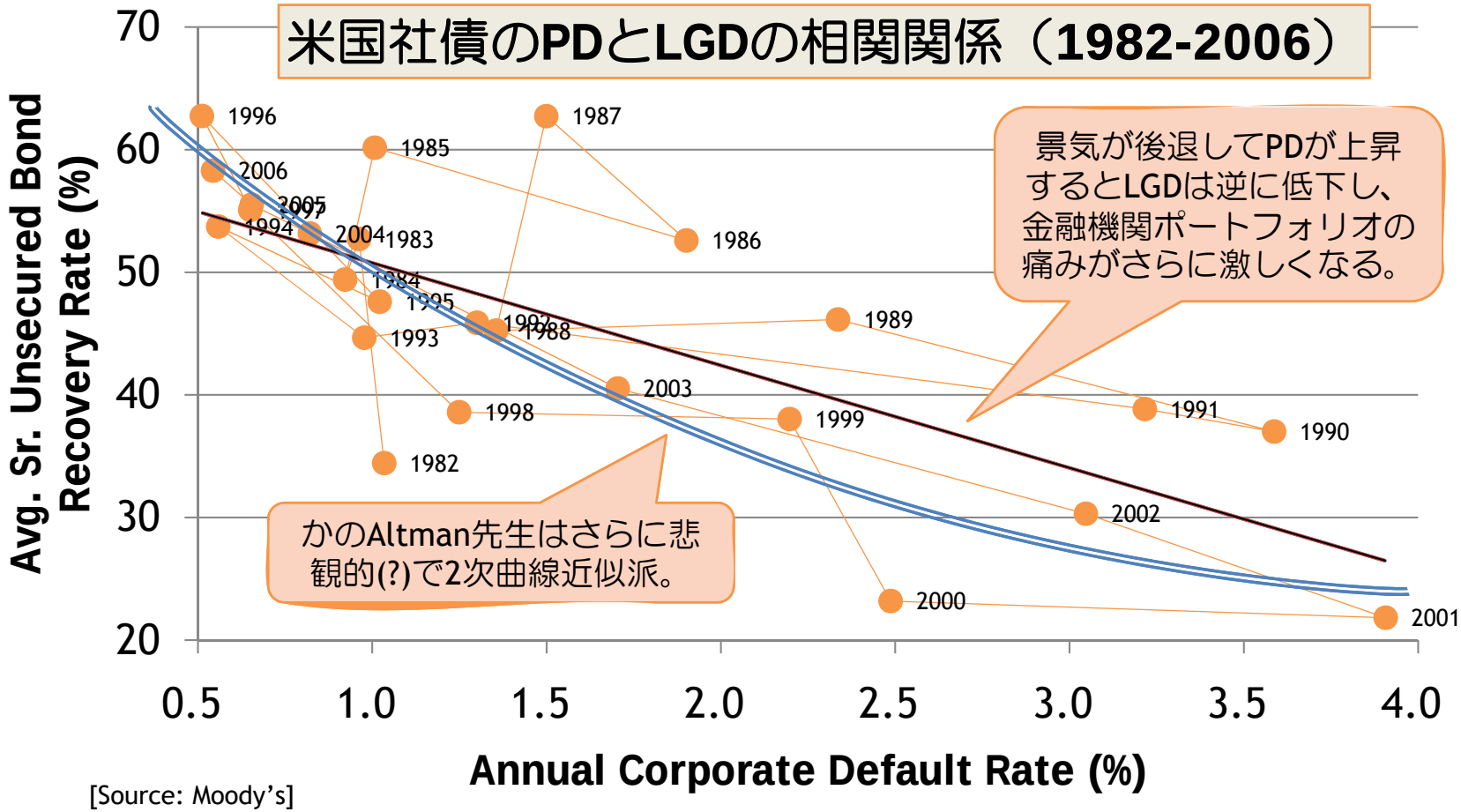
Copyright (C) Numerical Technologies Incorporated 2008/7/11

非統計的手法（What-Ifシナリオ）

- 結局一番有効な手法だろう。ただしサブプライム危機回避を目的としたならば必要要件は厳しい。
  - 金融工学的な商品の理解
  - トレーディングの経験
  - トレーダーとの日常的接触
  - P/Lの追跡
  - システムの理解
  - マーケットと市場参加者の関連性の理解
- ベーススプレッド、CDSスプレッド、ボラティリティスプレッドなどリスクファクターへの感覚は机上学問では身につかない。
- RMBSの償還スケジュールに疑義が生じた時、LIBOR調達が困難になるとあなたは想像できたか？ 一連の動きを頭の中でシミュレートするには豊富な知識と相当な能力がいる。今次危機を回避した投資銀行がある。リスクマネジャーの真骨頂と言えるだろう。
- これが次の金融危機でもできるか！？ この手法の問題点はシステムではなくて人材の確保と権限の委譲。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

Downturn LGD問題



[Source: Moody's]

Copyright (C) Numerical Technologies Incorporated 2008/7/11

PDとLGDは相関する

- サブプライム問題がreal economyにまで影響を始めた。
- Basel IIは、内部格付手法に用いるLGDに対して景気後退期の状況（Economic Downturn Conditions）を反映するように求めている（LGDに関してはPITよりもTTCを支持？）。
- 景気が後退してPDが上昇するとLGDは逆に低下し、金融機関ポートフォリオの痛みがさらに激しくなる。我が国でも経験した通り。
- 今回、米国が実際にresession入りしつつあるが、久々の景気後退期入りとあって米銀ポートフォリオのLGDには景気良好期に対しキャリブレートされた疑いがかかっている。簡単に言えばEconomic Capitalが過小だということ。
- ゆえに、ビジネスサイクル次第では非常に厳しい未来が想定される。逆に不況期に対してLGDがキャリブレートされた我が国にとってdownturn LGD問題は基本的に「対岸の火事」。
- ただしBasel当局側が動いていることを考えれば、日本の金融機関もパラメータが適切か点検するに越したことはない。

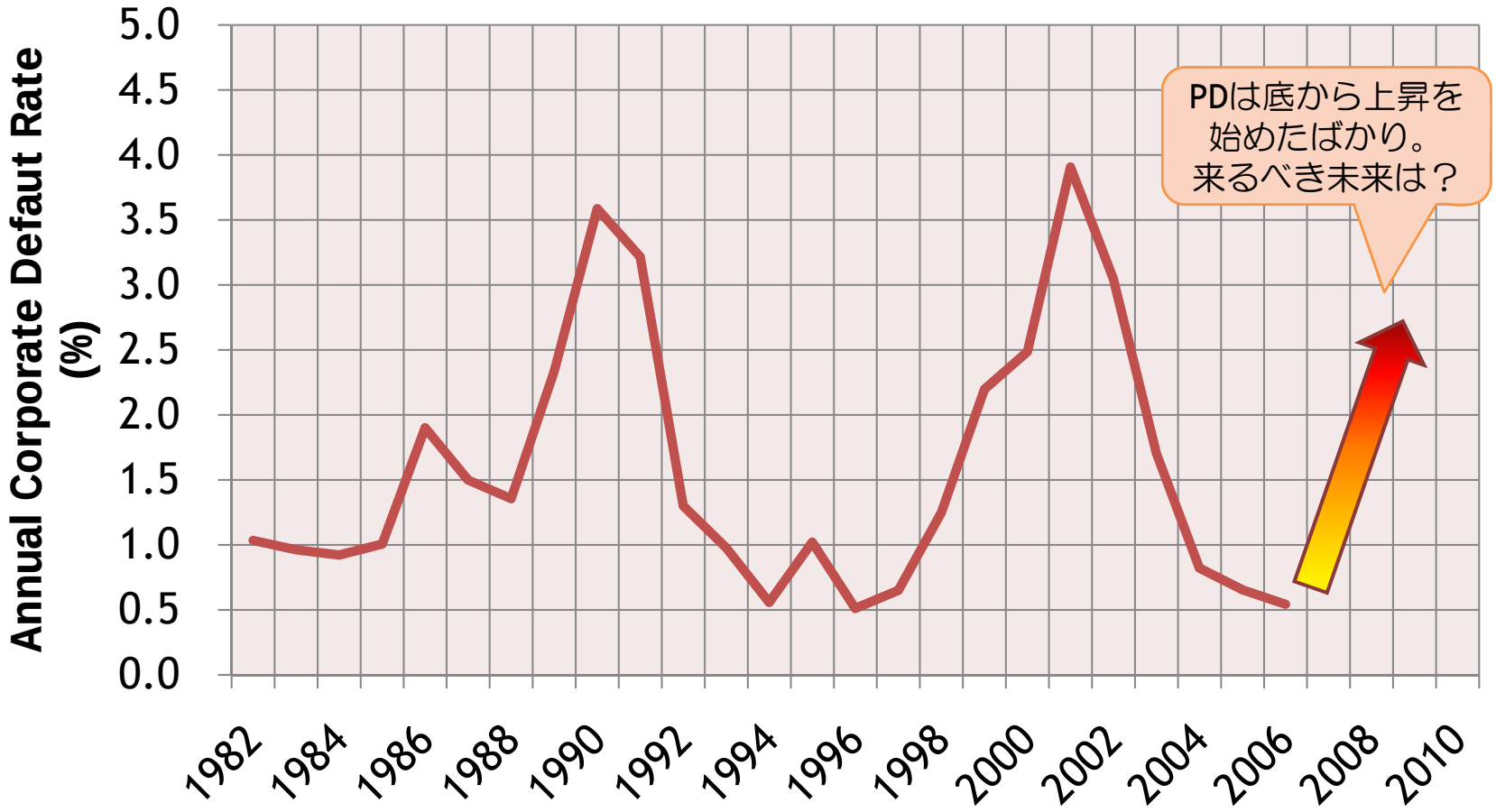
Copyright (C) Numerical Technologies Incorporated

ECモデルを検証しない米商銀

- 「（米国の）金融機関の多くはEconomic Capitalを算出する手段として、Moody's KMVモデルを、自行のポートフォリオに適合しているか検証することなく使用している。KMVが提供する相関係数はwholesaleポートフォリオに対してキャリブレートされたもの。銀行にとって適正なEconomic Capitalになっている保証はなく注意を要する」（2008/6/30 Fang Du/Senior Project Manager, Banking Supervison & Regulation, FRB）。
- CITIBANKで起きたモデルリスクだけが問題ではない。Goldman Sachsを除く米銀のリスク管理は我々の想像以上に杜撰のようだ。
  - 今後もさらなる資本不足問題としてつけが回りそう。
  - もし出資するならば入念なデューディリが必要。
- まるでバブル期の邦銀のようで耳が痛い話。

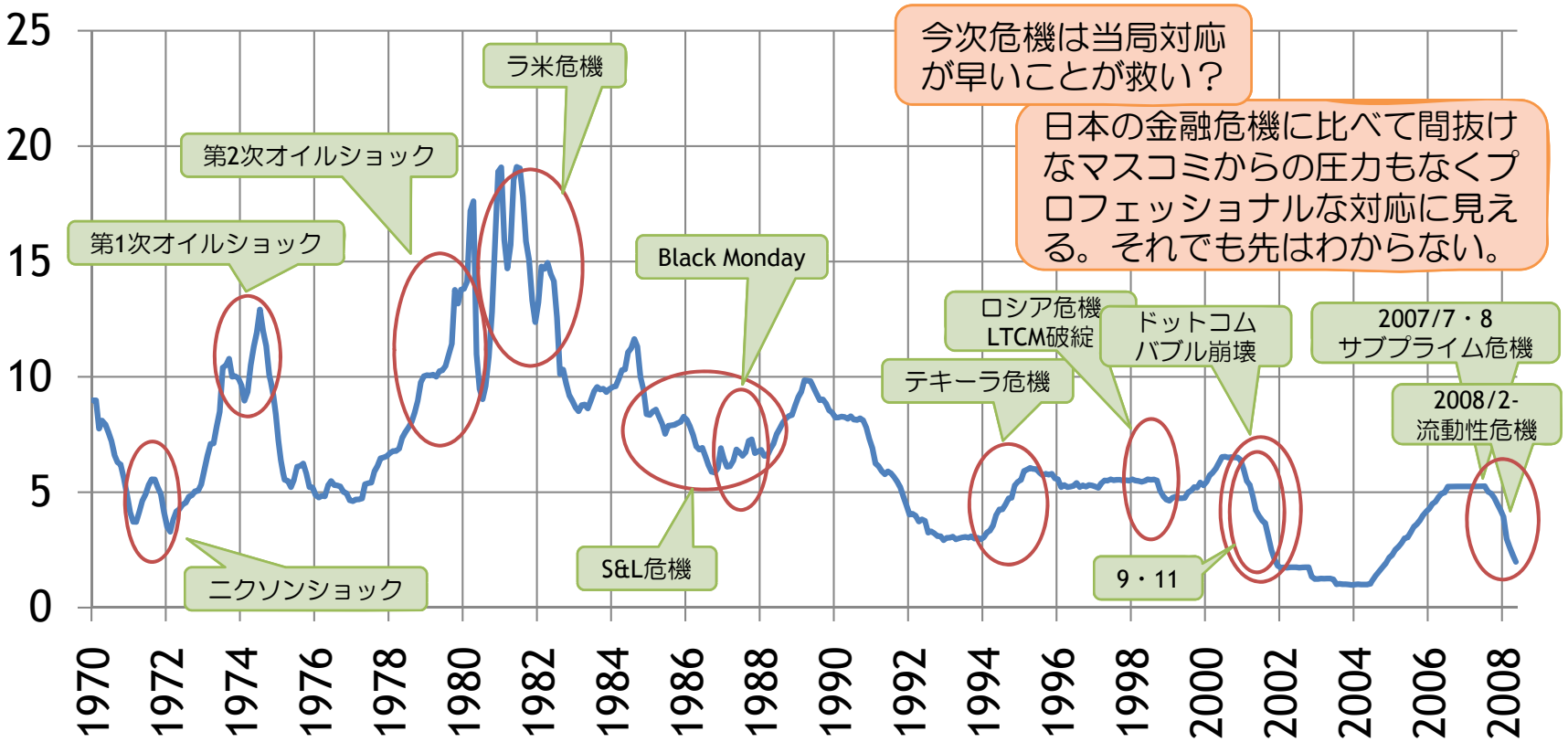
Copyright (C) Numerical Technologies Incorporated

米銀の資本不足問題の行方…



NUMERICAL TECHNOLOGIES Copyright (C) Numerical Technologies Incorporated 2008/7/11

FFレートの推移（1970/1-2008/6）



NUMERICAL TECHNOLOGIES Copyright (C) Numerical Technologies Incorporated 2008/7/11

嵐に備えよ



Copyright (C) Numerical Technologies Incorporated

シニアマネジメントの関与

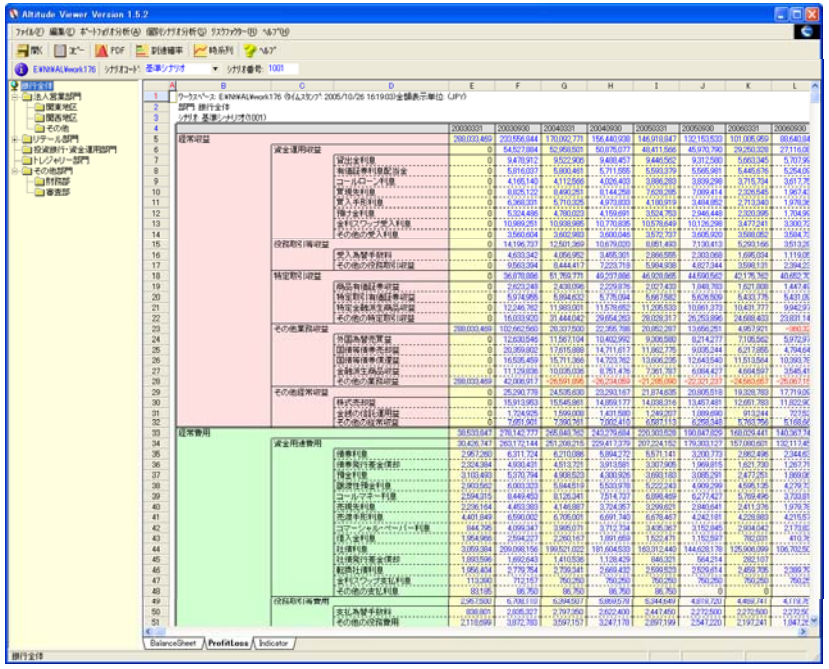
- Baselはシニアマネジメントの関与を強く求める。投資銀行ならばともかく、商業銀行では言うのは簡単、実現は大変。

Principle 3 (Senior management)
… 17. Senior management should closely monitor current trends and potential market developments that may present significant, unprecedented and complex challenges for managing liquidity risk so that they can make appropriate and timely changes to the liquidity strategy as needed. Senior management should define the specific procedures and approvals necessary for exceptions to policies and limits, including the escalation procedures and follow-up actions to be taken for breaches of limits. Senior management should ensure that stress tests, contingency funding plans and liquidity cushions are effective and appropriate for the bank, as discussed in later principles.

Copyright (C) Numerical Technologies Incorporated

経営陣向けには財務計数に換算して説明しないと無理ではないか

■ ALMシステムAltitudeの財務計数画面



- 日本の金融機関の場合、あるいは欧米でも商業銀行ならば、ボスは市場のプロとは限らない。ポジションシートやVaRからリスク体力を観察せよと経営陣に迫るのは酷だろう。
- 一方で、一般にマネジメント層の財務分析能力は下層のスタッフよりも上である。トップダウンに経営を見ているのだから。
- 弊社ではALMシステムに限ってながら、仮想的な1年や10営業日の単なるNPVやVaRではなく、カレンダーベース決算期末の財務諸表や推移表あるいはそのVaR/EaR概念として計数表示することにしている。

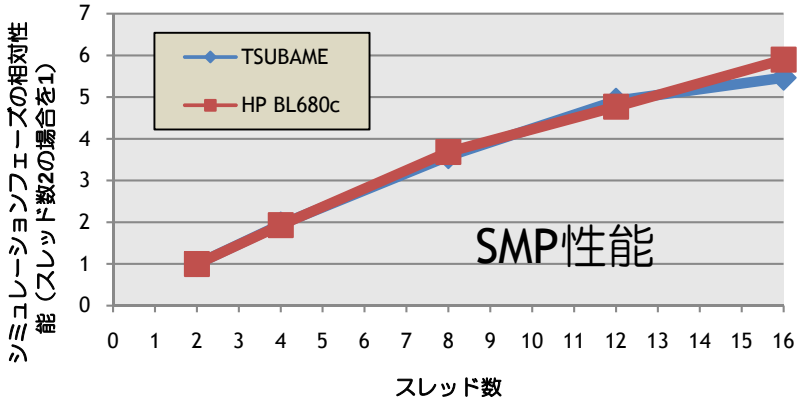
Copyright (C) Numerical Technologies Incorporated 2008/7/11

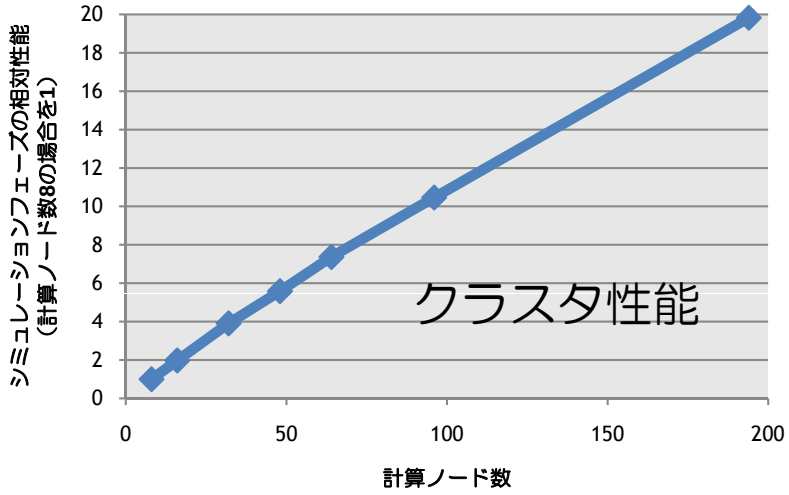
HPCを試す理由

- システム要件としてまとめれば…
  - バックアセットを含めた個別件別にまで分解しなければCITIBANKと同じ間違いとなり、リスク集中は計測できない。つまり膨大な件数を処理する計算能力が必要。
  - 1-factor Gaussianなどの簡易モデルを排除し、systemic defaultをシミュレーションしてストレステストするとなれば自前のトランチングが必要。計算量が増大する。
  - シニアマネジメントの関与、特にストレステストの理解を考えればVaRだけではなく財務計数を出したい。VaRならばNPV 1個を計算すればよいが、財務計数となると数千個に及ぶ財務科目を処理する計算量が必要。
- こうした理由から、サブプライムを踏まえた次のBasel規制の修正を睨みつつ、大規模データを使ってグリッド技術をテストしている（東工大での研究）。

Copyright (C) Numerical Technologies Incorporated

その研究成果

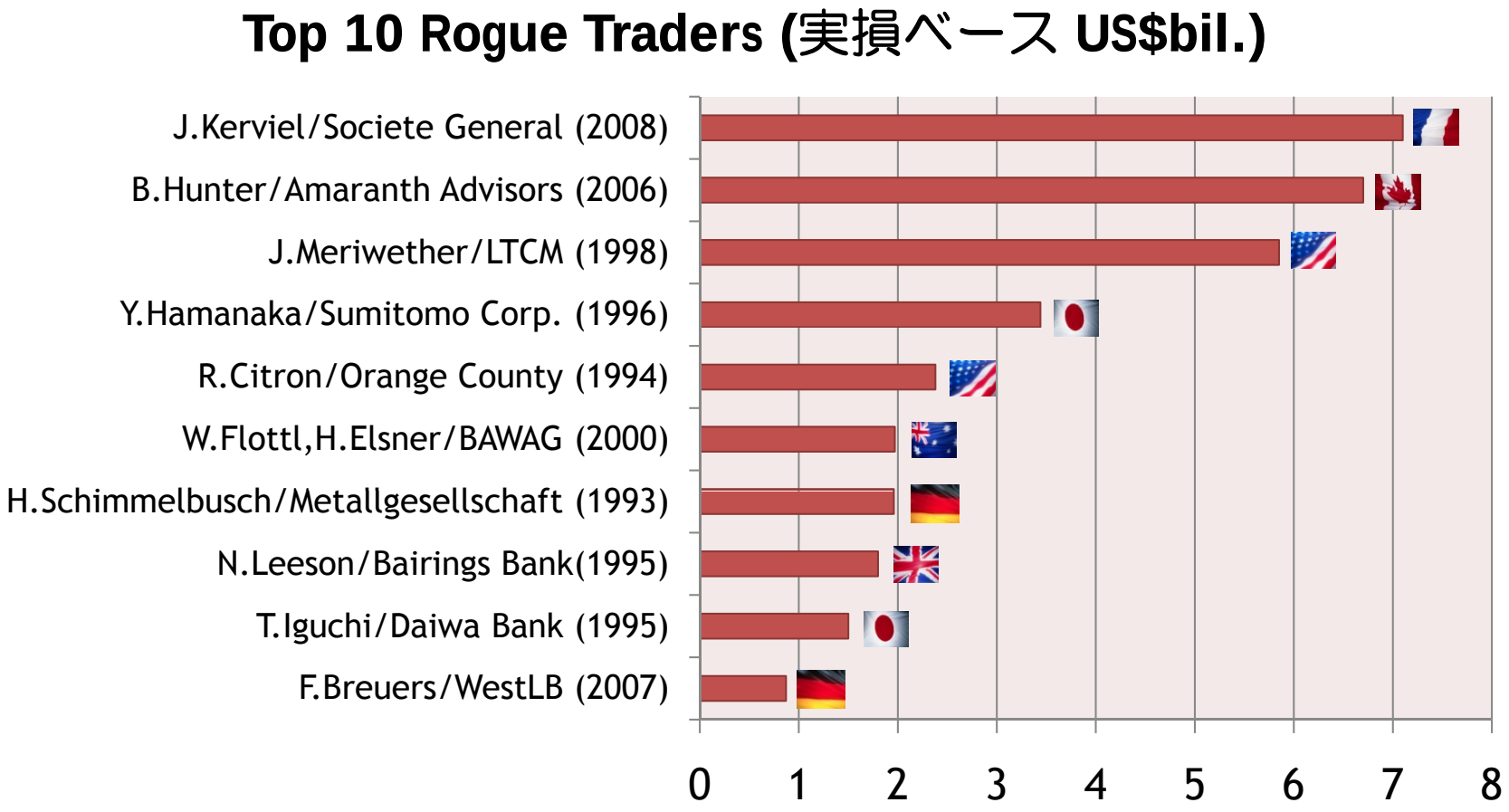




- グリッド環境でハードウェアを増やせば増やすだけ性能が上がることを確認。
  - 詳細は本資料末尾に添付。
- 結局、投資予算の問題であって技術的障害はない。
  - 最大級のメガバンクが仕様を満たす市場・信用リスクの統合管理を行うとしても、現時点の市中調達価格で10～20億円のハードウェアで済むだろう。
  - もちろん金融機関規模が小さくなるほど、機能を落とすほど、要求費用は小さくなる。
- コンピュータの並列化が進んでいるから所要コストは漸次低下する。僅か数年で計算量はリスク管理の障害でなくなる。もっとも規制要求もさらに高度化するかもしれないが。
- ただし、この種の投資を行っても、ヒューマンファクターのリスクは防ぎようがない（次ページ）。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

Rogue Traders



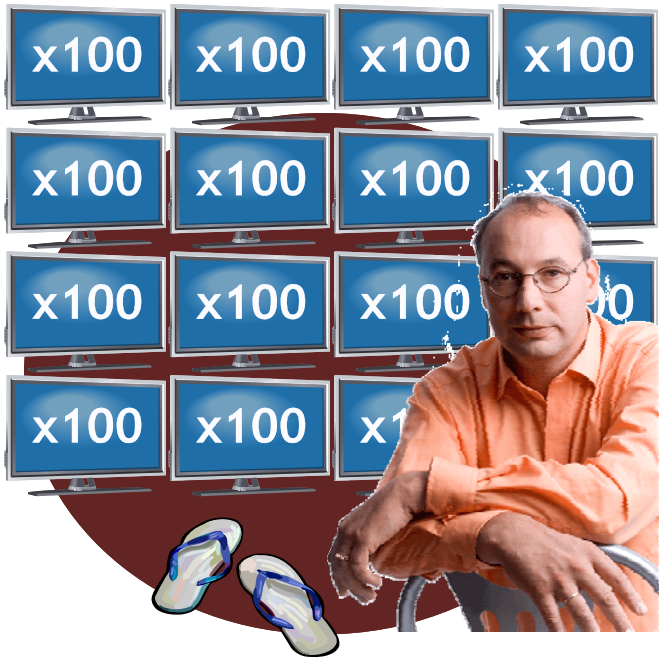
Copyright (C) Numerical Technologies Incorporated 2008/7/11



2. トレーディングとHPC

稼ぐのはどちらだ？





- あなたがリラティブバリュー・アービトラージなり統計アービトラージのトレーダーであるとしよう。
- 左はグリッドコンピューティング技術を使って通常の100倍の速度で個別株ペアトレードの組み合わせを発見し取引を自動執行するトレーダーだ。彼の後ろにいるのは彼専属の忠実なるシステム要員だ。
- 右は伝説のマイケル・ゴールドフィールド（誰でも適当なトレーダーの名前を入れて！）ばりに4x4スクリーンを前にしたアーブだ。スクリーンのひとつひとつが左のグリッドコンピューティング環境と同じ100倍の能力を持っている。彼は金利、エクィティ、コモディティ、思いつくままにリアルタイムでクロスマーケットの裁定機会を計算させトレードを執行する。
- どちらが勝つか？　右に決まっている！！！

Copyright (C) Numerical Technologies Incorporated

- あなたがJAFEEで発表して満足するだけのタイプではないのなら、将来はヘッジファンドへの転職を目指しているのならば…
- 次の話題にきっと興味を持つと思う。



他に足りないのは若き日の三菱の渡辺、住友の近藤・小西型のボスだけだ！

Copyright (C) Numerical Technologies Incorporated 2008/7/11

あなた専用のHPC環境

- デスクトップPCの計算能力を100倍以上に増強。
- パソコン1台がグリッド環境に匹敵。
- プログラミングはクオンツであるあなたが自分でやればよい。大体、人にやらせたらあなたのアービトラージのノウハウが流出してしまうではないか。技術面がわからなければ私自身が講習会を開いてあげよう。



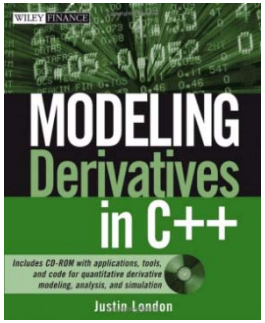
＝



Copyright (C) Numerical Technologies Incorporated

例題：BGMモデルを使ってヨーロピアンスワップション価格を求めよ

- 金利系で計算が大変なデリバティブ商品の例としてBGM[1997]※に基づくEuropean swaptionの価格計算を選び、素直にモンテカルロ法を使って解いてみた。
- 本事例はあくまで計算デモであって、金融工学の研究としてのインプリケーションはまったくない（そんな意図は私にない）。むしろ敢えて皆様の手で追試できるように実装方法はJustin London, “Moodeling Derivatives in C++,” John Wiley & Sons, p.652に出来るだけ忠実に従った（バグや理論的におかしな箇所のみ修正した）。
- 金利系ではなく為替系トレーダーの方であれば、例えばバミューダン・オプションの価格モデルは最悪モンテカルロを必要とするという意味で同じクラスの計算規模に属するから、以下の事例を適宜読み替えて頂きたい。



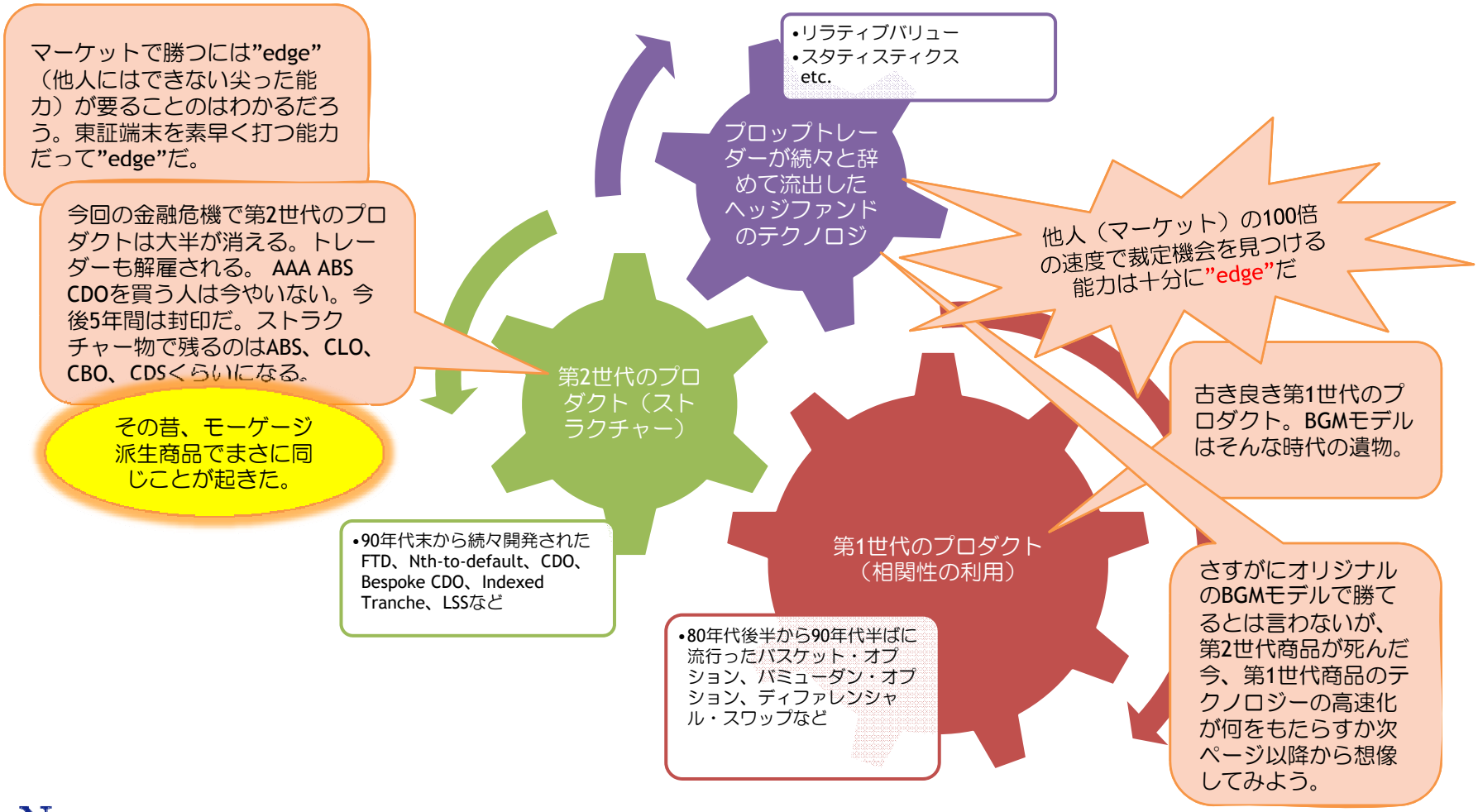
Copyright (C) Numerical Technologies Incorporated

BGMモデルに関する簡単な解説

- BGMモデル（= Brace, Gatarek, and Musiela model、別名lognormal forward-LIBOR modelあるいはLFMモデル）は、LIBORを原資産とする対数正規型連続型イールドカーブモデル。Blackの式によりcap価格を計算する。BGMには類似品Jamishidian[1997]モデル（LSM, lognormal forward-swap model）がある。 **BGMモデルは人気のある金利モデル**である。
- 人気の秘訣は、マーケットで値付けされているLIBORカーブとインプライドボラティリティを入力するだけで無裁定条件を満たせること。オリジナルのHJM[1992]モデル（BGMやLSMはHJMの派生物）やHW[1990]にみられる、「スポットカーブやフォワードカーブに合わせるfittingが必要でBloombergスクリーンだけではプライスが決まらない」とか「市場から観測できないパラメータが必要」なんてことはない。つまり**BGMモデルはシンプルで実用性に優れている**。
- BGMモデルの欠点は、モデルの複雑さが原因で金融工学屋の得意技である格子法(tree)による数値解法が使えないことだろう。 BGMモデルの効率的解法に関するペーパーが山のように出ているが、無理やり簡略化するのでなければモンテカルロシミュレーションを使ってBGMモデルを解くしかない。つまり**BGMモデルは計算に時間がかかる**。
- とはいえ苦労するだけの御利益はあって、BGMモデルを使えばcap、floor、swaption、そして他の金融派生商品まで単一のモデルで記述できる。つまり裁定機会を発見可能。だから**relative value tradeで稼ぐヘッジファンドや金融機関のプロップトレーダーにとってはBGMモデルは標準装備（少し古いが）**である。
- 次ページ以降の参考事例では、市場で観測したインプライドボラティリティに合わせる操作（キャリブレーション）は省略し、あらかじめ与えられているものとした。これは他意があってのことではなくて、単に皆様の手で追試できるように実装方法をJustin London, “Moodeling Derivatives in C++,” John Wiley & Sons, p.652に合わせて**客観性・再現性を備えた計算速度テストとして本事例を用意**したかったからである。
- なお、この本は”tons of bug”と言われる問題作品で、理論に沿って修正しなければプログラムが意味をなさないし、そもそも動作しない。またあらかじめこの本の原プログラムにあるループ内push_back操作のような非効率な実装はあらかじめ取り除き、**C++言語レベルのオプティマイズを済ませてから速度比較を行った**。そうしないと「1000倍も高速化した」ように見えてしまい誤解を生み本事例研究への信頼が損なわれるからである。
- 後述する必須のハードウェアが本セミナーの直前まで出荷されなかったから、**本例題の解法に使った作業期間は僅か1週間**。多少のバグは残っているかもしれない。しかしながら結論は不変のはず。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

“edge”をつくれ



Copyright (C) Numerical Technologies Incorporated 2008/7/11

例題の解法

- LFMを使ってLIBORのセットを求める。

$$dF_k(t) = \sigma_k(t)F_k(t)\sum_{j=n+1}^{k}\frac{\rho_{k,j}\tau_j\sigma_j(t)F_j(t)}{1+\tau_j F_j(t)}dt + \sigma_k(t)F_k(t)dz_k(t)$$

(注) 収束性を試す過程でMが非常に大きくなるので、ここでオリジナルのran1の代わりにMersenne Twisterに変更した。またLIBORカーブにマルチコを起こさない範囲での収束性改善策として対称変量法を試した。以上が原プログラムと異なる。

- オイラー法を使い上式を差分方程式に変換。

$$d\ln F_k^{\Delta t}(t+\Delta t) = \ln F_k^{\Delta t}(t) + \sigma_k(t)\sum_{j=n+1}^{k}\frac{\rho_{k,j}\tau_j\sigma_j(t)F_k^{\Delta t}(t)}{1+\tau_j F_j^{\Delta t}(t)}\Delta t - \frac{\sigma_k^2(t)}{2}\Delta t + \sigma_k(t)\Delta z_k$$

- ここまで準備してN個のtime step毎に求めたLIBORからpayoffを計算する。

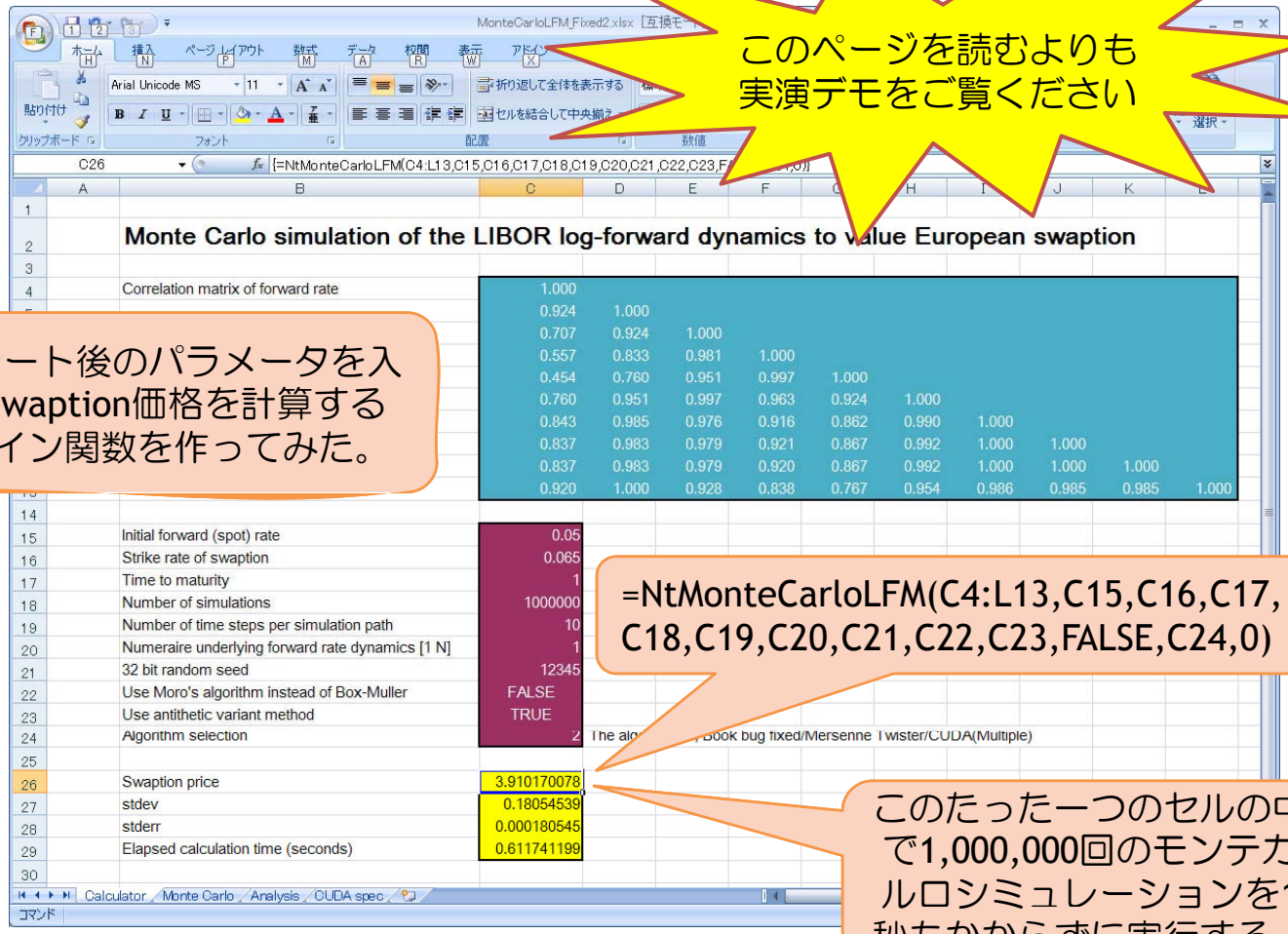
$$\max(S_{n,N}(T_n)-K,0)\sum_{i=n+1}^{N}\tau_i P(T_n,T_i)$$

- 以上の過程をM回繰り返して（モンテカルロ法）期待値をとる。

$$swaption\ price = P(0,T_n)E^{Q^{P(\cdot,T_n)}}\left(\max(S_{n,N}(T_n)-K,0)\sum_{i=n+1}^{N}\tau_i P(T_n,T_i)\right)$$

Copyright (C) Numerical Technologies Incorporated 2008/7/11

European Swaptionカリキュレータfor Excel（当社製技術サンプル：非売品）



このページを読むよりも
実演デモをご覧ください

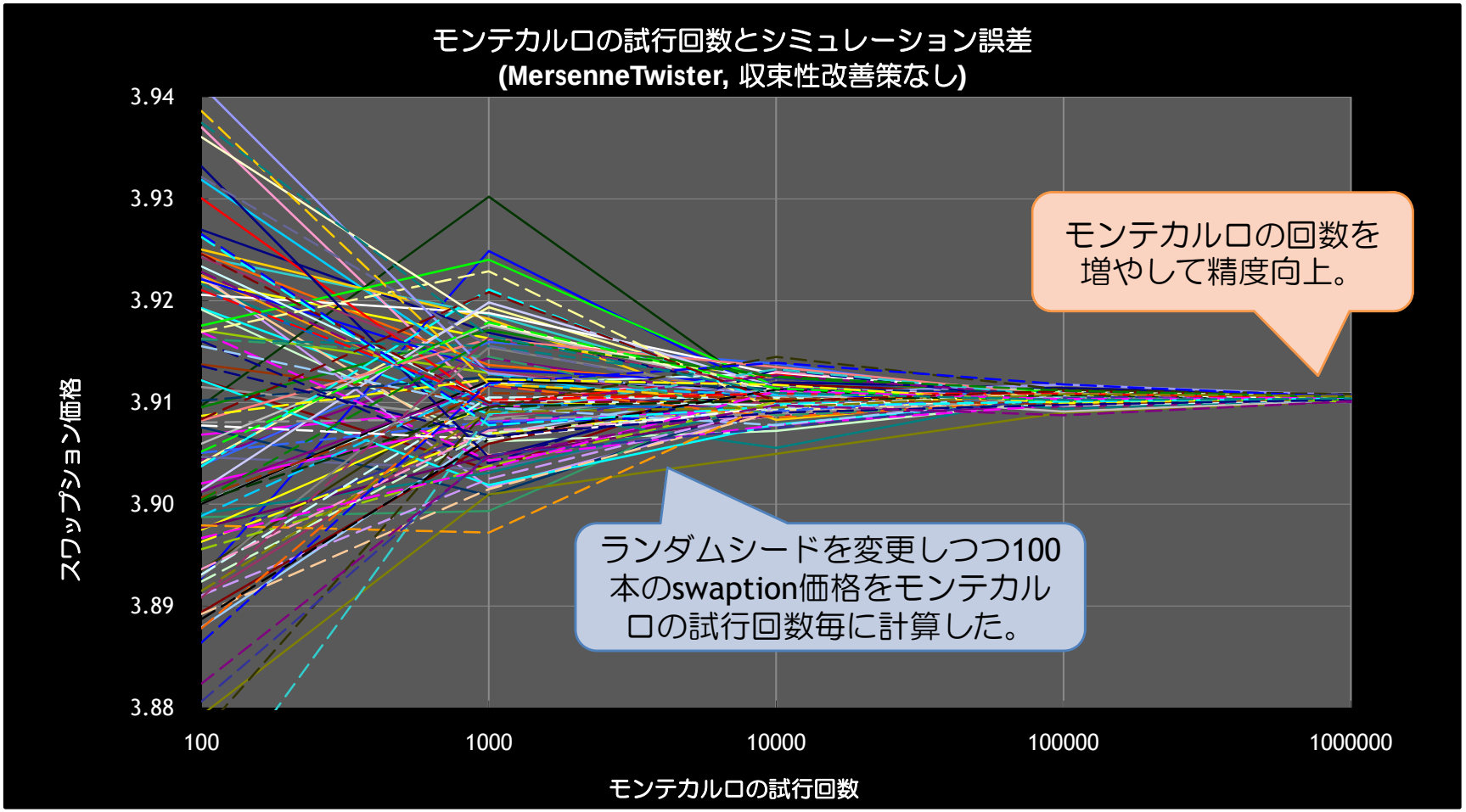
キャリブレート後のパラメータを入力としてSwaption価格を計算するExcelアドイン関数を作ってみた。

=NtMonteCarloLFM(C4:L13,C15,C16,C17,C18,C19,C20,C21,C22,C23,FALSE,C24,0)

このたった一つのセルの中で1,000,000回のモンテカルロシミュレーションを1秒もかからずに実行する。

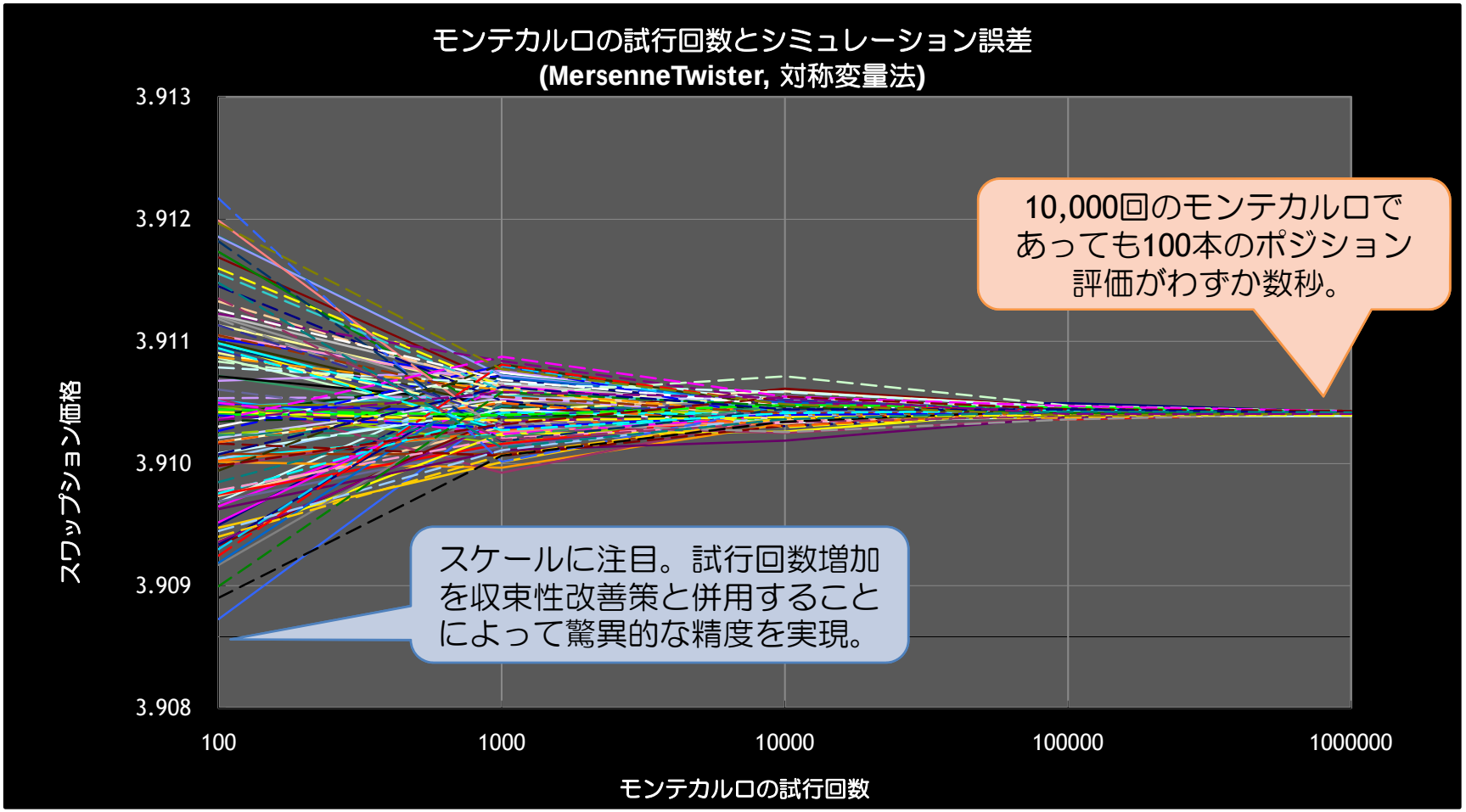
NUMERICAL TECHNOLOGIES Copyright (C) Numerical Technologies Incorporated 2008/7/11

175倍速の計算能力があれば大量計算も思いのまま



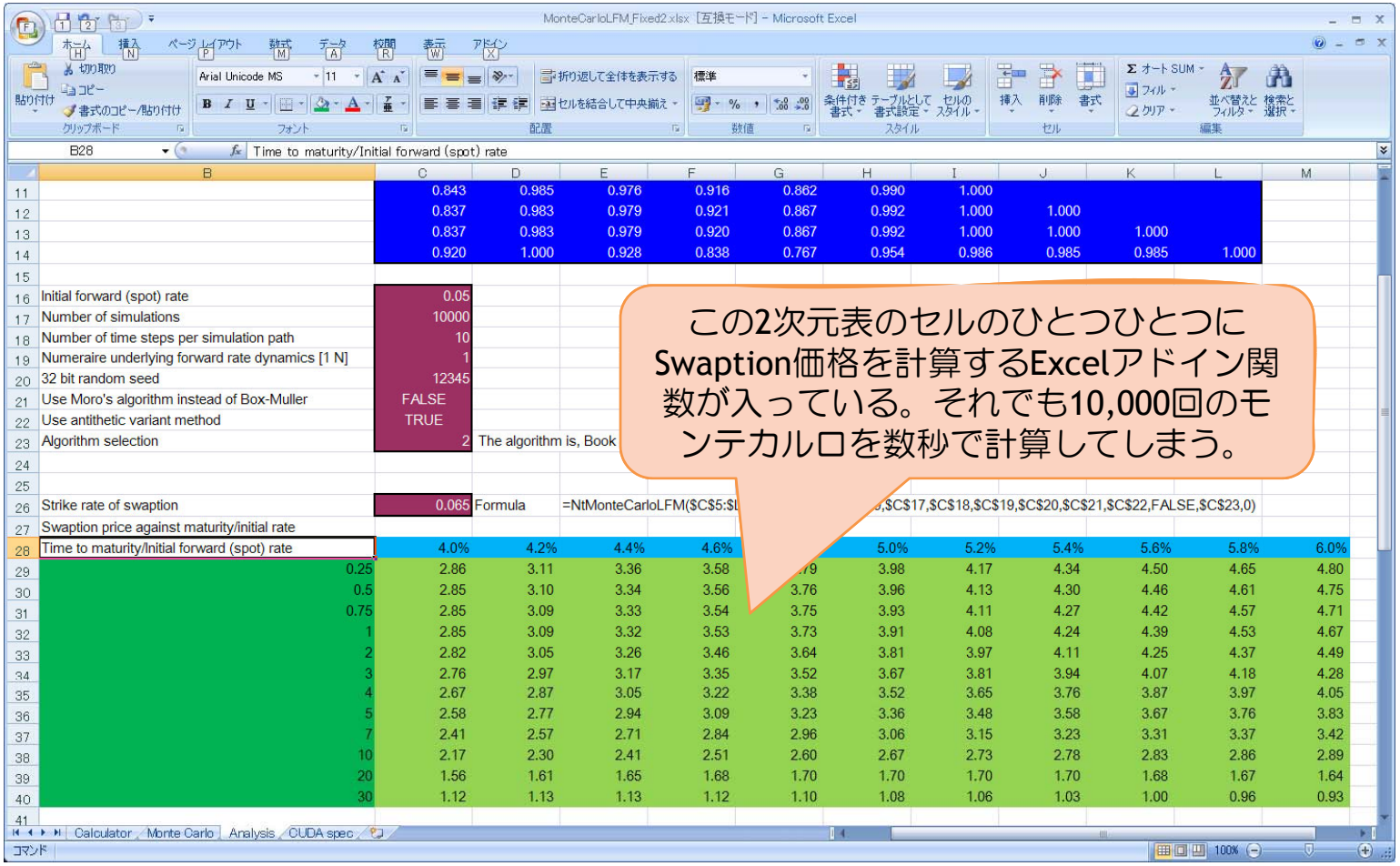
Copyright (C) Numerical Technologies Incorporated 2008/7/11

175倍速の計算能力があればポジション数が増えても大丈夫



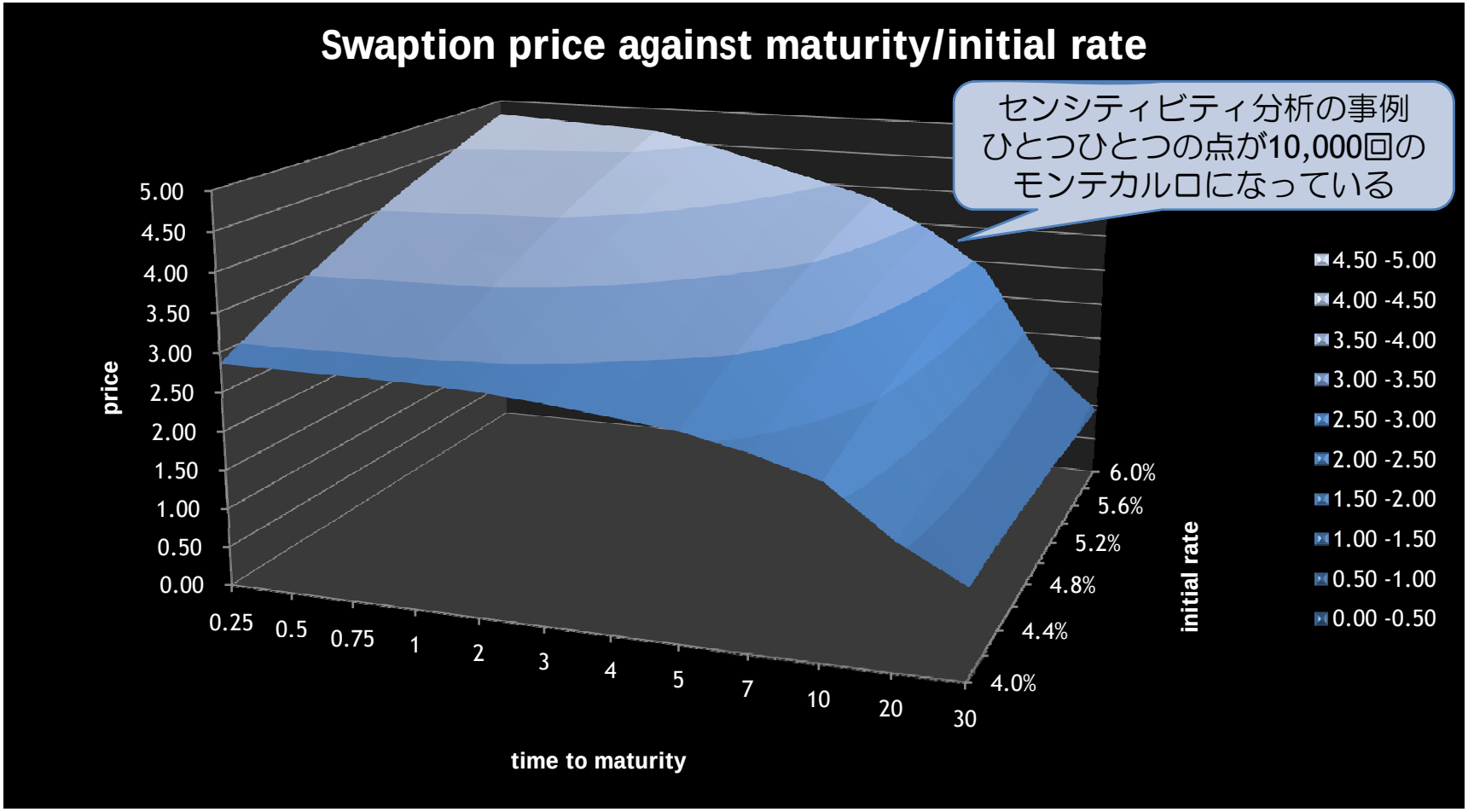
Copyright (C) Numerical Technologies Incorporated 2008/7/11

175倍速の計算能力があれば2次元表も素早く計算できるので…



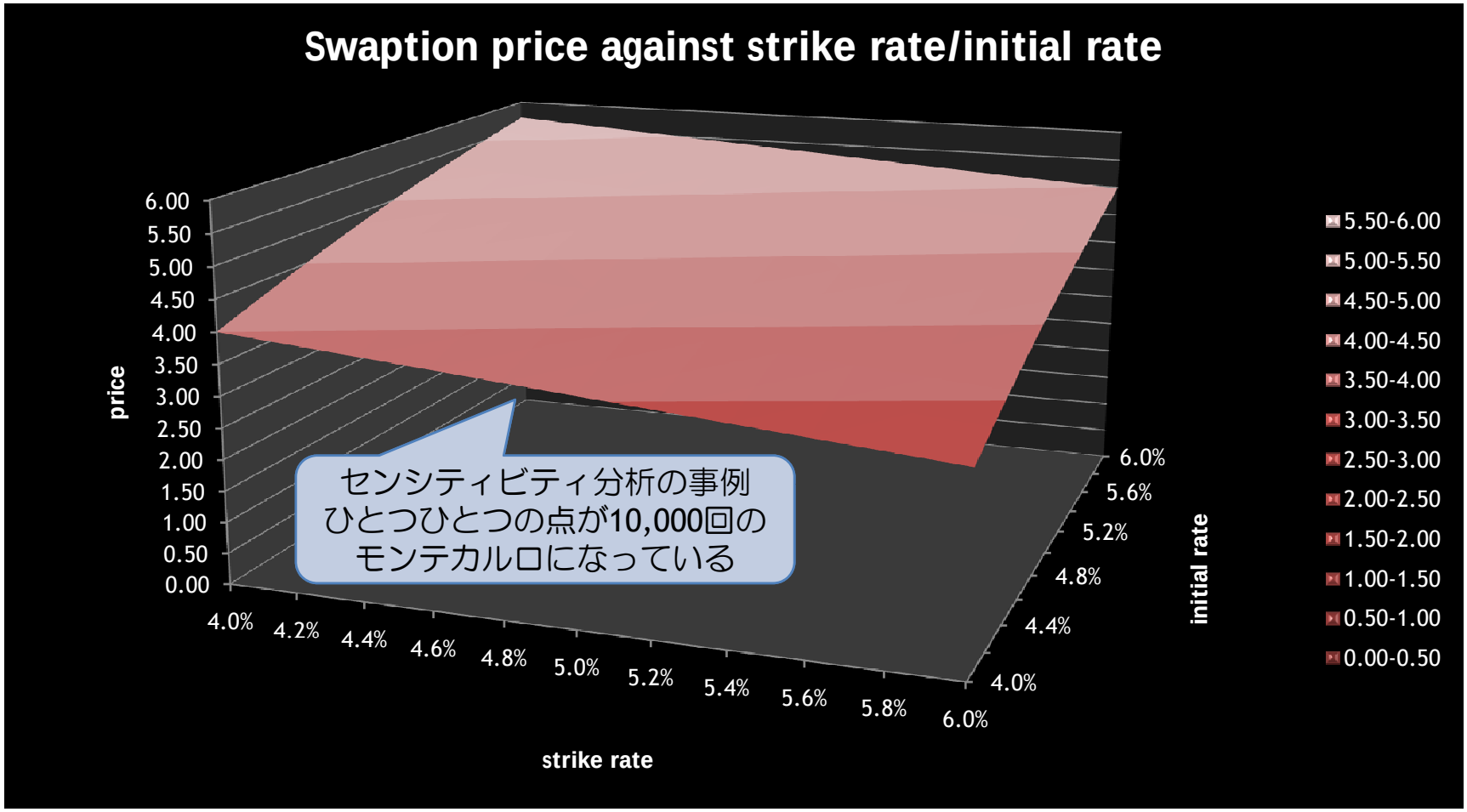
Copyright (C) Numerical Technologies Incorporated 2008/7/11

センシティビティ分析も容易



NUMERICAL TECHNOLOGIES Copyright (C) Numerical Technologies Incorporated 2008/7/11

175倍速の計算能力があれば複雑な商品も差分法を使ってさえ解ける



NUMERICAL TECHNOLOGIES Copyright (C) Numerical Technologies Incorporated 2008/7/11

高速化の秘密はアクセラレータ



- このデモでは本セミナーの直前2008年6月に発売されたばかりのNVIDIA社のGTX280を2枚、PCの拡張スロットに入れて使用した。
  - 詳しい方のために補足すればSLIは使っていない。これはテスト時点でCUDAがサポートしていなかったから。
- このアクセラレータはGPGPU（General Purpose Graphics Processing Unit）という新技術。
  - 同様のGPGPUにはATI社のRV770がある。GPGPU以外のアクセラレータではCell/B.E.、ClearSpeedが出荷中。Intelも2008～2009年に開発コードネームLarrabeeを出荷すると発表している。
  - 最近のHPC関係ではホットイシュー。
- なお、NVIDIA社のGPGPUを使ってテストしたのは価格も安く入手しやすかったためで、同社と弊社が特別な関係にあるといった特殊事情はない。弊社も他の選択肢を積極的に試していきたい。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

ベンチマークテストの結果

モンテカルロの試行回数と精度	GPGPUなし (すべて倍精度)	GPGPUx1	GPGPUx2
1,000,000回 倍精度	90.618秒	4.043秒 (22倍の高速化)	2.094秒 (43倍の高速化)
1,000,000回 単精度	90.618秒	1.017秒 (89倍の高速化)	0.596秒 (152倍の高速化)
10,000,000回 単精度	904.376秒	10.223秒 (88倍の高速化)	5.165秒 (175倍の高速化)

- このベンチマークテストはIntel Core2 Quad Q9550/2.83GHz搭載のPCを使い、CPUの1コアを基準として行った。なお、現在最速のIntel Xeon X5460/3.16GHzの1コア性能を別途測定した結果と比較すると、10,000,000回モンテカルロが760.044秒だったので147倍の高速化となる。SMPとの比較ではないから、4コア、8コア、あるいは16コアと比較したければ4、8、16で割って適当に倍率を推定してほしい。
- GPGPUなしの場合の計算精度はすべて倍精度とした。仮に単精度にした場合は、Intel系CPUの場合はかえって倍精度よりも遅くなるので実環境を想定した比較対象としては適さないと判断したもの。
- PCの計時機能は時間分解能が低く、またOSのスレッドスケジューリングも影響するので、1秒未満の時間計測には誤差が含まれている。またスタートアップにかかるアムダール則の影響も当然ある。他方、10,000,000回モンテカルロの傾向を見る限り性能向上曲線はリニアのようだ。したがってラフに言えば**本例題に関する限りGPGPUの1枚あたり88倍の性能向上がある**と考えてよいだろう。製造メーカーであるNVIDIAはHPC用として外付けのx4ユニットを今秋出荷予定。**GPGPUx4ならば最大350倍の高速化が予想される**ということ。
- GTX280の弱点は倍精度計算が遅いことだと言われる。FLOPS値で言えば倍精度は単精度の1/8の性能に過ぎない。それなのに上記の計測結果では1:3.5程度と善戦している。これはなぜかというと、金融計算では倍精度であっても整数演算が多く含まれるので、32bit integerで行われる整数演算が多ければ倍精度の遅さが隠蔽されるため。
- **これほどの性能向上がある以上、これまで単純なグリッドコンピューティングを使って行われてきた計算の一部は、今後この種のアクセラレータが置き換えていくだろう**。実際、現在世界最速のスーパーコンピュータはグリッドとアクセラレータのハイブリッド機（LLNLのRoad RunnerでCell B.E.とのハイブリッド）。東工大のTSUBAMEもハイブリッド機（ClearSpeedとのハイブリッド）である。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

今回デモの実現方法



1.秋葉原へGo



2.パーツをGet



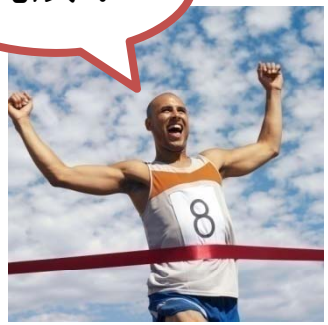
3.PCを組み上げる



4.きついプログラミング



5.そして
一週間後…



Copyright (C) Numerical Technologies Incorporated

構築コストも安い



＋

組み立て料金：¥19,500

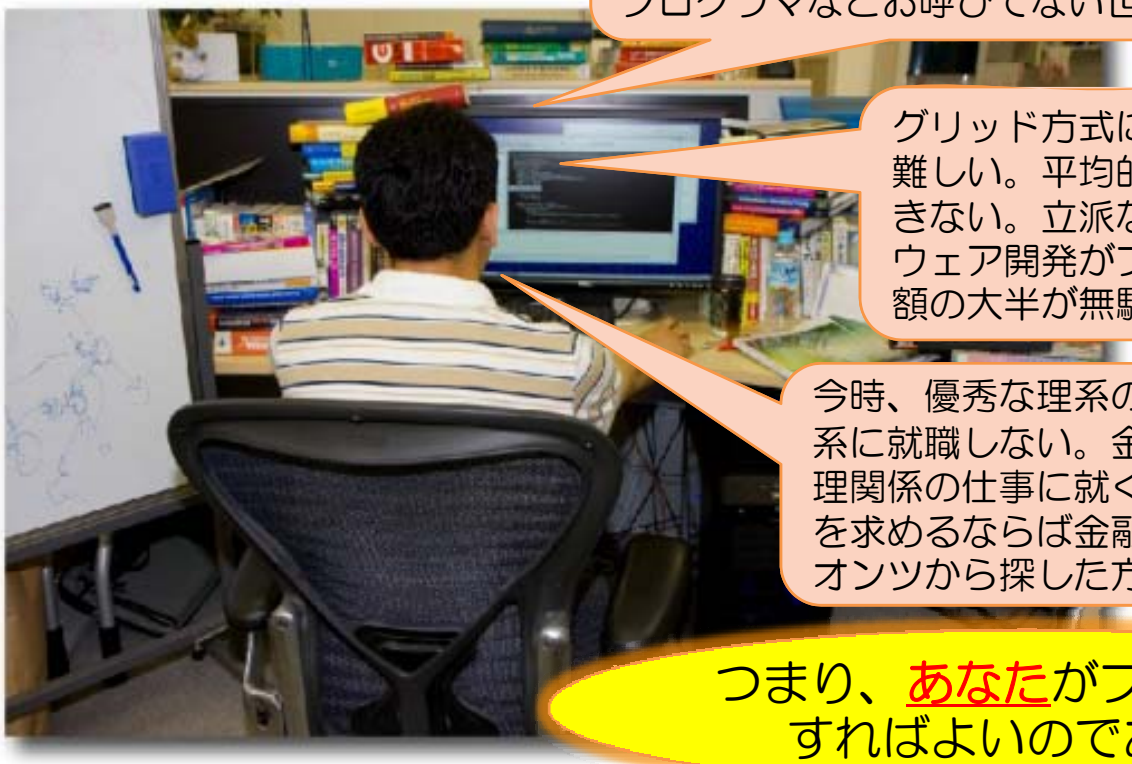
＝

合計：¥428,840

【同じことを試そうとされる方への注意】
市販のPCを使わずに自作パソコンを選んだのはGTX280が大容量の電源と新しいバス（PCI Express Gen.2）を必要とするため。今お使いのPCを無理矢理に使うと危険なので注意！
企業内でも安心な導入をしたければ、GTX280のGPGPU版製品（TESLA）に変更し、大容量電源を備えたHP、DELL、Sunのハイエンドpc（主にゲームユーザー向け）が適切だろう。
この自作例のようにGTX280を2枚入れる場合は電源容量の面でさらにハードルが上がり、NVIDIAによれば動作実績があるのはSun MicrosystemsのWindows PCなど、少数に限られる。あるいは外付けユニットを選ぶのが賢明だろう。東工大では流体計算アプリケーションで4枚も試していると聞く。
まとめれば、この図で計算したコスト試算は自作部分が多いだけ安い側に振れている。自作しない以上、金融機関が購入できる末端価格はより高価になるはず。

Copyright (C) Numerical Technologies Incorporated

難関はプログラミング



アクセラレータ方式を選んだら、ハードウェアから理解しなければ動作さえさせられない。委託開発の下請けJavaプログラマなどお呼びでない世界。

グリッド方式にしたところで並列処理は難しい。平均的なプログラマでは開発できない。立派なハードを買ってもソフトウェア開発がプアなせいでシステム投資額の大半が無駄になってはいないか。

今時、優秀な理系の人はメーカーやIT系に就職しない。金融機関に行って数理関係の仕事に就く。優秀な開発要員を求めるならば金融機関の中、特にクオンツから探した方がよいだろう。

つまり、あなたがプログラムすればよいのである。

NUMERICAL TECHNOLOGIES Copyright (C) Numerical Technologies Incorporated 2008/7/11

あなたの市場価値が上がる

- ここまでできる人材は少ない。ということは転職に有利。会社が傾いたら、嫌になったら、辞められる。過去のトレーダーたちもそうしてきた。
- とりあえず排出量権取引で稼ぐとか出来るかも知れないが、トレーディングを規制過剰の東京で行う意味がなくなりつつあることに注意すべき。将来、旧ソロモンのようにプロップ事業そのものが廃止かも。金融機関のアクティブポート運用はすべてオルタナ化するかも。あなたの居場所がなくなった後、クオンツトレーダーには海外移籍後も生き残れる実力が必要と思う。
- 金融機関側も「いつでも辞められる」くらいのアグレッシブなトレーダーが欲しい。だから雇用側・非雇用側で利益相反はない。

Copyright (C) Numerical Technologies Incorporated 2008/7/11



3. システム構築とHPC

NUMERICAL TECHNOLOGIES

HPCを実現する２つの方法



- ビジネスでHPCを実現するには、①グリッドコンピューティング、②アクセラレータ、の２つの方式がある。
  - かってのベクトルスーパーコンピュータ方式は今では選択肢として適さない。
  - 学術系と違ってビジネスでは性能だけ競うわけではない。現在世界最速のLLNL Roadrunnerはグリッドとアクセラレータのハイブリッド方式だが、ビジネスでハイブリッドをやるならば、
    複雑× 複雑＝失敗
    の覚悟が必要。
- どちらも万能ではなく、プログラミングも大変。

Copyright (C) Numerical Technologies Incorporated

両方式の優劣

	①グリッド	②アクセラレータ
汎用性	⭕ 高い。様々なプログラミングモデルを選択できる。	❌ 低い。現世代はSIMDが前提（限定的なMIMD）。繰り返し処理は得意だが、大量データ処理は苦手。
信頼性	⭕ 実績あり。ただし悪いベンダーが絡むとプアなグリッドにもなりうる。	△ まだ出始め。代替的手段で実行結果を確認して使わないと危ないだろう。しかもどの方式が生き残るかまったく読めない。
コスト	❌ 高額	⭕ 圧倒的に安い
開発リスク	❌ 高い。金融機関の場合は未熟なベンダーの言いなりで委託開発になるのが原因。学術系HPCからみれば大半が性能未達と言って良いだろう。	⭕ 低い。理由は簡単。現世代のアクセラレータは小さなプログラムしか格納できず、大規模プロジェクトにならないから。担当者レベルでできる。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

投入データ量が選択の鍵

- 個別商品のプライシングや小規模なリスク管理に使う
  - たとえ1-2年で製品寿命が尽きようと、②アクセラレータ方式を第一選択に考えるとよい。何と言っても圧倒的に安いから。
  - それればかりか、ノード間の通信が遅延要因になるグリッドよりも、同一筐体内で高速接続するアクセラレータの方が性能も上かもしれない。
- 大規模なポートフォリオのリスク管理に使う
  - 現段階では①グリッド方式しか選びようがない。
    - プログラミングモデルが複雑で大規模開発に向かない。
    - 出始めとあってアクセラレータの本命がわからない。アクセラレータを使う大規模な長期開発を始めても、出来上がった頃には当該製品が生き残っていないかもしれない。
    - 数値の信頼性が低い（IEEE754準拠レベルが低い）。
    - 現行世代のアクセラレータは高速メモリが8bitパソコン並みに小さい（NVIDIA GTX280ではわずかに16KBのshared memory）。低速なグローバルメモリの量も32bitパソコン程度（同じくGTX280では1GB）。このためさらに低速なCPU・アクセラレータ間のデータ転送（PCI Express Gen.2 x16で片方向8GB/s）がボトルネックとなり、64bit OSを要求する規模の大量トランザクション処理には困難を伴う。
  - とはいえアクセラレータ技術が安定したらハイブリッド方式も検討すべきだろう。それがいつになるのか不明だが。特に二重モンテカルロになるケースでは今でもハイブリッドが第一選択かもしれない。

Copyright (C) Numerical Technologies Incorporated

ベンダーとの利益相反

- グリッド・コンピューティング＝ブレードサーバーの大量販売。顧客にグリッド方式をやめてアクセラレータ方式に変更されたら、ベンダー側は大幅な受注額減になる。
- そんな提案書は、金融機関自らが要求しない限り、ベンダー側から持ってくるとは考えないほうがよいだろう。
- 無駄な生命保険部分をなくした海外旅行保険が買いにくいのと同じ理屈である。

Copyright (C) Numerical Technologies Incorporated

金融HPCの歴史

- 金融業界にはすでに国内・海外を問わず多数のグリッドコンピューティングの導入実績がある。
  - 非常に直接的に言うならば、1980-90年代前半に高速性から流行したSybaseを、その後Sybaseが落ち目になったのでOracle 10g以降のクラスタに置き換えたということ。グリッドコンピューティングとクラスタリングは言葉の使い方が曖昧だ。
  - ところが日本国内の金融関係で「グリッドコンピューティング」あるいは「HPC」と言った場合、通常はコンピューティンググリッドかつブレードサーバクラスタを指し、なぜか「金融工学関係で先進的なもの」と思われている。つまりbuzzwordとしての「グリッドコンピューティング」が金融HPCの現状に近い。これは2004年頃に行われたセールス攻勢が原因だろう。悪気はないにしても産総研（AIST）も加担したと言える。
- 過去にあった金融HPC
  - 海外では80年代末のMBSブームがHPCの第1期と言える。イタリア人トレーダーグループがイリノイ州のCRAYスーパーコンピュータを借りてARMやCMOの計算を行った話は伝説になっている。なお当時あったNASAのリストラにより人材がウォール街に移動したせいだとも言われるが、私の知る限りメインフレーム統合に従事していたのであって、デリバティブ関連はより若い世代が担っていたと記憶している（ただしそのボスは高齢でMBAもとってないのが当時の風潮）。その次は90年代前半のスワップビジネス立ち上げ期で、大半はC/C++を使ったが、モルガンスタンレーのAPLライブラリのようにインタプリタに拘った人たちもいた。後発組のゴールドマンサックスもGSAllからGSOneへと標準化もされていないC++言語にNIHライブラリを組み合わせて自社環境を作った頃（かのダーマン先生が南ア出身のプログラマとしてゴールドマンのFSGにいた）。Sun Microsystemsのワークステーションが全盛を迎えており、遅いインタプリタや表計算言語まで組み合わせてグリッドコンピューティングをしていた。これが第2期とすれば、マシンルームに数千ノードのクラスタを置くようになった現在は第3期と言えるだろう。そして第4期は今から始まるところであり、すでに見たGPGPU技術によるコストリダクションが始まりそうな予感がする。
- ディアスポラとノウハウ断絶
  - 日本では1990年代半ばに日系から非日系に金融機関の人材が大量流出した。バブル期に高い金を払い、欧米金融機関との資本提携と海外進出を通じて育成した戦略人材がここで失われたと言って良いだろう。HPC関連でもLindaを使ったネットワーク並列計算環境、SBCMの野望とか、富士銀行のブティックによるCray購入、東海銀行のデリバティブ子会社、三和銀行が買った日立製スーパーコンピュータ等々（皆は忘れても私は覚えている）、研究と運用がそれなりに行われていた。しかし、1990年代前半に起きた海外事業撤退劇、そして度重なる金融機関合併で部門そのものが消滅した結果、そういう動きもいったん終息した。それがなぜか2004年頃になって「我が国初のグリッドコンピューティング」という人たちが日系から出てきたから不思議で、人材とノウハウの断絶を示している。
  - 欧米でも似たような「社員がやめてノウハウがなくなった」話は多いようだ。前向きに言えば、転職先にそのノウハウが受け継がれ広まっていく。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

学術系HPCと金融HPCのギャップ

- 国内ばかりでなく先進的と言われる欧米の投資銀行でさえ、コンピューティングの専門家から見れば笑ってしまうような技術ブームが起こり当然挫折することがある。例えば1990年頃の米国金融界で何とSmallTalkがbuzzwordになり、しかも「最新の言語」として紹介され、ついに開発プロジェクトまで始める人が出るシーンを目撃した（CS出身の方ならばどれほど驚いたかは想像できると思う）。モルガンスタンレーのAPLもその種の例だろう。彼らは独自のバージョンまで作ったと言うから恐れ入る。他にもIBM/AS400でデリバティブとか、国内ではニューヨークで行われた第1回Java Oneカンファレンスに引っ掛かった人たちがJavaを「C++よりも高速な純粋オブジェクト指向言語」というキャッチフレーズで国内金融界に紹介するところを目撃した。当時ですらジョークに聞こえたのは言うまでもない。
- 国内の場合、金融HPCのプロジェクトでは、調査会社（ガートナーなど）を使うよりも、ベンダーの言いなりで決める傾向がある。金融に限らず先端技術は属人的。IBMもHPも関係ない。それなのにコアの技術であっても開発者に直接コンタクトして「大丈夫か」調べることなどまずない。金融HPCではなぜか発注側の金融機関は目標性能をベンダー側に保証させない。開発契約に達成すべき性能を明記しないしFLOPS値を実測しないこともある。目標性能に達しない場合はベンダーを責めずになんと追加発注してしまう。こういう「甘さ」は学術HPCから見れば信じられないと思う。また金融HPCでは、ブレードサーバーを売り込む熱心な営業マンの提案の方が、真っ当な技術的提案よりも通りやすかったりする。「シンクライアント」とか「クラウドコンピューティング」とか「オートノミックコンピューティング」とか、HPCとしては目的外使用の「営業用語」に引っ掛かる人も多い。
- 金融HPCでは工期見積もりが甘く、学術系HPCとは逆にハードウェアを買ってからソフトウェアを作る傾向がある。HPCは難易度が高い。欧米投資銀行を見ても長期の研究開発が蓄積された自社開発かパッケージ購入であり、短期の委託開発向きではない。それなのにーから行うアプリケーション開発を短期間（1年以下とか！）で計画し当然目標未達になったりする。重要でない問題、例えばRedHat LinuxかSUSE LinuxかWindowsかの選択、に拘りながら、ネットワークトポロジやI/O設計など性能と投資効果を半減させかねない問題に無関心だったりする。しかも本来外側で防御すべきHPCクラスタの個別ノードにアンチウィルスソフトを入れたりする。
- こんなことが起こるのは、金融機関にも介在するシステムインテグレーターにも、肩書きが研究職であろうとコンサルタントであろうと、計算機科学系の計算センター級のスタッフがいないからだろう。少なくとも私はそういう人をほとんど知らない。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

HPCは目的ではなく手段

- 金融業界の人はHPCをやりたいのではない
  - 実利目的だから金融HPCは健全。目的が先にあり手段がHPCになっただけだ。システム開発のリスクなど本来とりたくはないし、TOP500競争とは無縁。金融業は営利ビジネスであるから、過大投資にはブレーキが働く。
  - だから業務職が主導する導入目的が明快なHPCプロジェクトはうまくいくが、研究職が主導する自己目的化したHPCプロジェクトは大抵は挫折する。
  - JAXAの超高速インターネット衛星「きずな」やES2、京速に対する批判と類似の心配は、金融HPCに関する限り無縁であろう。

Copyright (C) Numerical Technologies Incorporated

システム部員をSC08に送ろう



- アングロサクソン的な方法
  - プロフィットセンターたるフロント部門のボスの采配により、金に糸目をつけずに外部から人材を集めればよい。
  - 実際、1980年代中頃からバブル崩壊までの間、邦銀は海外現法を使ってそうしていた。
  - では2008年の今、東京にお雇いインド人を集めればよいのか。それは東京でデリバティブ業務をやる発想からして無理がある。
- もう一つの方法
  - 主要なHPCの学会のひとつSC。次回SC08は2008年11月15日。海外では金融HPCに関する学会発表も目立つようになっている。
  - 最近の日系金融機関はリサーチにかける経費を絞り過ぎに見える。今時の最優秀な理系学生は金融機関に就職する。メーカーには行かない。せっかくの有為な人材を派遣して情報収集に努め、ダメなベンダーやインテグレータの言いなりになって意味のないグリッド構築にならないように、自ら技術トレンドについて行くべきだろう。
  - そうでないと、いつまでも「RedHatかWindowsか迷ってます」とか「Platformのミドルウェアを入れたから大丈夫です」のレベルを超えらない。低レンテンシ広帯域ネットワーキング、トポロジ、並列I/O、並列プログラミングモデルといったHPC独特の専門知識が蓄えられず、ベンダーの言いなりになってせっかくのシステム投資が無駄になりかねない。

Copyright (C) Numerical Technologies Incorporated



APPENDIX:
東工大でのALMモデル研究

テストデータ

入力した取引の件数	約340万件（スクランブルされた銀行データ）
キャッシュフロー総数	約5億個
勘定処理数	1シナリオあたり約44億件
入力した企業の総数	約20万社
日次シミュレーションの対象期間	3年間(1096日)
シミュレーション回数	1000（実使用では10,000回級）
会計処理方式	時価法+アモチ・アキュム処理
計算ノード数	1台から最大231台までスケールしてテスト実施

- モデルの時間計算量は、勘定処理の個数とモンテカルロシミュレーションの回数の積にほぼ等しい。
- 金融機関の実使用環境では数時間で解を求める。
- この計算規模(4兆4千億件の勘定処理) は、東京証券取引所の処理能力（2007年末で約500万件/日）の、実に**100万倍**！



Copyright (C) Numerical Technologies Incorporated 2008/7/11

アプリケーションの概要（１）

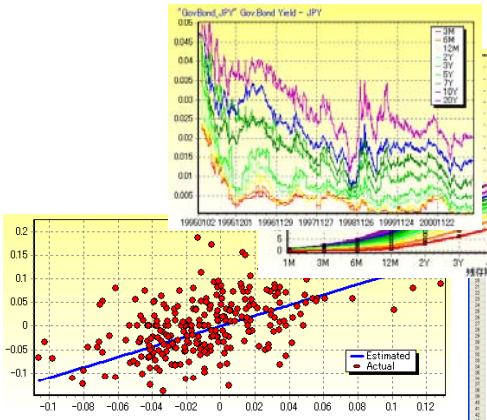
- 実在する金融機関を対象にモンテカルロシミュレーションを実施
  - 我が国最大のメガバンクの全資産負債ポートフォリオに合わせたスクランブル済みデータを入力
    → 銀行業では世界最大級。
- 金融業界で言うところのフルモンテカルロ方式を採用
  - 取引件別を明細レベルで入力
    → 取引件数の約150倍規模のキャッシュフローをモデル内で生成する。
  - 市場リスクと信用リスクを反映
    → 日次で市場価格と企業格付けを変動させる経済環境シナリオ生成モデルを内蔵する。
  - 相関性を考慮した多変量乱数を使用
    → 相関行列は rank 数10^3規模＋対角成分10^6規模のスカイライン型で、一般に事前予測が困難なマルチコを含む、つまり特異性が強く質の悪い線型構造である。
  - 吸収壁の存在
    → 時間軸方向に経路依存で、日次に離散化された、マルコフ連鎖モデルである。
- 財務シミュレーションである
  - キャッシュフローを現実の勘定科目に沿って仕訳処理
    → 1キャッシュフローあたり平均8.8個の勘定仕訳処理を行い、モデル内でシミュレートされた総勘定元帳に反映する。
  - カレンダー日付方式
    → 未来の貸借対照表・損益計算書を経済環境シナリオ毎にカレンダー日付に沿って生成する。
  - ビジネスユニットを記述
    → 部門間の内部振替の仕訳処理を行い、企業を部門毎に分解、部門毎の貸借対照表・損益計算書をシミュレーションする。
- モデルは決定論的でなければならない
  - シミュレーション試行は確率密度分布のtail部分まで完全に再現可能
    → オートマトンの計算環境への物理マッピングが変っても計算結果の同一性を保証せよとの、金融機関の業務要請からくる制約がある。
- 商用アプリケーションである
  - ビジネスとしての採算性を意識する非学術系ならではの制約

Copyright (C) Numerical Technologies Incorporated 2008/7/11

アプリケーションの概要（2）

- C/S型
  - Windows PCをクライアントとするC/S型アプリケーション。
  - 背後にグリッド環境があることを意識させない設計。
  - ユーザ認証と暗号化を行う。
- fault tolerant機能
  - 計算ノードの自動縮退機能を有する。そのため、ノード間でハートビートしている。
- ジョブスケジューラを内蔵
  - Platform、DataSynapsなどのサードパーティ製のスケジューラは使用せず自前でジョブ管理を行う。顧客にコスト面で迷惑をかけないため。
  - もちろん必要があればこれらスケジューラとの共存も可能であり、実際TSUBAME環境ではn1geと共存している。
- マルチOS
  - Windows、RedHat、SUSE、をサポートしており、ノード毎に異なるOSの混在も許している。
- ハードウェアベンダー非依存・中立
  - Sun、HP、IBM、Hitachi、NEC、Fujitsu… 対応がリーズナブルならば、会社としてどこでもサポートする方針。
- 金融HPC独特の要請
  - システム環境のコンフィギュレーションが変ったり、アプリケーションのバージョンが変っても、モンテカルロシミュレーションの結果が変ってはならない、という業務要請がある。
  - ゆえに決定論的なシステムとして設計しなければならず、コンパイラのバージョン固定とIEEE754コンフォーマンスレベル調整を併用している。Intel系CPUの場合、ここにregister sizingの問題がある。
  - そこで、64bit double型表現でsizingし、SSEは使用しない、FMAは行わず、NaN透過とし（sanitary testを行っている）、演算順序の最適化はしないことにしている。

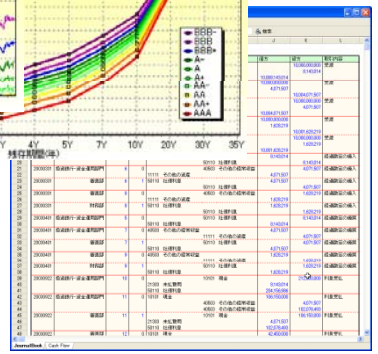
Copyright (C) Numerical Technologies Incorporated 2008/7/11



● 相関推定

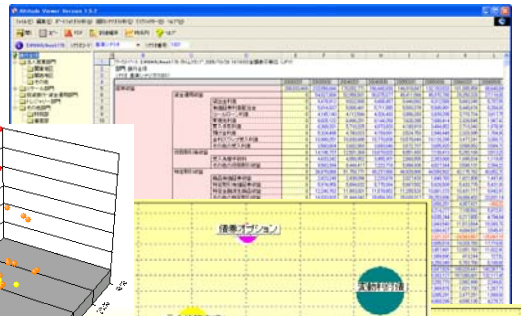
● 信用格付け推定

● 取引データ



数年間日次の将来シミュレーションを実施
●人間が定義する想定シナリオによる方法
● 1,000回～1,000,000回のモンテカルロ法

● 将来財務諸表導出
収益目標誘導



● BIS規制比率
最大損失額予測

● 寄与度解析
事業・取引先
撤退判断

NUMERICAL TECHNOLOGIES Copyright (C) Numerical Technologies Incorporated 2008/7/11

TSUBAMEの学外利用を選んだ理由

- 我々のようにOSの種類の差や特殊なハードウェアに慣れた立場からという意味では使い慣れたグリッド環境に見え、予想通りに動き、サポートも良い。ただし、利用期間の大半は学内利用で混雑していてノードを占有できず苦労した。
- 弊社内の環境からTSUBAME環境にアプリケーションを移植するまでに僅か１日を要したのみ。コンパイラを通して数カ所修正しただけで済んだ。ポータブルに作ってきたと言っても驚くべき事。
- これほど理想的な環境は、T2Kが稼働するまでの間は、TSUBAMEが国内唯一のものであったと思う。
- HPCの新分野に属する我々は、ES2的な計算センタよりも、民生分野との連続性が保たれたアーキテクチャを採用した計算センタを望む次第。
- そうでなければ国内のクラスタは見捨てて、海外のクラスタの計算アカウントをもらうべく、躊躇なく国外脱出していた。ネットワーク化された今日、計算センタの競争相手もグローバルだと思う。

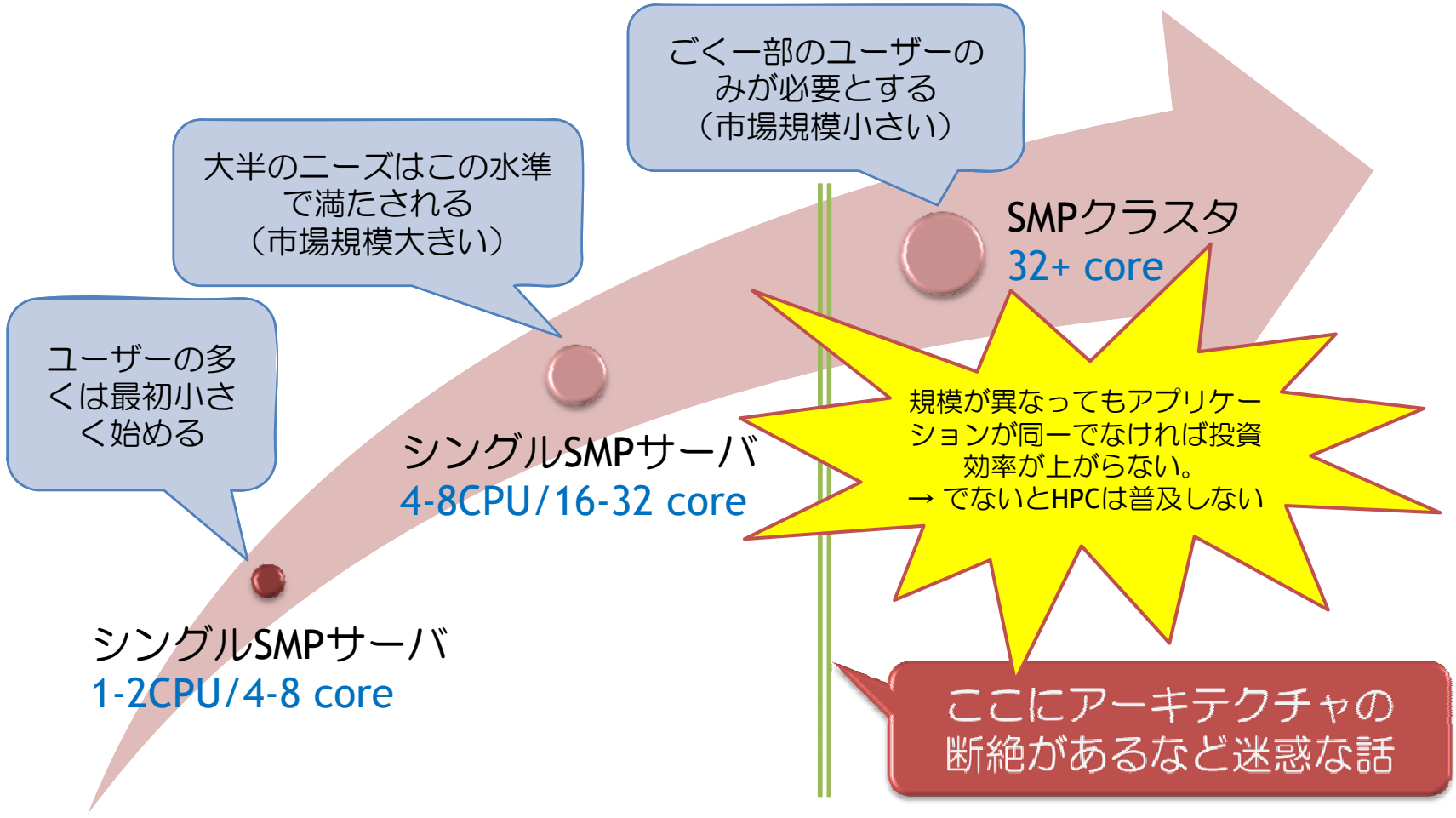
Copyright (C) Numerical Technologies Incorporated

汎用アーキテクチャが重要

- 今日、計算機センターのHPC系ジョブの大半を長年かけて熟成された既成のソフトウェア（創薬研究で使われるGaussianなど）が占めている。その種の既成HPC分野のニーズと金融のような新しいHPC分野のニーズは違う。
- 新規分野では、これからソフトウェアに開発投資し、そして投資回収しなければならない。ここで例えば特殊で普及していないvector型ハードウェアを相手にソフトウェアを書いたとしても採算ベースに乗らない。我々の顧客はビジネスのためにソフトウェアを使用する。我々はビジネスのためにソフトウェアを書く。採算に乗らないならばやらない。
- だからCOTSアプローチの環境でなければ意味がない。商用環境で性能を訴求した場合、マルチコアCPUの時代だから普通にスケールアップすればSMPクラスタ、スケールアウトしてconstellation になってしまう。ここでconstellation型のTSUBAMEが意味を持つ。

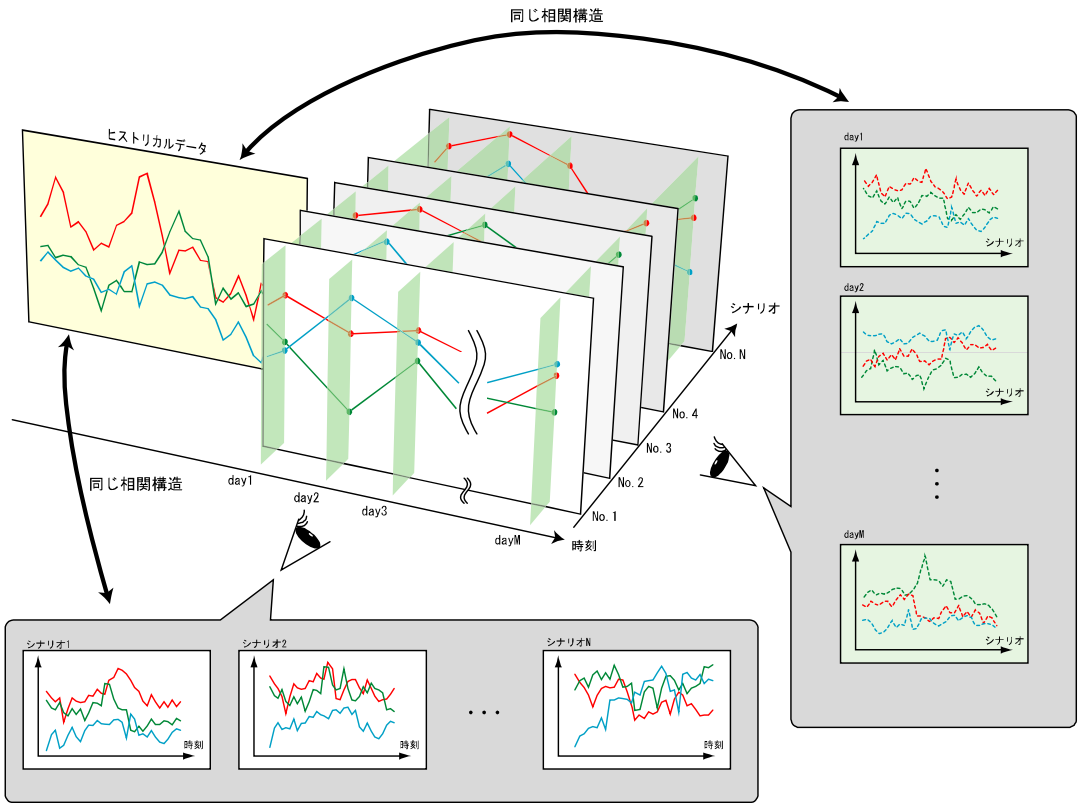
Copyright (C) Numerical Technologies Incorporated

連続的に拡大できる設計が重要



Copyright (C) Numerical Technologies Incorporated

伝統的HPC分野との違い



- BLASに性能依存するのは経済環境シナリオ生成モデル（左図）だけと言って良い。計算時間に占める比率はコンマ以下のパーセンテージであり、HPLで示される性能の優劣は意味をなさない。
- 十分に最適化された金融系計算プログラムは、(1)日数計算（day counting）、(2)割引計算（discount function）が大半を占める。我々の経験では、概ね半々である。
- (1)は整数演算。分岐がありパイプラインを乱す存在。
- (2)は主に超越関数expで、浮動小数点演算性能に依存。
- ベクトル化のチャンスは少ない。MIMD向きの計算。

Copyright (C) Numerical Technologies Incorporated

アプリケーションの計算フェーズ

TSUBAMEのノード1台を管理サーバ、N台を計算サーバとして占有。
ボス・ワーカ型で並列動作させる方式とした。

各計算フェーズの
時間計算量

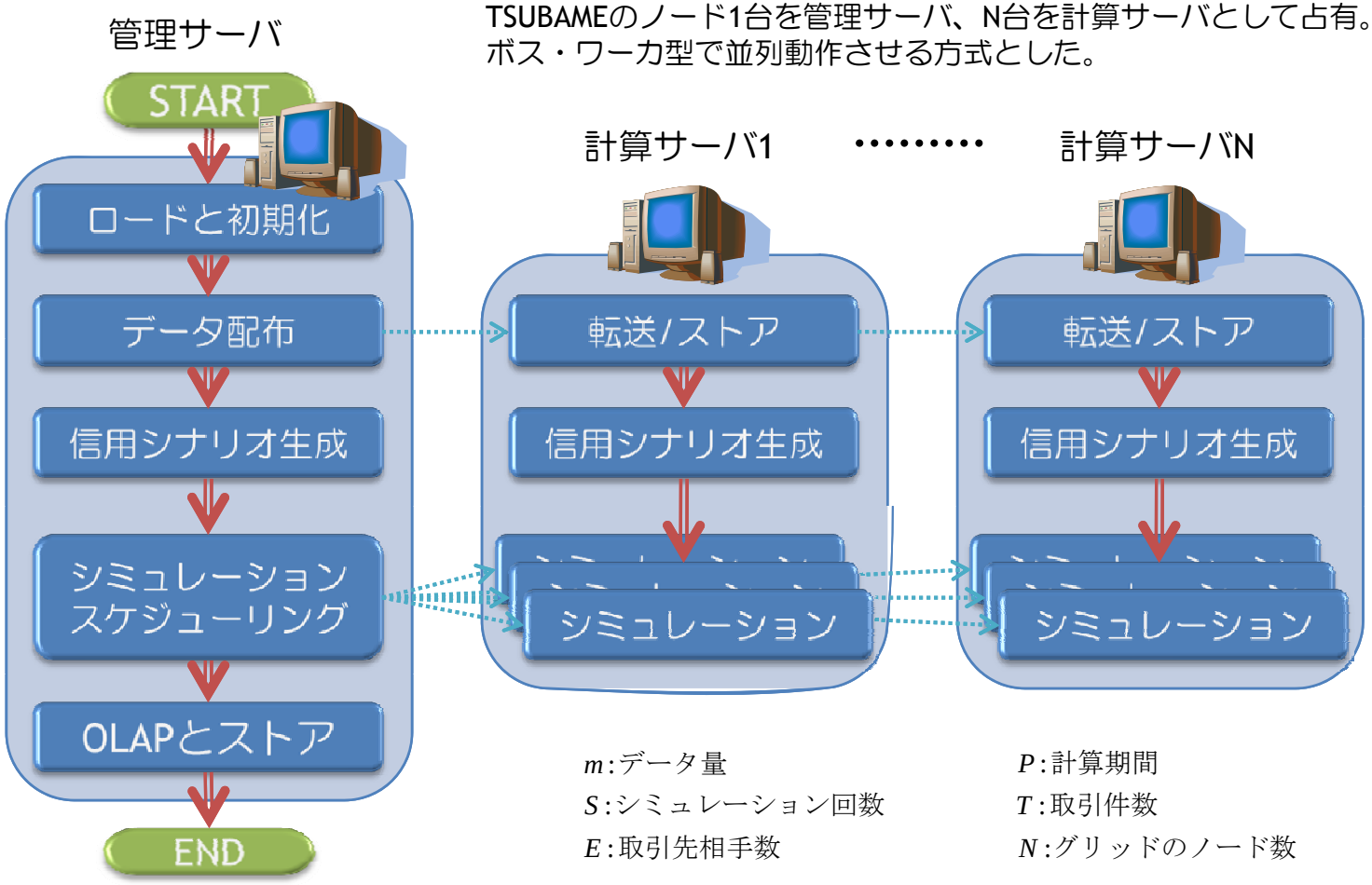
$O(m)$

$O(m)$

$O(S \times E \times P)$

$O\left(\frac{S \times T \times P}{N}\right)$

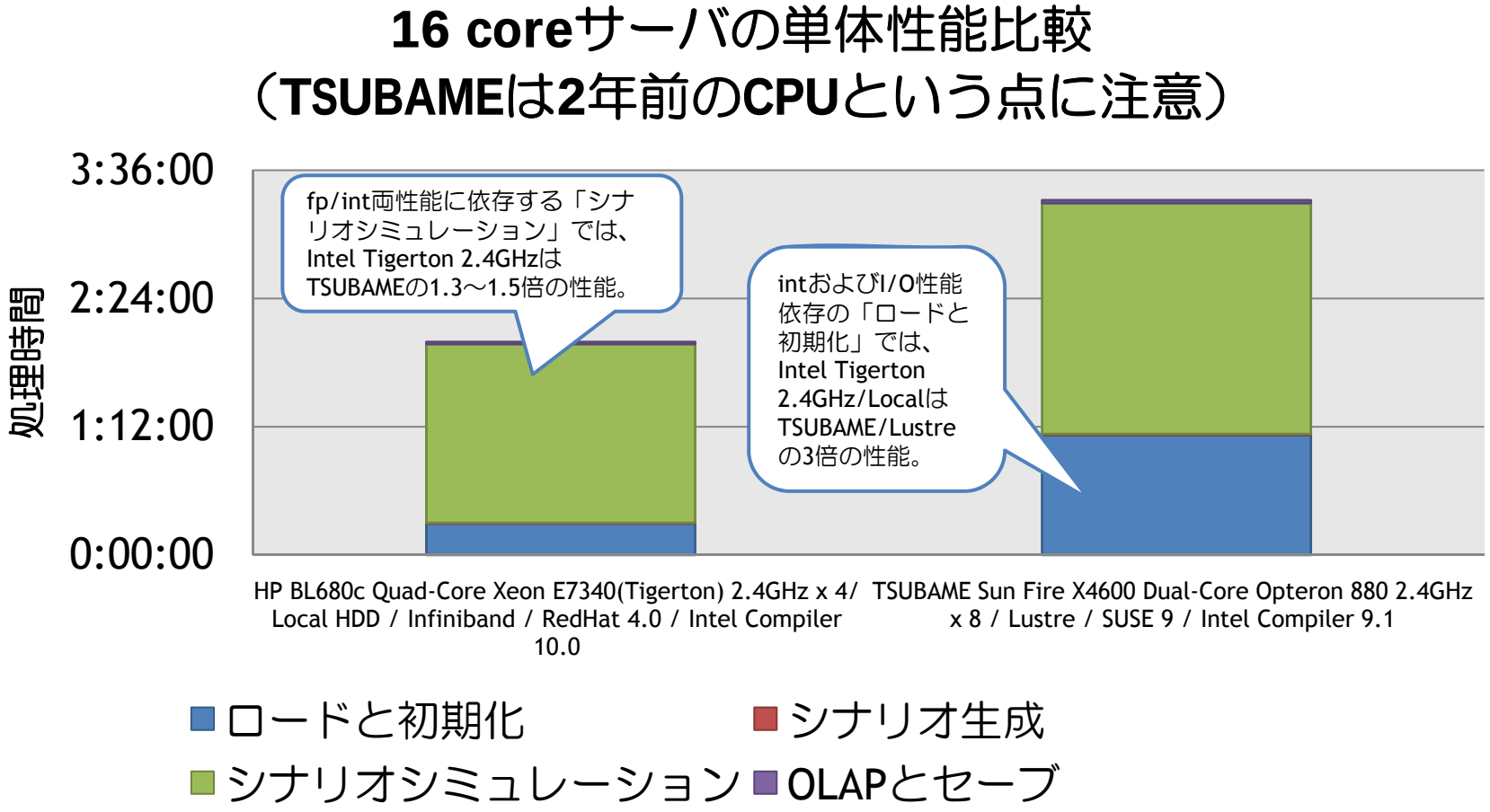
$O(m)$



m:データ量
S:シミュレーション回数
E:取引先相手数
P:計算期間
T:取引件数
N:グリッドのノード数

Copyright (C) Numerical Technologies Incorporated 2008/7/11

注意！ TSUBAMEのハードは2年昔のもの



Copyright (C) Numerical Technologies Incorporated

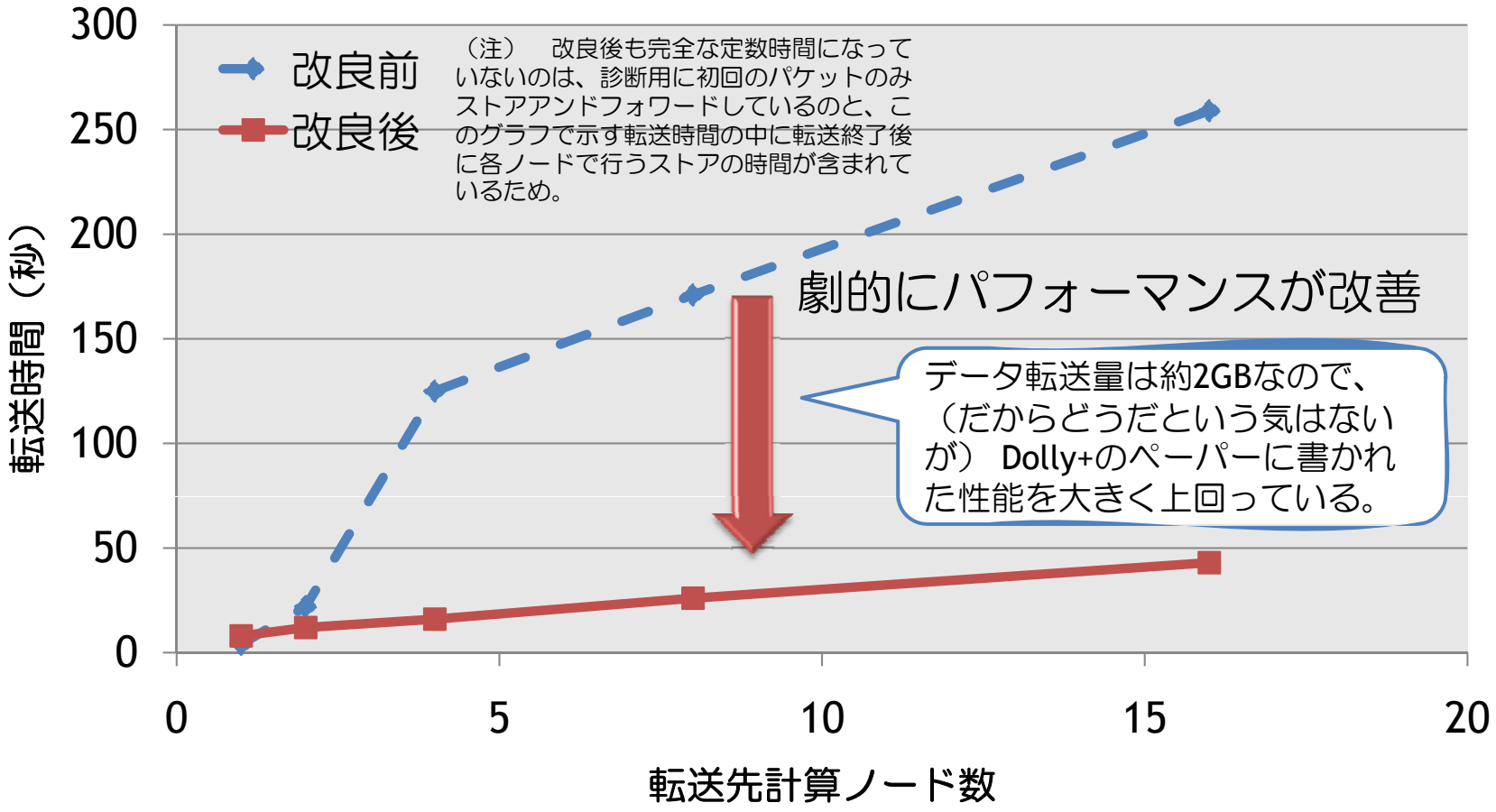
データ配布フェーズの改良

- 本アプリケーションをボス・ワーカ型で動作させるために、最初にサーバ・プロビジョニングする必要がある。
- 初期設計では管理サーバから計算サーバへ逐次ファイル転送していた。ここでロングパケット化とACKの改良は実施済み。ところが、投入データ量が大きいため、ノード数が増大するに従ってデータ配布フェーズにかかる時間が問題となった。
- そこでカットスルー方式でデータ転送する方式に変更。これでレイテンシが隠蔽され、計算サーバのノード数に関係なく定数時間でデータ配布フェーズを終えられるようになった。
  - この結構大変なプログラム変更を弊社では休日出勤して2日間で行った。月曜日になりその話を松岡教授にしたところ、松岡研でまったく同じ発想（マルチスレッドとパイプラインによる高速化）によるDolly+を作ったという話を聞かされてガックリ。
  - ムッとした弊社では、TSUBAMEのSUSE Linuxではなく、弊社内のWindows環境でも同じ改良を試してみた。関連する話題としてTCP Chimney Offloadの有無により、まったく性能特性が異なることがわかっていた。またTCP Chimneyに関連する初期のWindows CCSのバグも影響する（Windows 2003 Serverにも同じ問題がある）。
  - いろいろと研究したが、結局レイテンシ隠蔽には乱暴だがNagleをoffするのが手っ取り早い。パケットサイズが適切ならばTCP Chimney Offload機能のないNICや、Network teaming環境下であっても、バグのあるWindows CCS環境であっても、これでTCP Chimney Offload有効な環境とほぼ同等の実行転送速度が出る。もちろん瞬間的にCPUがbusyになるが、ディファレンシャルに動作しているから気にならない。



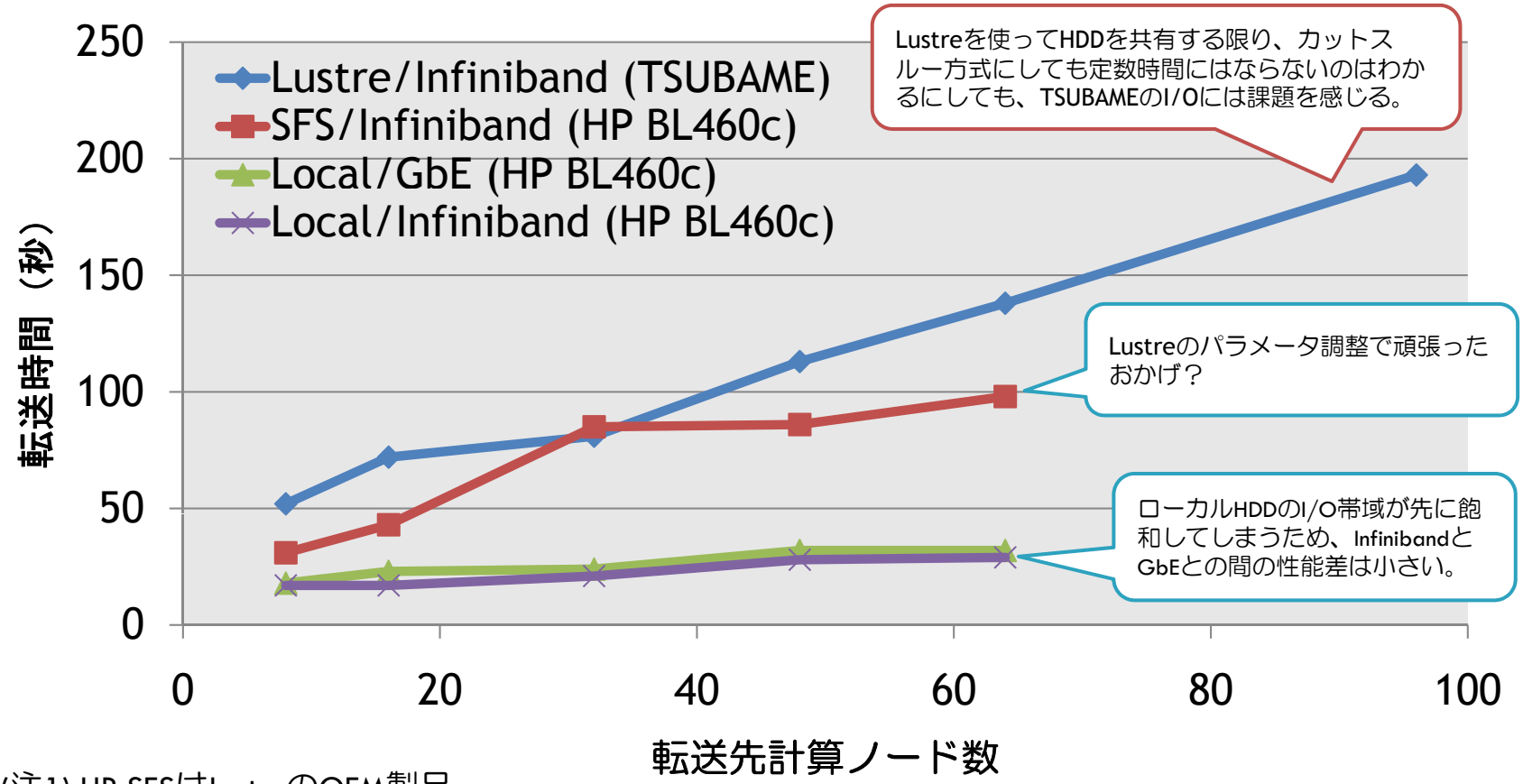
Copyright (C) Numerical Technologies Incorporated

劇的効果



Copyright (C) Numerical Technologies Incorporated

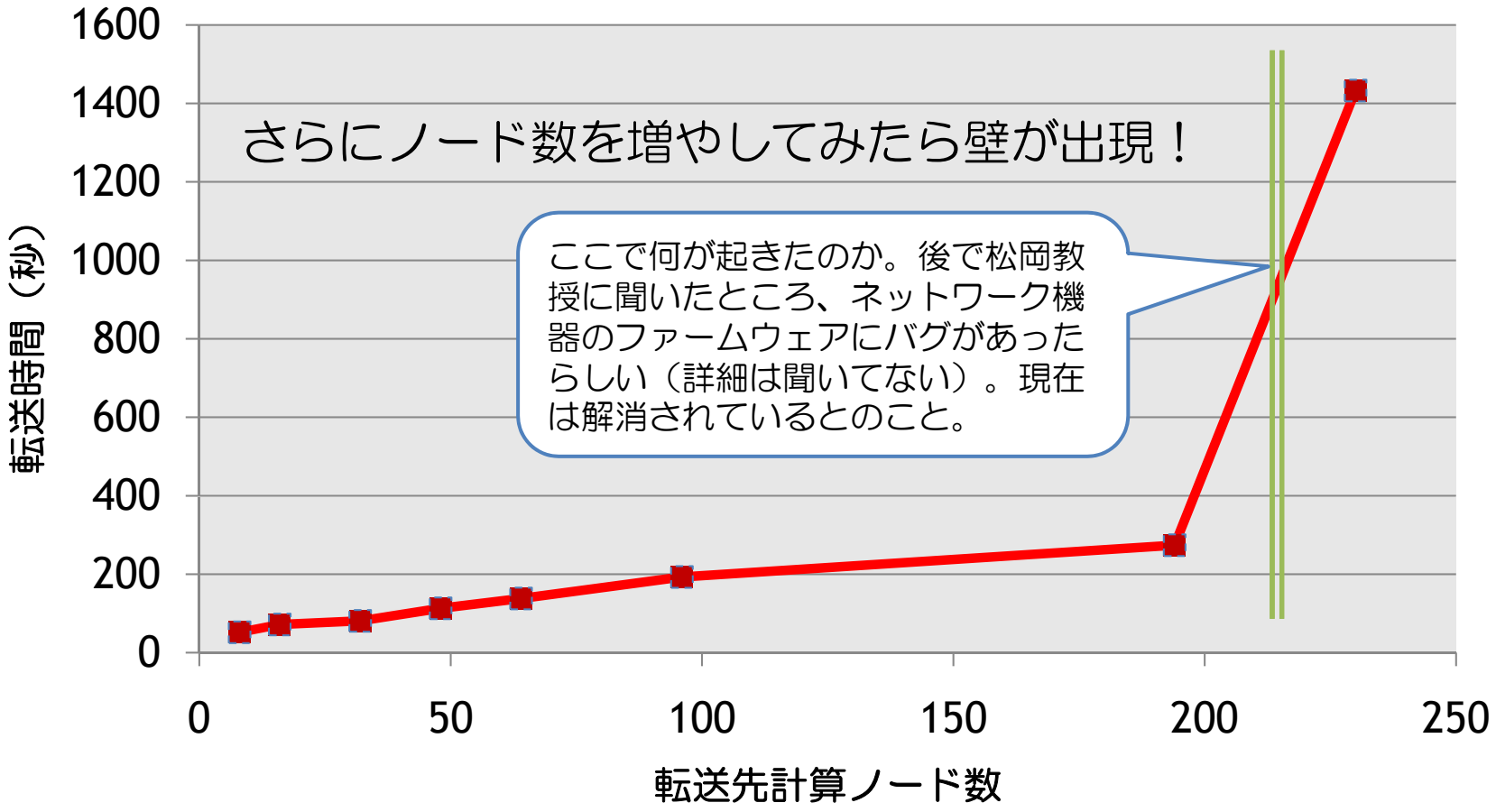
I/O設計は巧拙の出るところ



(注1) HP SFSはLustreのOEM製品

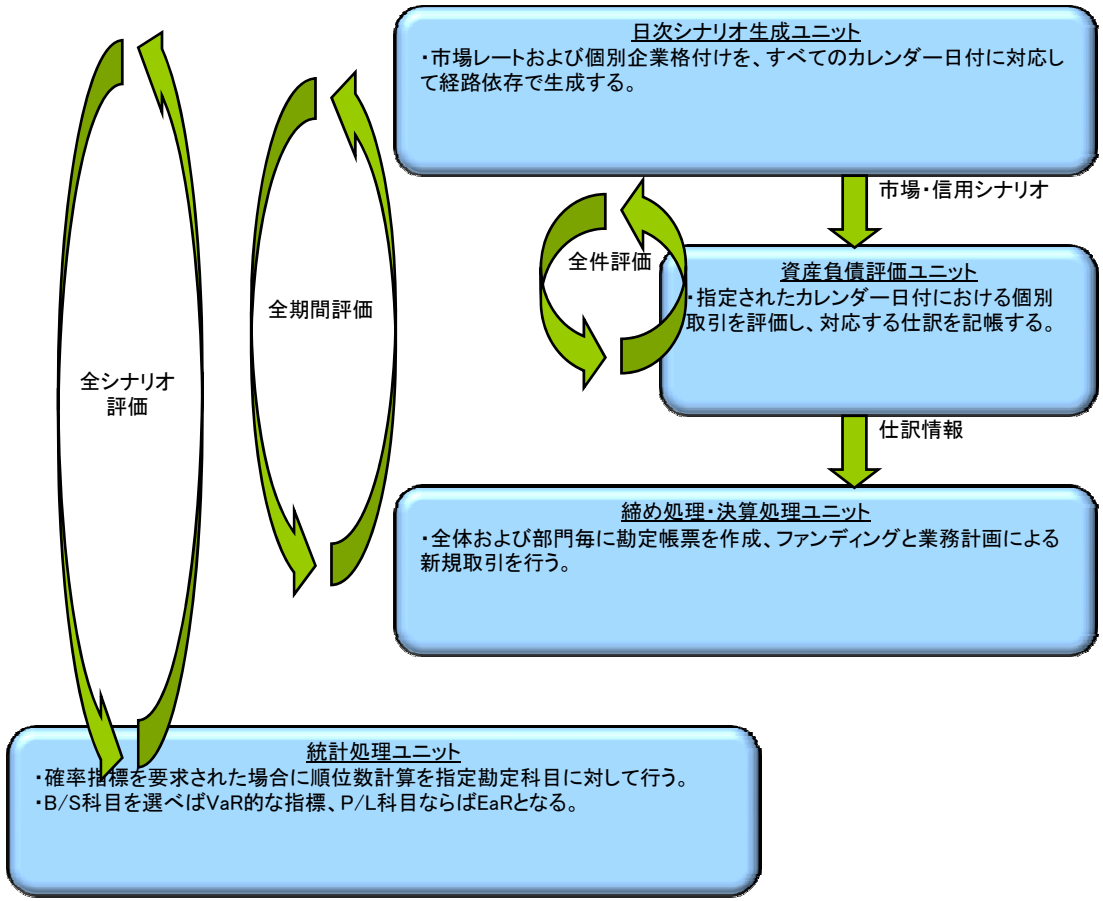
(注2) HP BL460cはQuad-Core Xeon X5365(Clovertown) 3.0GHz x 2、RedHat 4.0

Copyright (C) Numerical Technologies Incorporated 2008/7/11



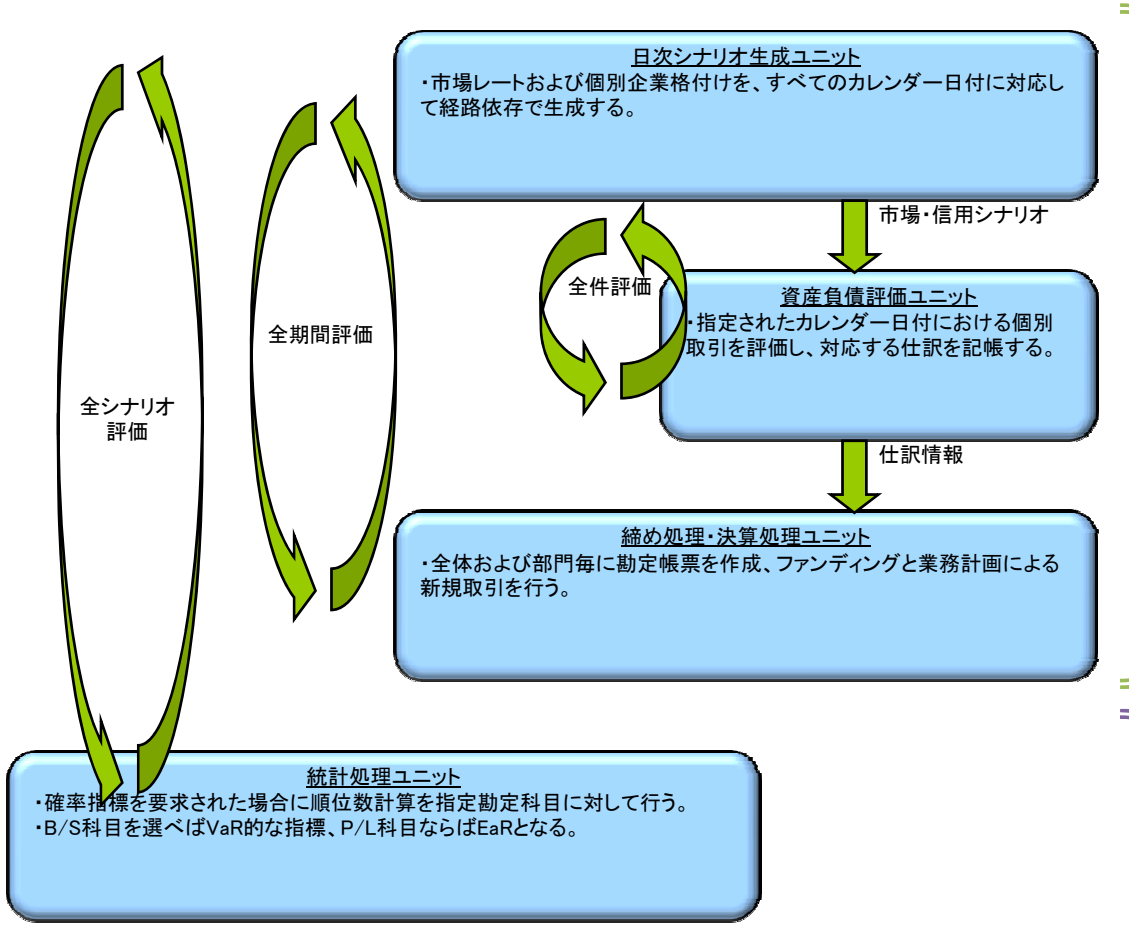
Copyright (C) Numerical Technologies Incorporated 2008/7/11

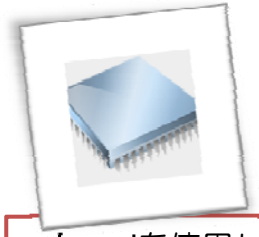
シミュレーションフェーズの改良



- 計算時間のうち最大部分を占めるシミュレーションフェーズは、単純に示すと左図に示す３重ループ構造になっている。
  - 最内側の“全件評価”ループは基準日取引データ約340万件＋シミュレーション中に生成された新規取引を勘定処理する。時間軸方向に処理する点を除けば、全件評価”ループ内の処理は件別毎のパラレルな計算で構わない。しかし次のレベルである“全期間評価”ループとのデータ依存性が高いために、密結合の並列処理である必要がある。
  - “全期間評価”ループは締め処理と決算処理を行う。ファンディングと再投資・再運用の業務計画の適用もこの階層で行う。“全期間評価”ループの進行はシーケンシャルな処理。同期が必要であり、barrierを使って実装する。
  - 最外側の“全シナリオ評価”はモンテカルロの回数だけ繰り返される。他分野のモンテカルロシミュレーションと同じく、金融の場合もモンテカルロの各シナリオ間では処理は独立している。基本的にshared nothing、したがって疎結合の並列処理に向いている。
- この論理構造を物理アーキテクチャにマップするところが本研究における最大の腕の見せ場。

Copyright (C) Numerical Technologies Incorporated 2008/7/11





pthreadを使用し、ノード内の8CPU16コアにマップした計16スレッドを走らせて、密結合の並列処理とした。



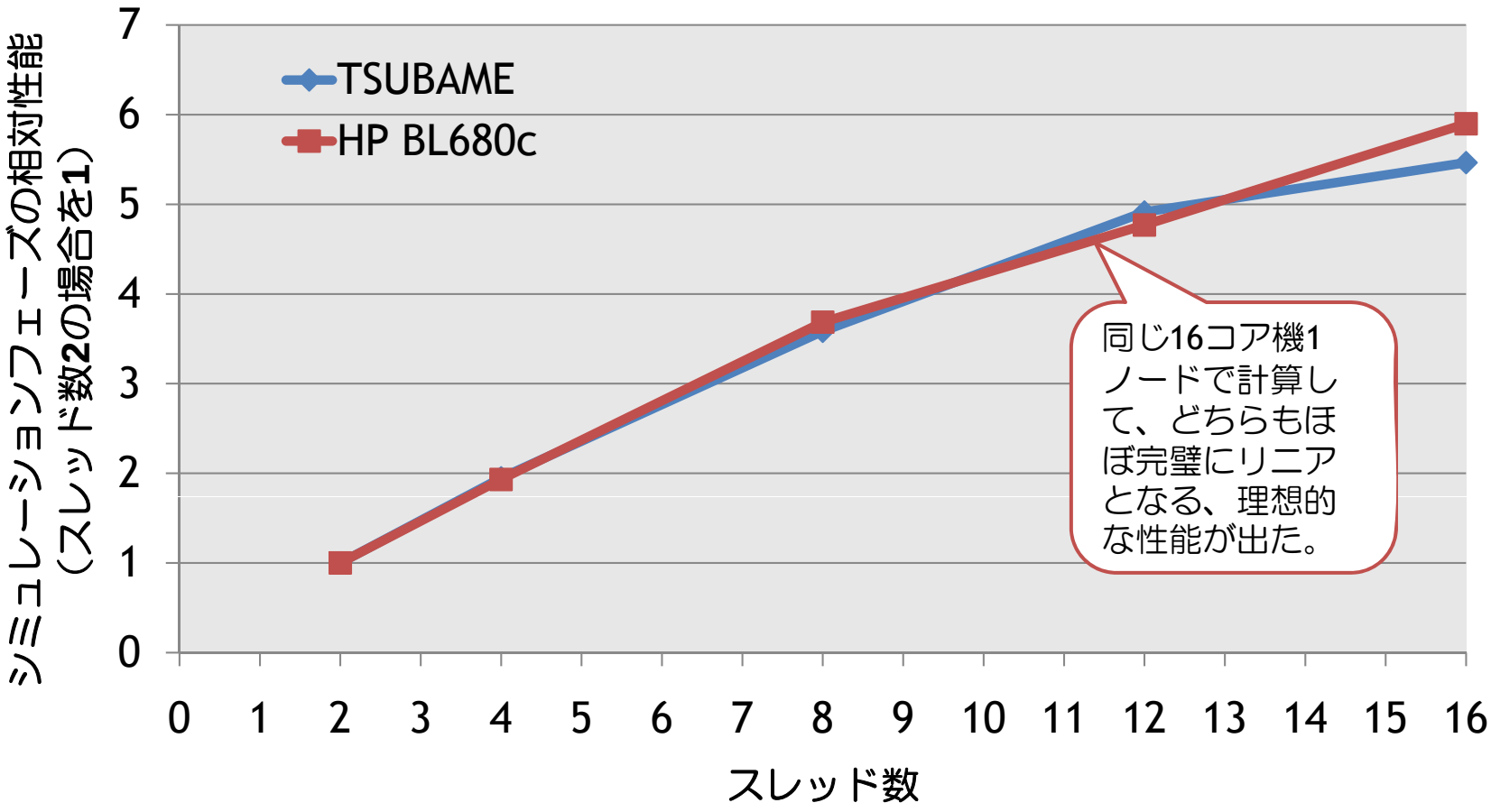
最大231ノードに1対1でマップしたプロセスを走らせて、疎結合の並列処理とした。だから現状はInfinibandの帯域にもレイテンシにも頼ってはいない。実際、GbEでも同等の性能が出ている。

Copyright (C) Numerical Technologies Incorporated 2008/7/11

- スレッド毎にヒープメモリマネージャ（自社製）を配備
  - 排他制御によるシリアル化を避けるとともに、フラグメントを抑えメモリ局所参照性を高めるため。もともとDigital UNIX 4.1FやWindows 2000で痛い目に遭った、例のSMBバグへの対処として実装したもの。
- コンテナクラスの最適化
  - グローバルなヒープを乱すので標準C++ライブラリのstd::vectorなどSTL由来コンテナクラスはstd::valarrayも含めて極力使用をやめた。
  - 代わりにスレッド毎のヒープメモリマネージャに対するSTL互換のvector, matrix, tensorライブラリを作成し、例えばNt::Memory::MemoryPoolVectorとしている。その上に、例えば<algorithm>を使用した。
- BLAS互換のメモリ配置
  - ただしFORTRANではなくC互換配列に統一している。その上でATLASライクな最適化を施し、CPUストールを回避した。
- CPUストール解析
  - Intel® VTune™を使用した。

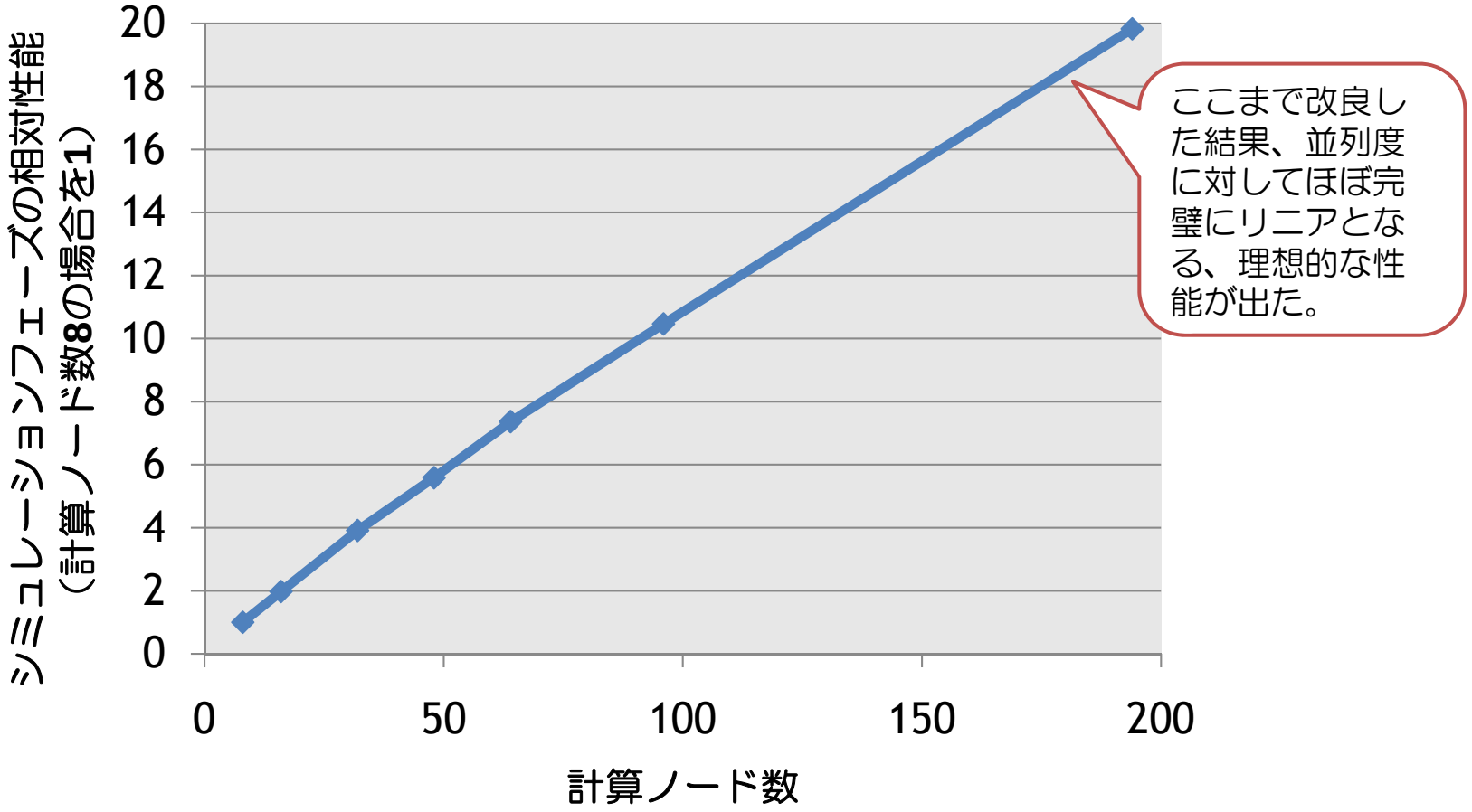
Copyright (C) Numerical Technologies Incorporated

SMP性能はほぼリニア



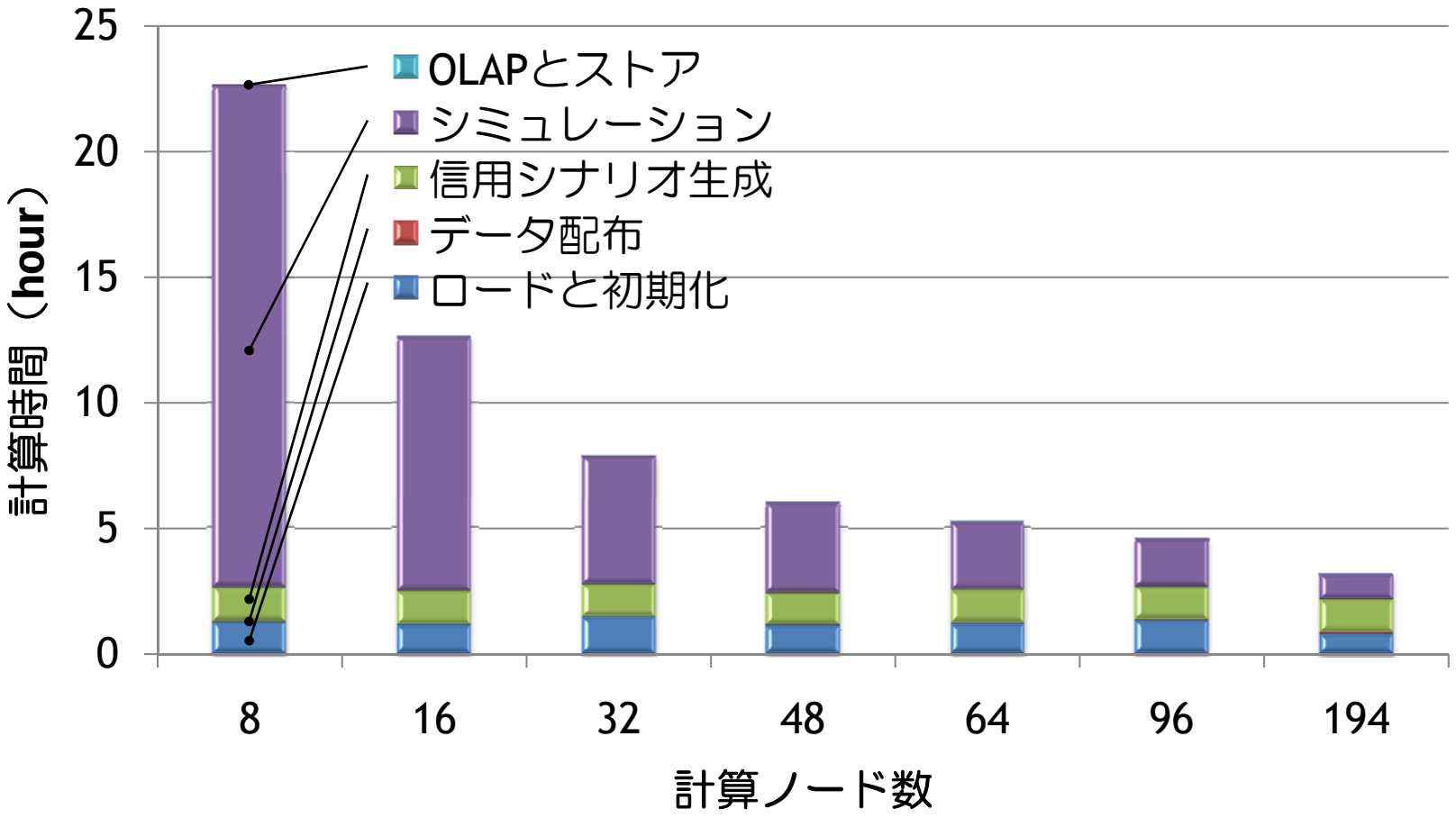
Copyright (C) Numerical Technologies Incorporated

クラスタ性能もほぼリニア

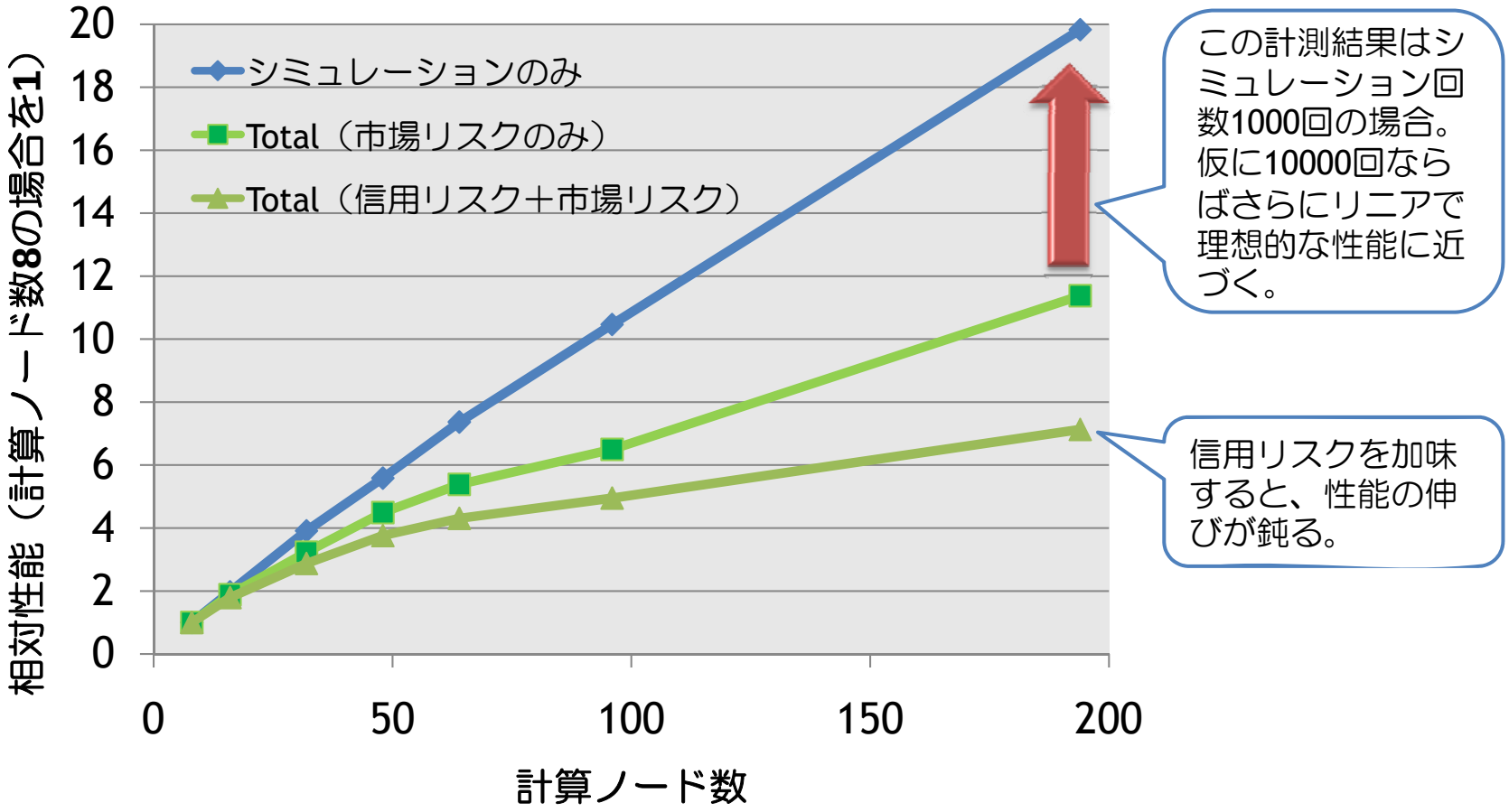


Copyright (C) Numerical Technologies Incorporated

総合性能

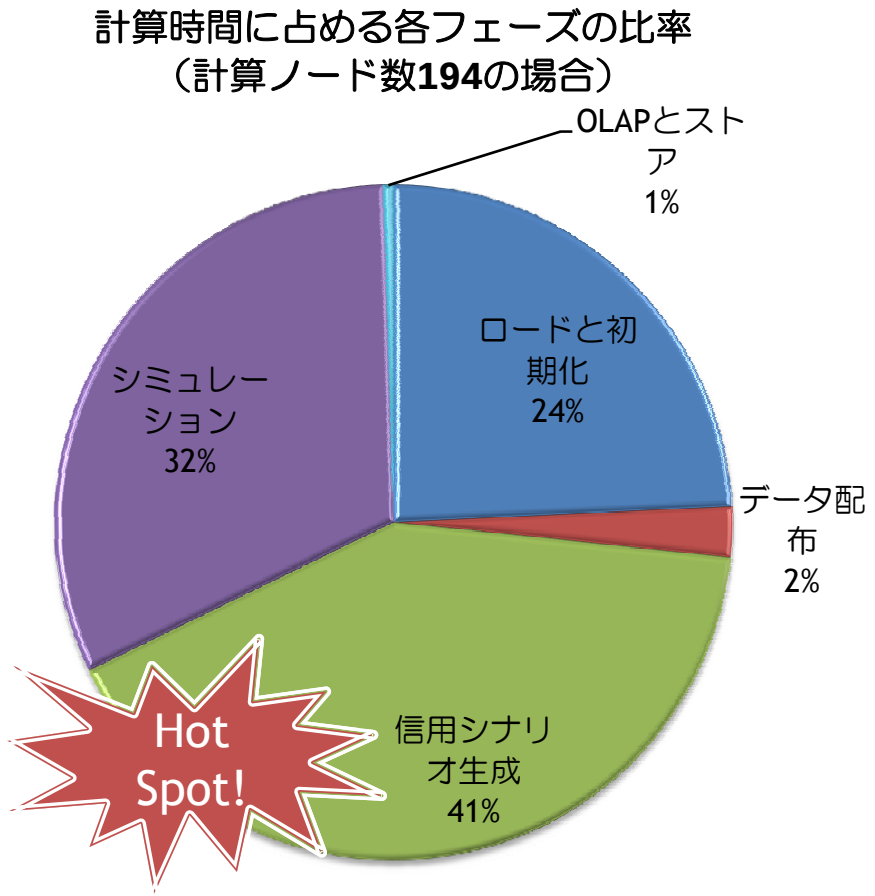


Copyright (C) Numerical Technologies Incorporated



Copyright (C) Numerical Technologies Incorporated 2008/7/11

新たなHot Spotの発見



- 当初の性能ボトルネックであったデータ配布はpipeliningで、シミュレーションはshared nothingなparallelismとコア毎にallocateした密結合のhybridで、それぞれ最適化に成功した。
- その結果、従来はシーケンシャルな処理で済ませてきた信用シナリオ生成が目立つようになった。
- 今期の研究で行う最適化のターゲットである。

Copyright (C) Numerical Technologies Incorporated

先期研究のまとめ

- 200ノード超まで性能がスケールすることを確認。
- 最大規模のメガバンククラスでALM用の詳細なモンテカルロシミュレーションを行う場合に必要なノード数を確認。
  - 市場リスクのみであれば数十ノード
  - 信用リスク、バックアセット込みならば数百ノード
- 新たなホットスポット（信用シナリオ生成）を発見。この部分の作業をやり残した。今年度から本研究は

戦略分野推進：社会基盤のリスク管理シミュレーションへのHPC応用技術の開発
「大規模ALMシミュレーションへのHPC技術の適用」

として採択して頂いており、さらなる成果を目指す。

Copyright (C) Numerical Technologies Incorporated

今期研究の方針

- 信用シナリオ生成のパラレル化
  - 既に着手している。
- MPI or NOT MPI？
  - MPIに依存すれば開発が楽になる。しかしMPIに移行してよいのか？
    - 金融機関の管理会計分野である我々のアプリケーションは製品寿命が長い。10年前に導入されたDECのUNIX上のシステムを現在でもサポートしている。これから導入するユーザーも頻繁なバージョンアップを嫌う点では同じだろう。今のMPIは10年後もサポートされるのか？
    - ノード間IPCに使用している現行のメッセージパッシング方式はフォールトトレラント性を持つ。普及しているMPIはフォールトトレラント性を持たない。MPIを採用すれば製品の信頼水準が損なわれる。
  - MPICH2ベースで部分的にMPIを採用してみようと考えている。
    - オープンソースだからメンテナを我々自身がやると決断すれば長期保守問題は解決する。
    - Windows CCS（Windows HPC Server）との互換性を維持できる。
    - 前項「信用シナリオ生成のパラレル化」をターゲットにする。現行のメッセージパッシング方式とハイブリッドにすることでフォールトトレラント性を維持する。
  - その間にMPIの研究が進むことを期待。

Copyright (C) Numerical Technologies Incorporated

- ジョブスケジューラAPIの公開
  - PlatformやDataSynapseに対抗するわけではないが．．．
  - 同じグリッド環境に後から他社あるいは金融機関自身が別のアプリケーションを追加する時、高価なミドルウェアを購入しなければならないとしたら、単純にもったいない。
  - せっかく既に持っているのだから、リソース管理、ジョブ管理に加えて、ユーザー認証、プロビジョニング、性能モニタ、障害監視、の各APIを公開し他社向けにサポートしようかと考えている。
- パラレルI/Oの採用
  - 現在は、グリッドノード間がGbEで接続され、単一のサーバに直結した大規模ストレージ（つまりノード間のI/O効率が均等でない）を前提にして設計してある。企業内ではこれが一般的な環境だからである。
  - しかし、せっかくLustre/Infinibandがあるのだから、もう少し頼ってもよいのではないか。

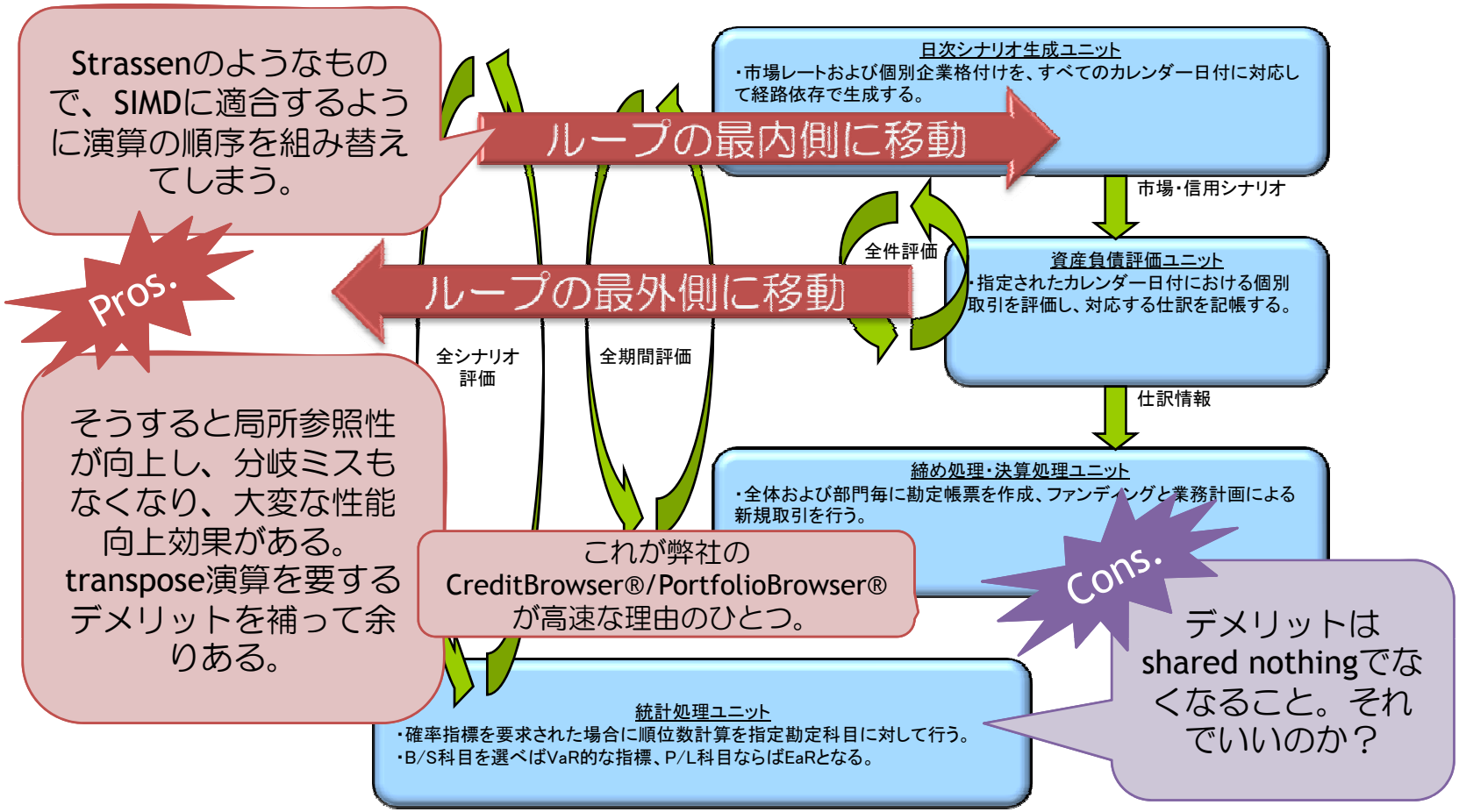
Copyright (C) Numerical Technologies Incorporated

アクセラレータの検討

- GPGPU、あるいはアクセラレータの検討
  - 民間HPCでもトレンドになるのか、それとも学術HPCだけの存在に終わるのか。
  - 弊社も実機を購入して調査を始めた。初見では、CUDAの宣伝文句にあるような劇的なパフォーマンス向上はリスク管理では非現実的。性能向上率は例えば4-5倍ではないか。他方データ量の小さい金融アプリケーションではイールドカーブモデル等でうまくいき、圧倒的な性能向上率を見せる。
  - そうなる理由は、(1)金融系では大きな粒度のSIMD演算が少なく、(2)しかも小さなローカルメモリ（Cell B.E.を例にとれば256k）で処理可能な金融系SIMD演算は限られており、(3)これを超えると少なくともDMA、悪くすればPCI Expressの転送コストがかかるから。
  - それでも4-5倍ならば開発して採算ベースに乗るのかも。実際、弊社では10年前に開発したCreditBrowser®でその種のベクトル化演算が高速化に寄与した実績があるから迷っている（次ページに粒度を調整したSIMD化の例を示す）。

Copyright (C) Numerical Technologies Incorporated

GPGPU Or NOT GPGPU？



Copyright (C) Numerical Technologies Incorporated 2008/7/11

お断り

- 本文書の著作権はニューメリカルテクノロジーズ株式会社に帰属します。
- 本文中で使用されている製品名・社名などは、一般にその所有者の商標または登録商標です。
- ニューメリカルテクノロジーズ株式会社は、本文書の完全性、正確さを保証致しません。いかなる場合においても、ニューメリカルテクノロジーズ株式会社は本文書に関連して生じた通常の直接的、間接的、必然的、偶発的、特別な、あるいは懲罰的賠償について、たとえニューメリカルテクノロジーズ株式会社がそのような賠償が発生する可能性があることを通告されたとしても、なんら責任を負いません。