High-Performance Distributed Solar Computing (?)

--- Towards a Grid that Computes like Trees---

松岡　聡(Satoshi Matsuoka)
東京工業大学学術国際情報センター
matsu@is.titech.ac.jp

Abstract

Power-heat dissipation as well as the associated CO_2 emission are becoming serious bottlenecks in scaling of large supercomputers. Indeed a single day's operation of TSUBAME, the fastest supercomputer in Asia-Pacific circa 2007, incurs as much CO2 emission as an entire Formula-1 race. Instead, the use of photovoltaic power generation is promising to minimize or eliminate the emission altogether. While the traditional methods would incur simple attachment to a power grid, and involve very little effect or merit from grid computing, we actually claim that grids and distributed power generation go hand-in-hand to create a robust and self-sustainable computing infrastructure that could scale to TSUBAME-class applications. For robust operation as a pragmatic operational infrastructure, much continuing research would be required customizing and integrating the results from P2P, autonomic computing, sensor networks, etc.

1. はじめに

消費電力はすでにスパコンなどの大規模計算インフラストラクチャにおいて主要な律速条件となっている。2006年に開発・設置された東工大TSUBAME（図1）は、現在でもアジア最速の85テラフロップス・1.6ペタバイトの性能や規模を誇り、しかも種々の省電力向けの技術適用を行っているが[1]、冷房設備を含めた最大消費電力は1.2メガワットにもなり、東工大全体の消費電力の10%・年間1億円の限界的な電気代負担となっている。さらに、後継のTSUBAME2.0や、さらにその次の後継機種では、Mooreの法則による速度向上が達成されても、リーク電流などの問題で電力消費の相対的増大を招く懸念が大きい。

一方、大規模サーバやスパコンによる大気中へのCO_2排出は増大しつつある。TSUBAMEにおける電力消費を計算すると、わが国におけるCO_2の排出の平均データである：

- **電力　　1 kWh　　0.357kg**
- **ガソリン　1 litre　　2.31kg**

および最大平均消費電力（1.2MW程度))から換算すると、TSUBAMEのCO_2排出は一時間あたり 428kg, 一日では10.3tにも達する。一方、Formula 1 のレースカーでは 1 レースあたり約 200 リットル強ガソリンを消費するので、上記からは車あたり500kg, 22台のレースでは(リタイアを含めて)やはり10t程度のCO_2を排出する。さらに、年間18レースがあると仮定すれば、年間約 350 日程度稼動するTSUBAMEは所謂F1サーカス全体より20倍近いCO_2を排出することになる(予選などを含めても10倍程度となる)。

図1：東工大TSUBAMEの概観

このような消費電力およびエミッションの問題を解決するのに、すでに米国では Power Aware HPC 分野として種々の研究が存在する。松岡がゲストエディターとなったNSF CyberInfrastructure Quarterly ([2])の特集号では、BlueGene[3]や SiCortex[4]に代表される組込系省電力プロセッサを大量に並べるアーキテクチャや、近年の CPU の動的電圧制御(Dynamic Voltage Scaling, DVS)をソフトウェア的に最適制御する Rosenthal, Wu, Cameron らなどの研究を紹介しており、さらに Green Top 500 などのラン

キングも提唱されている。

しかしながら、それらの研究の成果による電力削減効果は高々数%から倍程度であり、例えば携帯電話における省電力化の効果と比較すると極小である。その主な理由は、今までの組込系省電力の場合と異なり、HPC におけるワークロードの違いにある：組込系の用途はリアルタイム&メディア処理中心で、処理は短時間のバーストか、あるいは定型的なストリームメディアである。前者は DVS の適用が容易・有効であり、後者は MPEG でコーダなどの省電力化した専用ハードウェアを適用しやすい。一方 HPC では、常時計算パワーを要求され、かつ計算内容はアプリにより多岐に渡る。よって、組込み系の省電力化手法の直接的な HPC への適用の効果は限定的であることが最近の研究により判明している。

一方、CO_2排出を根本的に減らす方法として、風力・地熱・潮力などによるカーボンサイクルに頼らない発電法が旧来より提唱されている。その中でもっともポピュラーなものの一つが、太陽光発電[5]であり、太陽電池パネルの高効率化・低コスト化などの技術進歩により、さまざまな状況において用いられるようになってきている。

そこで、本稿では太陽光発電を用いて、CO_2排出を行わない、いわゆる ゼロ・エミッション(Zero Emission) HPCの可能性に関して探究する。その根本原理としては、従来手法と異なり分散型の発電を行い、「(発電された)エネルギーを伝送せず、(その場の計算によって生じる) 情報を伝送する」ことであり、それによりエネルギーロスや機器設置の制限、さらにはコストや冷却の問題を解決する。用途は現状でDesktop Gridで代表される逐次か小並列のアプリのパラメタサーベイ型の計算に主に限定されるが、実は多くのスーパーコンピューティングでのアプリ種別をカバーしており、従来型のスパコンを高度な同期や通信が必要なアプリのみに集中させることができる。簡単な試算では、現状の 2007 年のテクノロジを用い、東工大TSUBAME[1]クラスの性能をゼロ・エミッションを実現するのに、2500m²程度(つまり、野球場の 1/4)程度の設置面積で実現でき、しかも電源やネットワークケーブルなどを一切用いずに「木を植林するように」設置することが可能となることが判明した。さらに、大規模な設置コストやメンテナンスコストがなくなり、よりコモディティのパーツが用いられるのでTSUBAMEなどのハイエンドサーバを用いたスパコンのアーキテクチャと比較して、その総コストは数分の１となることもわかった。さらに、専用の「ソーラー計算タイル」を開発することにより、さらに環境に計算を「埋め込む」ことが可能であるが、これは別項に譲る。

これらの技術は Weiser らが過去に提案した所謂埋め込み型のユビキタスコンピューティング[6]と似ているが、HPC で多大なる計算やネットワーク負荷が生じる大規模並列利用が行われる分、新たな技術的チャレンジが大きい。実際、Peer-to-Peer (P2P)、センサーネットワーク、デスクトップグリッドなどの種々の技術を適用することにより、はじめて運用可能なインフラとなる。しかしながら、単純な部分部分の適用だけでは困難で、今後インフラとしての全体的な性能・消費電力・コストモデルを含めて、種々の研究を行う必要性がある。

2. Tree Grid: 分散型太陽光発電による大規模分散並列計算インフラの提案

全地球環境としては、常時 90 ペタワット程度の太陽エネルギーが降り注ぐことが知られており[5]、太陽光発電はそれを有効に、CO_2ガスの排出なく直接利用できることから、近年代替エネルギー源の強力な候補として着目されている。太陽光発電全体に関する技術詳細は多く研究・発表が行われ、インターネットの各種サイトでも豊富にデータがあるので、ここではその議論は大幅に割愛し、必要最低限の関連知識を述べる。

通常の太陽光発電の種別化として、基本的に電力網(power grid)に接続する場合と接続しない形態(distributed power generation)がある［5]。家庭やオフィス・工場などの、通常の大規模インフラ上の電力需要に対する太陽光発電の利用では、前者を用いる場合がほとんどである。それは、太陽光発電は発電電力のピークがあり、かつ当然昼間にしか発電はできないので、サイトにおける発電と消費の全体のバランスを平均化するために、電力網を実質的なバッファとして利用することになる(図 2)---昼間は余剰電力を売電し、夜中は(安価な)夜間電力を消費する。(また、雨天時などの発電量不足の状況にも自然に対応する。)

通常のスパコンは 24 時間連続稼動するため、通常は当然ながら電力網接続による太陽光発電を考える。しかしながら、TSUBAMEクラスのスパコンになると、設置面積に対する実際の発電量は、全体の消費電力と比較すると微々たるものとなる。実際、TSUBAMEは 300m²程度の設置面積で 800kwの電力を消費するが、高効率の単結晶シリコンの太陽電池ですら 100W/m2 程度の発電量なので、可能

な総発電量はせいぜい 30kWであり、実際の建屋で利用できる設置面積では 100m²以下なので、実際のピーク発電量はせいぜい 10kWとなり、ベストでも 1/40 以下となる。

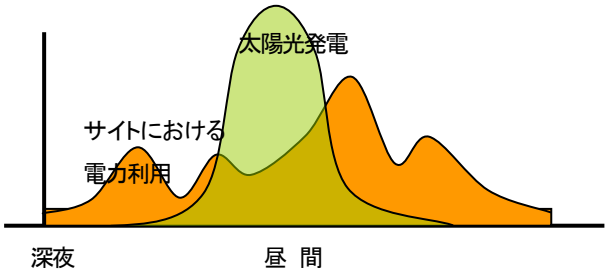

図 2. 電力網接続による発電と電力消費の平均化

より多くの電力が必要な場合は、通常の発電所のように、遠隔地に集中的に太陽光発電設備を設置することも可能ではあるが、その場合は高電圧に DC/AC 変換などを行う必要があり、4-12%程度と送電ロスが大きく、またインフラの維持費がかなりかかることになる。よって、電力網接続の場合は、通常の太陽光発電所以上のメリットや、情報インフラとしてのメリットは基本的にはない。これは、基本的に計算機を冷蔵庫や照明など、固定された電力消費を行うインフラとしてのみ捕らえ、その分散性などの性質を全く考慮していないことに起因する。

そこで、我々は情報的な性質をフルに活用し、効率が良くかつ設置コストや維持管理コストを本質的に最小化できるような分散型の太陽光発電による計算インフラ Tree Grid の構築を提案する。その本質的な性質としては、計算における局所性の原理を最大限に活用し、かつ必要なときには電力ではなく情報のみを、メッセージとして、超省電力・自律構成型・超近接網の無線ネットワークを通じて伝達させることにより、エネルギー自身の消費を局所化し、電力網の利用を基本的には一切なくして、太陽光発電のみで駆動されるインフラとすることにある。太陽光のサイクルによる発電の有無や、突発的なシステムの故障などには、基本的に P2P グリッド技術[7]を用いて、自律的(autonomic)にシステムからアプリケーションの実行形態をリアルタイムに適合していくことによって、信頼性を上げる。逆に、そのようなインフラの変化にも対処し効率良く実行できるアプリケーションの実行に絞ることにより、ハードウェアやソフトウェアの設計を超消費電力・低コスト・かつ容易かつアドホックな分散設置を可能とする。

これはいわば樹木が分散して自然発生的にマスに成長し、光合成を行いエネルギー活用をすることに似ており、我々はそのような理由で"Tree Grid"というメタファーを与えている。

3. Hayashi-1: Tree Grid プロトタイプノードによる実験

以上のコンセプトの初期の妥当性を検証するために、局所的な太陽光発電のみで駆動され、外部と無線(801.11g)のみで通信し、アドホックな分散並列計算機を構成する Tree Grid のノードとなるべくプロトタイプ Hayashi-1 を構築した。Hayashi-1 のスペックは以下の通りである(図 3)：

- Hayashi-1 ノード計算機(PC)
  - ベース：Sony VAIO U70
  - Intel Core Solo 1.0Ghz (2G Flops peak), 512MB memory
  - **High Performace Linpack 実行時実計測 16V 0.8A, < 20W Linpack**
  - HDD: 30GB (1.8in)+ 160GB HDD (2.5in)
  - 無線 LAN (802.11g/b)
  - Microsoft Windows XP
  - 電池によるバックアップ駆動：約 2 時間
- Hayashi-1 ソーラーパネル（太陽電池）
  - 多結晶シリコン型 41.2cm x 66.6cm + 電圧制御装置
  - 最大発電能力 30W, 1.77A/17.0V
  - 重量 3.4kg
- 非接触型 DC 電力計(リアルタイム計測用)
- 全体重量：約 4.5kg, すべての機器はソーラーパネルの裏面に装着
- 製造コスト：ノードあたり約$1500 US.

上記のスペックの装置を二台分作成し、最小構成(2 ノード)のクラスタを構築し、802:.11g のアドホックの無線ネットワークで接続できるようにした。2007 年 6 月末の晴天日の夕方、東京都目黒区の東工大大岡山キャンパスの夕方 4 時ごろの太陽光でも、並列 Linpack を実行しているノード駆動し、かつ計算ノード内のバッテリーを充電するのに十分な太陽光発電が行え(16V 以上・1.2A 程度)、安定した動作を示した。また、Linpack 自身も安定した性能を示した。

図 3: Hayashi-1 プロトタイプノード

プロトタイプは2006年初期の省電力CPUのテクノロジを用いたので比較的低性能であったが、2007年のテクノロジを用いれば遥かに高性能の Tree Gridのノードのプロトタイプ Hayashi-2 を構成可能である：

Hayashi-2: ULV Core 2 Duo 1.2Ghz in Sony VAIO VGN-TZ90S: 4.8Gflops (Peak) ~ ほぼ東工大 TSUBAME の個々の CPU コアと同等

- 個々の CPU の Linpack 等の性能値はほぼ同等
- 最大 メモリ 2G Byte
- GbE ネットワーク, PCI-E (express card)など
- 32GB Flash + 160GB HDD
- 重量~1kg,
- 電力消費は U70 とほぼ同等

よって、メモリ・ネットワーク・I/O のバンド幅に敏感ではないアプリケーションに関しては、個々のユーザにとっては TSUBAME との区別化は困難であるので、結果として 10,000 ノードのシステムにより、それらのアプリケーションに関してはほぼ TSUBAME と同等となる

このようなTSUBAME規模のHayashiクラスタの設置面積はどの程度であろうか。上記のHayashi-2 プロトタイプからすれば、20GigaFlops/m²が発電量に対する計算密度となる。以下のパラメタを鑑みると

- **5000 ノード/48 Teraflops**
- **20 TeraBytes Memory,**
- **1.6PetaBytes Storage**
- **すべて太陽光のみにより駆動**

であるので、約 50m x 50m、即ち野球場の 1/4 程度の総合面積で実現が可能である。この際の多結晶シリコン太陽電池によるピーク発電量は 150kW で、平均では 100kW の電力を消費する(ちなみに、これは TSUBAME の実際の消費電力の 1/10 程度である)。また、先に述べたように、その設置は一箇所に集中している必要はなく、図 4 にあるように適切に分散設置することが可能である。その際には各ノードは無線で近接通信することとなり、大量のノード間の共有利用の競合により全体のバンド幅は大幅に低下し(801.11n でも 100Mbps 程度なので、数百ノード規模で 1 チャンネルを共有したとして 100-200kbps)、その適用可能なアプリケーションは基本的にはデータ利用が局所化されるパラメタサーベイや最適化、アンサンブル計算等に限定される。幸いなことに、それらのクラスの計算は、TSUBAME のプロファイルでは数十%を占めることがわかっており、有効にインフラとして機能することが期待される。ISV アプリケーションでは Gaussian, Amber などである。また、Stanford Folding@Home などでも、同様に分子動力学のアプリにおいてスケーラブルな性能が得られることが報告されている[8].

図 4: 分散設置された Hayashi-1 ノード

4. 実際の大規模 Tree Grid の運用用件と技術

実際のスパコンのアプリケーションの中で、マジョリテ

ィな割合を占めるアプリケーションのクラスにおいて Tree Grid は有望なインフラとなりうることを議論した。しかしながら、実際のスパコンは言うにおよばず、desktop を集約したグリッドと比較しても、障害や不安定要因、さらには分散型の太陽光発電自身に起因する阻害要因が多々存在する。それらの解決のためには、実際の運用や障害モデルをきちんと立脚し、それに合わせて P2P や autonomic 系のアルゴリズムやシステムを適合していくことが今後の研究の課題となる。ここでは、その概念設計レベルの要求等を議論する。

運用モデルは以下の通りである

- 各ノードはビルの屋上・壁面から、半専用のオープンスペースなど、十分に太陽光が確保できる場所に、ある程度の相互間密度およびクラスタリングの性質を保ちながら分散配置される。
- 基本的に各ノードには電力網からの電源供給はなく、太陽電池パネルとバッテリーによって分散発電を行う。
- ノード間は、UWB など、基本的にノードの追加や欠落時の自己組織化が可能で、高速かつ近接の無線通信網によって通信が行われる。(一部、制御を行うノードや、外部との通信 proxy のノードは有線接続され、かつ外部給電される可能性もある)。
- システム全体は完全に自律的に動作し、ノード追加や故障時のノード排除なども自動的に行われる。人間のメンテナンスが入るのは比較的長い間隔の定期のものだけである。
- 全システムは 24 時間動作しうるが、当然一日の太陽光のサイクル、および天候などでその動作は左右される。発電量が一時的に十分でなくなったとき、あるいはある程度の夜間時の動作を確保するために、各ノードは(ラップトップと同等の)バッテリーを備える。また、電源効率を最大にするために、通常のスパコンと比較して遥かにアグレッシブな電力制御を行う。
- 夜間時には、大量のノードが休眠状態(hibernation)となる。多くのパラメタサーベイ型のアプリケーションはスループット重視なので問題は少ないと見られるが、ある程度の turnaround time が必要なアプリケーションは(地球の裏側の)昼間時のノードや、あるいは電力網給電が行われている計算インフラに動的にマイグレーションする必要がある。

このような運用を実現する技術要素としては、

- Automatic Node Detection & Registration (Self-organization)[9]
- Dependability, Frequent Node Failures (Self-healing)
- Robust distributed security
- Distributed and Scalable scheduling
- (Very) Ad-hoc (Wireless & High-Bandwidth) (Overlay) network organization
- Distributed power & energy control

などが挙げられるが、これらは基本的には P2P、センサーネットワーク、低消費電力 HPC[2], デスクトップグリッドなどで培われてきたものである。必要なことは、これらをいかに Tree Grid 向けに技術適合し、全体のシステムの設計をしていくか、である。

5. まとめ：Tree Grid に未来はあるか?

P2P や自律コンピューティングの技術を駆使し、電力網に頼らない分散型の発電を行うことによって、ゼロ・エミッションのスーパーコンピューティング環境を実現する Tree Grid の提案を行い、その妥当性や必要な技術開発の検討・検証を行った。今後は、それぞれの技術要素の適合性を検討し、より実システムに近いプロトタイプを作成するとともに、全体のエネルギー収支を含んだシステム構築および総合ランニングコストのトレードオフをより綿密にモデル化し、システムの真の妥当性を検証していく必要がある。

参考文献

[1] Satoshi Matsuoka. "The Road to TSUBAME and Beyond", in Petascale Computing: Algorithms and Applications, David Bader (ed.), Chapman & Hall / CRC Press, 2007 (to appear).

[2] Satoshi Matsuoka. "Low Power Computing for Fleas, Mice, and Mammoth --- Do They Speak the Same Language?", NSF CTWatch Quatery,, Vol. 1 No. 3, pp.2-11, Aug. 2005.

[3] George L. Chiu et. al. Blue Gene/L, a System-On-A-Chip. Cluster 2002, IEEE International

Conference on Cluster Computing. IEEE Computer Society, September 2002.

[4] Matt Reilly, Lawrence C. Stewart, Judson Leonard, and David Gingold. "Sicortex Technical Summary---White Paper", December 2006, http://www.sicortex.com/

[5] "Solar Power", Wikipedia Free Encyclepedia, http://en.wikipedia.org/wiki/Solar_power, 2007.

[6] Mark Weiser, "Some Computer Science Problems in Ubiquitous Computing," Communications of the ACM, July 1993.

[7] Cécile Germain, Vincent Néri, Gilles Fedak and Franck Cappello. "XtremWeb: building an experimental platform for Global Computing", Proc. IEEE Grid 2000 Workshop, Dec. 2000, IEEE Press.

[8] Folding@Home, http://folding.stanford.edu/, 2007.

[9] Stoica, I., Morris, R., Karger, D., Kaashoek, F. and Balakrishnan, H. "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications", SIGCOMM Conference, San Diego, CA, USA, 2001, The ACM Press.