＜特集：炉物理研究へのPCクラスタの利用・PCクラスタの構築＞

PCクラスタの構築

（株）グローバル・ニュークリア・フュエル・ジャパン　中隆文
takafumi.naka@gnf.com
池原正
tadashi.ikehara@gnf.com

1. はじめに

グローバル・ニュークリア・フュエル・ジャパン(以後、GNF-Jと略します。)では、LinuxベースのPCクラスタを構築し、分散、並列処理を必要とするエンジニアリング業務に利用しています。構築方法さえ間違えなければ、非常にコストパフォーマンスの良い、スーパーコンピュータクラス[1]の性能を入手する事がPCクラスタで可能になります。

本稿では、PCクラスタについて

1. なぜ、GNF-JでPCクラスタを構築するに至ったか、
2. PCクラスタハードウェア選択時に考慮すべき事
3. GNF-JのPCクラスタの紹介
4. まとめ-将来に向けて考える事

等といった事を紹介していきます。

図1：GNF-J本格クラスタ全容写真

2. なぜPCクラスタ導入に踏み切ったのか

コンピュータの処理をより素早く完了させることを考えてみましょう。それには、

① プロセッサー単体の速度を上げる

② プログラム中で独立に処理できる部分を複数のプロセッサーにて同時並行処理する

などの方法があります。ベクトル計算機にしろ、並列計算機にしろ、これらは同時並行処理という点で類似の方法であり、ベクトルスーパーコンピュータや超並列計算機と呼ばれるものは、これらの究極の姿と言えます。しかし、あまりに高価なシステムですので、コストと性能両方を重視するならば多くの場合、これらは最適解にはなり得ません。

では使い勝手がよく、研究・設備予算で無理なく賄え、実戦力となり得る実用的な計算機システムとはどんなシステムでしょうか。これは計算機システムをとりまく技術の変遷と共に変化してきました。

---

[1] **スーパーコンピュータクラス：**コストパフォーマンス良く構築出来た時点でそれを「スーパー」と呼んで良いのかという疑問はわきますが、5年程前なら世界のトップクラスであったスーパーコンピュータの性能が非常に安価に入手出来るという意味で、ここではこのように表現しました。

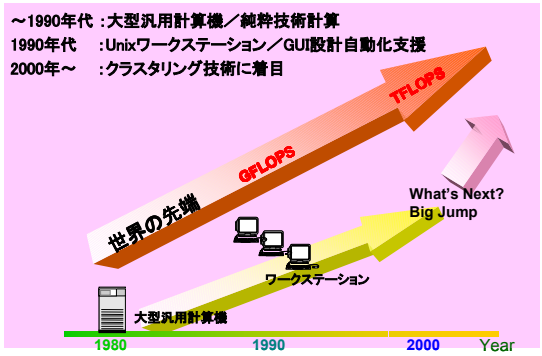

図2:計算システムの変遷

中でも1980年代後半から市場に出回り始めたUNIXワークステーションが科学技術計算分野に果たした役割は計り知れないものがあります。マルチユーザ・マルチプロセスや優れたネットワーク環境、GUI環境を提供してくれるUNIX OSは、従来の大型計算機に比べて理解し易く、変更が簡単で移植性が高いという優れた特徴をもっています。筆者らも、単なる解析コードの開発の枠を越え、設計作業全体を自動化するようなシステム開発やWSの導入計画を始めとする計算機システム自体の設計などに関わってきました。このように多面的にUNIXワークステーションに接する過程で、ネットワーク接続されたWSを複数結合し相互通信させることでシステム全体を一つの高速計算システムとして機能させるという夢は、誰しもが自然に抱くクラスターの着想だと思います。

実際、UNIXに少し習熟してくれば、複数のプロセスを起動し、プロセス同士通信させながら協調して一つのジョブを行わせるようなプログラムを書くことができます。また、実行ジョブにプライオリティをつけたり、ネットワーク上の他マシーンにジョブを投入したりすることも容易です。データの共有に関しては、共有メモリを使用してプロセス間でメモリー上のデータを共有したり、リモートディスク上のデータをディスク・マウントにより相互アクセスさせることができます。

このようにUNIXの世界では並列計算で求められるホスト間データ通信やプロセス間の同期制御、データの共有手段が提供されています。また負荷分散については至って簡単で、各ホストに独立な仕事振り分けるためのリモートシェル環境で十分と言えます。恐らくUNIXにちょっと習熟したユーザであれば、並列計算環境や負荷分散環境実現に向けた概念設計図くらいは頭の中に描くことができるでしょう。

続いて世の中の動向です。複数のプロセッサーを用いより速く計算するためのコンピュータ技術には、筐体内に複数のプロセッサーを内蔵させメモリーやI/Oサービスを共有する方式と各プロセッサーが独自にメモリーやI/Oサービスを持つ方式があります。前者の例はSMP[2]サーバーとして知られており商品名にもなっています。一方、クラスターと呼ばれるものは後者に属します。どちらを選ぶかは利用方法にも依りますが、数十から数百台数のプロセッサーを同時利用する、しかも低コストで、という条件を課すと現時点での最適解はクラスターとなります。さらに運用中の構成機器の少数台の故障を許容できる様に構築すれば、構成機器に対し過剰な信頼性を求める必要もありません。そして、いくら故障を許容すると言っても、OSを始めとするソフトウェア部分には十分な安定性、信頼性が必要となります。こうして行き着いた解がPC Linuxクラスターです。

Linuxクラスターの実例としては1993年NASAで生まれたBeowulf型が有名です。これは特殊なソ

[2] **SMP:** 対象型マルチプロセッシング(Symmetric Multi Processing)の略。複数のCPUがメモリを共有するタイプのマルチプロセッサシステムであり、現在のほとんどのマルチプロセッサEWSあるいはPCはこのタイプとなる。

フトウェアやネットワーク構成、またはOSのチューニングに頼ることなく極普通のLinux PCを複数台ネットワーク接続することで高速な並列コンピュータを構築する技術です。その気になればLinux OS、GNUコンパイラー、MPI等の並列計算ライブラリーは無償で入手することができます。勿論、有償のコンパイラー製品の中には開発環境一式（デバッガー、自動並列機能、バッチ管理システム）をパッケージングしたものもありますし、最近ではクラスタ構築のためのコンサルテーション提供会社もあります。このようにLinuxクラスターは一部の専門知識をもった人だけのものではなく、広く門戸が開かれた技術と言えます。

我々の構築したPC Linuxクラスターもこの Beowulf型です。何と言っても、ユーザから見た使い勝手の良さと柔軟性の高さが最大の特徴です。つまり通常のシングルCPUのLinux計算機としても機能しますし、並列プログラムに対しては並列コンピュータとして、またパラメータサーベイ計算のように膨大なケースを実行する場合には負荷分散クラスタとして機能します。

以下では、こうしたPC Linux クラスター構築の実体験を通し得られたシステム構築プロセスについて勘所も含め紹介したいと思います。

3. ハードウェアの選択

PCクラスタのハードウェアの選択において、留意すべき点はいくつかありますが、決まりきった方法は無いと考えて良いと思います。予算、設置場所、利用可能な電源容量、計算の種類、ネットワークの状況、人件費をどう考えるか、将来の拡張やリプレースの計画など、個々の計画をとりまく条件は千差万別です。また、その最適解も時とともに変化します。そして、そこからベストな設計を自分で見つけ出す事が出来る点がベンダー任せのスーパーコンピュータ購入には無い、面白さであり、また、同時に難しさとなります。

PCの拡張性のおかげで、PCクラスタ自身も改良、拡張が比較的容易です。利用状況の変化に伴い、クラスタ自身を改良していく事が比較的容易に行えます。例えば、当初はネットワーク周りにはお金をかけず、PCに標準で付いているNIC[3]を利用して構築。その後、必要になった時点で高速な(そして高価な)NICにグレードアップするといった計画が可能となります。一般にPCの周辺機器価格は時間とともに低下しますので、トータルのコストを考えた場合も、各種拡張は必要になった段階で実施した方が、安くあがります。これも、ベンダー任せのスーパーコンピュータではなかなかありえない話です。（高価なスーパーコンピュータを購入しても、実用的な並列計算アプリケーションの開発がなかなか立ち上がらず、せっかくの高速通信機能が殆ど利用されずに遊んでいる例も多いのでは無いでしょうか？）

PCクラスタの構築にあたっては、頭の中を柔軟にして自由な発想で構築する事をお勧めします。(ただし、なるべく特定のスタイルに拘らず一般的に記述するつもりですが、やはりGNF-Jが構築したクラスタシステムの構成に引きずられてしまう事をご容赦下さい。ユニークなクラスタのアイデアの

[3] **NIC :** Network Interface Card （ネットワークインターフェースカード)の略。PCをネットワークに接続する為に利用する為に用いるインターフェースカード。マザーボードに最初から備え付けられている場合など、必ずしも物理的形状がカード型であるとは限らない。

出現に期待します。）

3.1.　パイロットシステム

本格的なPCクラスタを構築する前に、小規模なパイロットシステムを構築しノウハウを蓄積する事を推奨します。その上で実際にアプリケーションを実行して本格クラスタの性能が予測できます。また、本格クラスタ構築時のハードウェア選定の精度が高まり、より投資効果を高める事が可能となります。

さらに、ノウハウの蓄積以上に大きいメリットは、周囲の人を説得する効果です。以前に比べれば随分と減っているとは思いますが、今でもPCハードウェアの信頼性あるいはフリーソフトを中心としたソフトウェアの信頼性に不安を持つ人は多い様です。こういった不安に対しては、パイロットシステムとは言え、実際に構築した経験というものが説得力を持ちます。

本格クラスタ導入時にその一部として使うことを前提にパイロットシステムを考えても良いでしょう。もしパイロットと本格クラスタの間に時間が経ちすぎて、スペックの乖離が激しく一体のクラスタとして運用するのが困難な場合でも、OSを入れ替えて別の用途に転用する事が容易な所がPCクラスタのユニークな特徴の一つです。[4]

パイロットの構築にあたっては、「専門家のコンサルテーション」や「入門用小規模PCクラスタ購入」等が役に立ちます。PCクラスタについてコンサルテーションをしてくれる業者は幾つか存在しますが、GNF-Jは「(株)ソフテック」(http://www.softek.co.jp)のコンサルテーションを受け、大きな成果を得ることが出来ました。購入してすぐに使える小規模クラスタの例としては、ぷらっとホーム(株)が出している「PC-Cluster Package」製品があります。(http://www.plathome.co.jp/solution/service/cluster/index.html)

3.2.　基本構成

ここでは、PCクラスタの基本構成を整理していきます。現在一般的なPCクラスタの主要構成は以下の様になります。ちなみに、ここに挙げたものはあくまで一例です。様々な工夫をこらした構成のPCクラスタが存在します。メーカー、常識の枠に捉われずに自由に構成出来るのもPCクラスタの利点。いろいろ考えてみると面白いでしょう。(図3参照)

A:　計算ノード

---

[4] **転用**:クラスタから解列した後のPCがOSを入れ替えて様々な用途に転用可能である点はPCクラスタの大きなメリットと考えています。最新スペックのPCでクラスタを構築すると、数年経過した後でも通常の事務用途のPCとしては十分な性能を持っています。
万が一…PCクラスタが(性能が思ったように出ない等)失敗しても…様々な用途で利用可能です。失敗した場合のロスが相対的に小さい事もPCクラスタの魅力の一つだと思います。

PCクラスタの主役、実際の計算を行う主役達です。

B： マスターノード

計算ノードを統括管理するまとめ役です。バッチシステムの統括などを行います。

C： ファイルサーバ/データサーバ

計算に必要なデータを蓄えておく役目です。

D： ネットワーク

A～Cをつないで、データを交換します。ある意味影の主役です。

E： その他のハードウェア機器

ラック、コンソール切替器、マシンルームなどです。

3.3. 計算ノードの選択

PCクラスタの構成の中心となる機器です。この選択でクラスタの性能の6割方は決まってしまいます。考慮すべきポイントは以下の様なものです。

- 機器の信頼性
- CPUの選択
- メモリ
- ハードディスク
- ネットワークインターフェースカード
- 筐体のタイプ
- その他周辺機器

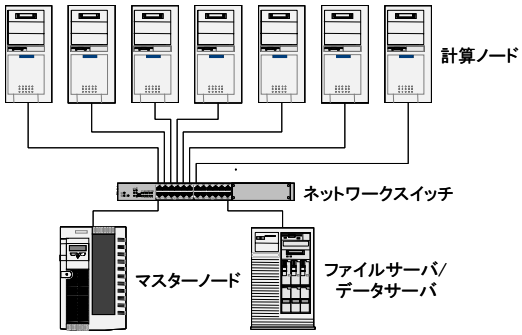

図3：PCクラスタ基本構成機器

3.3.1. 機器の信頼性

機器の信頼性については十分な考慮が必要です。信頼性が不十分ではクラスタがうまく機能しませんし、逆に過剰ですと折角のコストパフォーマンスが悪くなります。コストパフォーマンスを重視するあまり、信頼性の低い機器でPCクラスタを、しかも単一故障が許容されない設計で構築すると、頻繁に故障し、ユーザの計算が何度も損なわれ実用になりません。個々の機器の故障率はかなり低くないとクラスタはまともに動かないので、注意して下さい。故障頻度の目安はAppendix　Aを参照して下さい。

計算ノードはPCクラスタ上で最も数の多い機器ですので、故障率の低い信頼性の高い機器を選ぶことが重要となってきます。しかし、過剰に信頼性の高い機器を採用すると、コストパフォーマンスが悪くなってしまいます。必ず信頼のおけるベンダーから購入して下さい。くれぐれも自分でパーツから組み立てるような事は避けて下さい。手垢が原因の長年の間に進んだ酸化膜や、組立時に残った歪などが原因の故障が数年後にかなりの確率で発生することになります。どうしてもそうする必要があるなら、良く作業方法を工夫する必要があります。

結論としては、計算ノードには後述の各種性能面も鑑みて、ワークステーションクラスあるいはサ

ーバクラスのハードウェアが良いと思います。

3.3.2. CPUの選択

CPUの選択にあたっては、「CPUの種類」、「クロック」、「シングルプロセッサなのかマルチプロセッサなのか」などを考慮する必要があります。

CPUの選択の先立って、CPUの価格性能比について調査をしました。これについては、「Appendix　B」をご覧下さい。結論として、IntelのPentium4 CPUは価格性能比が良く、単体CPU性能でも最高速のEWS用チップに若干の見劣りがする程度ですので、現時点では最もPCクラスタに適したCPUと言えると思います。

クロックについては、アムダールの法則[5]の問題もありますので、よほど並列化効率が良い問題を解くのでなければ、価格性能比が良くても遅いCPUは選択しない方が良いでしょう。

“並列化率50%のプログラムの並列加速率はCPUをどれだけ用意しても倍の速度では動きませんが、ムーアの法則によれば、18ヶ月待てば倍の性能のCPUが手に入る。”事を忘れないで下さい。

次に、シングルプロセッサを選ぶかマルチプロセッサ(SMP)を選ぶかですが、Pentium III, Pentium 4 等のIA-32アーキティクチャのCPUはマルチプロセッサでの処理速度の向上率がEWS用のCPUに比べて悪いので注意が必要です(「Appendix　C」参照)。もし、実行したいプログラムが既に存在するならば、SMPマシンでパイロットシステムを構築し、シングルプロセッサ環境に比べてSMP環境ではどれだけ性能が劣化するか実測すると良いでしょう。もし、SMP環境でもの計算性能劣化がわずかであることが確認出来れば、今度は逆に省スペース、台数減による管理コストの低減、CPU当たりの消費電力低下などマルチプロセッサ機のメリットを生かすことが出来ます。(有償のパッケージを利用する場合は、その料金も半分になりますね。)

4CPUクラス以上のSMPマシンは性能がリニアに伸びない事と、価格性能比が極端に悪くなる(4倍程度。Appendix　B参照)のでおすすめしません。

これらの要素を鑑みて、現在における結論としては、なるべく高クロックのPentium 4あるいはXeon CPUのデュアルプロセッサマシンを選択するのが良いという事になります。

3.3.3. メモリ

PC上で数値計算を実行する場合、性能はCPUクロックに比例しません。これは、CPUのクロックと周辺機器のクロックの差が大きいために発生する現象です。特にIA-32アーキティクチャのPCでは、メインメモリのアクセス速度がボトルネックとなる傾向があります。従って、出来るだけ高速なRAMを

---

[5] **アムダールの法則**:プログラムには並列化可能部分とそうでない部分があるために、並列処理における加速率には限界がある事を示す法則
並列化率:プログラム中の並列化実行部分の割合。(=p)
並列加速率:プロセッサ1台での実行性能に対してN台での実行性能が何倍になったかを表す。(=1/(1-p+p/N) =N/(N-pN+p))
並列化効率:並列加速率を使用しているプロセッサ台数Nで割った値。プロセッサの利用効率の目安。(=1/(N-pN+p))

利用する必要があります。現状では最低でもDDRタイプ、出来ればRDRAMタイプ、しかもなるべく高クロックのものが良いでしょう。標準的なRDRAMはPC800(転送レート3.2GB/sec)ですが、最近はPC1066(4.2GB/sec)タイプの高速なメモリが出回り始めています。

3.3.4. ハードディスク

全体のコストパフォーマンスを上げたいのであれば、ハードディスクにいたずらに大容量のものは不要です。現状HDDの容量は最低でも18GBであり、OSとある程度のテンポラリファイルを格納する用途では十分な容量です。

インターフェースとしては、現状ではIDEかSCSIタイプから選択すると良いでしょう。IDEタイプは最大100MB/sの転送速度を持つATA-100規格が一般的です。最近では最大133MB/sの転送速度を持つATA-133も出てきています。SCSIタイプは最大160MB/sの転送速度を持つUltra160規格が一般的です。最大320MB/sの転送速度を持つUltra320規格も出てきています。

通常のPCで利用されている32bit/33MHzのPCIバス規格は最大転送速度132MB/secですので、それ以上のバス転送速度を持つドライブを利用する場合は注意が必要です。

3.3.5. ネットワークインターフェースカード

ネットワークは非常に大切な要素です。ネットワーク機器の選択と不可分なので、詳しくは「3.6ネットワーク」を見て下さい。

大規模でノード間通信が多い並列処理の場合は、Gigabitクラスのネットワークカード、すなわち1000Base-TやMyrinetを、そうでなければ標準装備の100Base-Tで十分です。1000Base-TはNICの価格がこなれていますが、スイッチングハブの価格がまだ高く多ポートの製品が無いので、1000Base-TのNICを採用し、将来価格が低下してからスイッチングハブのみ交換する方法も悪くありません。

3.3.6. 筐体のタイプ

筐体のタイプとしては、幾つか候補が考えられます。デスクトップ-ワークステーションタイプ、タワー型-ワークステーション、タワー型-サーバ、ラックマウント型-サーバなどです。

各タイプ別のメリット/デメリットを表 1にまとめました。

ラックマウントタイプのサーバは省スペースが実現できますが、単位面積当たりの重量が大きくなるので床荷重に注意が必要です。ラックマウントタイプは、その設置密度の高さから、排熱の密度もかなり高くなります。（まさに

表 1:筐体毎の各種メリット/デメリットの比較

筐体	デスクトップ	タワー	タワー	ラック
タイプ	WS	WS	Server	Server
省スペース	○	△	△	◎
床荷重	○	○	○	×
熱対策	○	○	○	×
CPU性能	◎	◎	○	○
可用性	○	○	◎	◎
拡張性	○	◎	◎	○
価格	○	○	△	△

◎:メリットが大きい
○:クラスタ用途において十分メリットがある。
△:若干不利である。
×:注意して対策する必要がある。

筐体の背後から熱風が吹き出すという状態になります。）そのため、冷却について十分考慮する必要があります。また、最高速のCPUはワークステーションタイプに最初に搭載される事が多いので、CPU性能で見劣りすることも多いです。（ただ、現在はラックマウントタイプの最高性能CPUはデスクトップWSの最高性能CPUと同一の2.8GHzです。）

可用性は故障せずに使える確率であり、信頼性に相当し、高いほど良いと言うことになります。一般にサーバタイプのハードウェアはワークステーションタイプに比べて信頼性が高いと言われていますが、最近ではワークステーションクラスでもPCクラスタ用途には十分な信頼性を持っています。

価格について言えば、一般にサーバタイプはワークステーションタイプに比べて数割割高であることが多いです。（Appendix　B参照）

また、WSタイプとサーバタイプではチップセットが異なってきます。場合によっては、構築したいクラスタの特性を良く考慮し、それぞれの特徴を考えて選択する必要があります。

3.3.7.　その他周辺機器

いかに無駄な投資を避けるか、とにかくちょっとした価格差が台数分に効いてくるので、注意が必要です。（ここは、変わった周辺機器を試してみたいという個人的な欲望と、冷静なコスト意識のせめぎ合いとなります。）

インストール時の便宜性などを考えると、CD-ROMドライブはあった方が使い勝手が良いでしょう。ただ、CD-R、DVD-RAMなどの高機能のドライブにする必要性は無いと思います。価格差が小さければ、将来のOS等の供給メディアが変化する事を考えてDVD-ROMを選択するのも良いかもしれません。

FDDはBIOSのアップデートや後述の効率的なクラスタインストールに利用する事があるので、あった方が良いでしょう。

ビデオカードは、ほとんどの場合、クラスタの性能とは無縁なので、、選択できる一番安いものにしておきます。

数十台規模のクラスタでそれぞれに画面、キーボード、マウスといったコンソール機器を備えると、かなりのスペースが必要となり、現実的ではありません。クラスタが稼動開始してしまえば、ほとんどの操作はリモートで実行出来るので画面、キーボード、マウスといったコンソール機器は取り外しても構いません。モニタ/キーボード/マウスは、可能であれば無しで購入し、後述のコンソール切り替え器を利用するか、まったく接続せずにリモートで利用すると良いでしょう。

その他、クラスタとして不要な機能、テープドライブや余計なサウンドカードは全てはずしてしまうと良いでしょう。

3.4.　マスターノード

計算ノードを統括管理するまとめ役です。バッチ処理の統括などを行います。これが故障するとクラスタ全体が機能しなくなりますので、そのことを意識した工夫が必要となります。

一つの方法は計算ノードとは別に準備し、高速性よりも信頼性を重視して設計する方法です。

各種構成機器を二重化したサーバ機を利用し、故障に備えます。

もう一つの方法は、マスターノードの予備機を用意する方法です。バッチ管理ソフトによっては、マスターノードが故障すると別の予備のマスターノードが自動的に機能を引き継ぐものが存在します。(Sun Grid Engineなど)

3.5. ファイルサーバ/データサーバ

計算に必要なデータを蓄えておく役目です。ここでは、性能もさることながら、信頼性について良く考える必要があります。PCクラスタ構築に限らず、計算機システムを構築する場合にはデータをどう保護するかについて良く考えなければいけません。

PCクラスタの場合は、この部分をNFSファイルサーバとして構築する場合が多いと思いますが、データベースを利用したり、その他のユニークな方法をとっても良いでしょう。

この部分について考える時、データを蓄えておく部分と、データを各ノードに提供する部分に分けて考える事が出来ます。

データを蓄えておく部分は、普通はハードディスクで構成する場合が多いでしょう。その場合、RAID[6]ドライブ等を利用して耐障害性を高めておく必要があります。小規模の場合はRAID1ドライブ、大容量の場合はRAID5ドライブを利用すると良いでしょう。

データを各ノードに提供する部分の部分は、小規模であれば計算ノードあるいはマスターノードを流用しても構いません。ただし、信頼性の確保に留意して下さい。具体的には、信頼性の高いサーバクラスハードウェアを採用し、内部の各種構成機器(システムディスク、電源など)も多重化すると良いでしょう。NICすなわち各計算ノードとのインターフェース部分はアクセスの集中を考慮して(計算ノードのNICで100Mbpsクラスを採用したとしても、)1Gbpsクラスの高速なものを採用するべきです。

ファイルサーバ/データサーバを置かずに、NAS[7]を採用するのも一つの考え方です。その場合もNICが1Gbpsクラスの高速なものを選択するべきでしょう。

なお、高速なNICを利用してもどうしてもファイルサーバ/データサーバにはアクセスが集中してシステム全体のボトルネックとなることが多いようです。この点をNFSでは無くSAN[8]を用いた共有デ

---

[6] **RAID:** (Redundant Arrays of Inexpensive Disks)の略。複数のハードディスクを組み合わせたディスクシステムで性能や信頼性を高めるたけの一連の技術の総称。信頼性を確保する為の技術としては、RAID 1(ミラーリング。2n台でn台分の容量を確保。ミラーリングのペアの片方が壊れてもデータが保護される。)あるいはRAID 5(データにパリティを付加、パリティを各ドライブに分散して構成。n+1台でn台分の容量を確保し、1台壊れてもデータが保護される。)

[7] **NAS:** Network Attached Storageの略。ネットワークを通じてファイル共有サービスなどを提供するストレージ機器。ネットワーク上では通常のファイル・サーバと同様に見えるが、サーバでは無いので設定と管理が容易となる。

[8] **SAN:** Storage Area Networkの略。ストレージやバックアップデバイス、ホストを1Gbps～2Gbpsの光ファイバーで接続しストレージのネットワークを構成する。ストレージの一元化管理、遠隔設置、ネットワーク&ホスト低負荷バックアップ等を実現する。TCOの削減の観点から、近年企業への導入が相次いでいる。

ィスク方式で解決しようとした例が、東京大学生産技術研究所喜連川研究室[9]にあります。

3.6.　ネットワーク

ネットワークは、ある意味計算ノード以上に重要な部分です。特に、大規模でノード間通信が大きな並列問題を大規模クラスタで扱おうとすると、高速かつ低遅延のネットワークが不可欠です。逆に、10CPU程度の小規模なPCクラスタならば、100Base-Tの100Mbpsクラスのネットワークでも何とかなります。

PCクラスタで高速ネットワークを実現する方法としては、これまではMyrinet(http://www.myri.com/ 参照)(最大転送速度1GBps)あるいはMyrinet 2000 (2GBps)が使われる事が多かったようです。Myrinetは転送速度もさることながら、低レイテンシ(通信遅延が小さい)という優れた特徴を持っています。しかし、Myrinetは1ノード当たり20万円以上のコストがかかりますので、PCの価格がこなれている現状では割高感が否めません。

100Base-TのEthernet環境でも、低遅延のドライバを使うことによって、性能向上を図れますが、NICを選ぶ場合がありますので、注意が必要です。

最近では、1000Base-Tのネットワークインターフェースカード及びスイッチングハブの価格がこなれてきていますので、これと低遅延のドライバを組み合わせて利用すると良いかもしれません。

3.7.　ハードウェアの選択 – その他のハードウェア製品 -

PCクラスタを構成する場合に忘れてはいけないものが、上記の基本構成以外の構成品です。数十台規模のPCクラスタとなりますと、その物量はあなどれません。

具体的な数値は設計次第ですが、PC本体価格の数割にはなると思います。予算獲得や発注前にはこの点について良くベンダーと相談することをお勧めします。

3.8.　ラック

台数が少ない場合は机に並べるタイプでも良いのですが、余裕があればある程度しっかりしたラックを用意すると良いでしょう。安く上げようと思う場合は文房具のラックなども利用可能だと思いますが、どのような方法をとるにせよ転倒対策などの安全対策に十分注意を払ってください。

3.9.　コンソール及びコンソール切替器

クラスタが稼動開始してしまえば、ほとんどの操作はリモートで実行出来るので画面、キーボード、マウスといったコンソール機器は取り外しても構いません。しかし、一々何かしようと思った時にキーボードとマウスと画面を切り替える必要があるとインストールや管理作業が大変です。数十台規模のコンソールを切り替える装置がありますので、そういう装置を使うとこれらの問題は解決できます。

---

[9] **東京大学生産技術研究所喜連川研究室のSAN共用ディスク方式PCクラスタ:**
http://www.brocadejapan.com/sansolution/Library/Tokyo-Univ..pdf
http://www.tkl.iis.u-tokyo.ac.jp/~kgoda/pccluster/

3.10. マシンルーム

PCクラスタが数十台規模になれば、それをどこにどのように設置するか良く考えねばなりません。重量、体積、消費電流、発熱量など考えると、ある程度設備の整ったマシンルームが必要と考えてください。

図4:クランプ電流計

まず、その重量と体積です。ラックを使う場合は床の荷重に十分注意が必要です。場合によっては耐震対策なども必要になってくるでしょう。

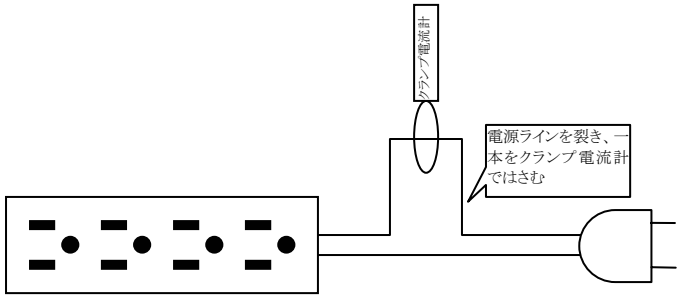

図5:クランプ電流計とOAタップを利用したPC消費電流実測装置

PCクラスタの発熱量を、所詮PCとあなどってはいけません。最近のPCはCPUだけでも最大100W近くの熱を消費します。これが何十台となると、数KW、数十Aとなります。それでは、システム全体で必要な電源容量はどうやって決めれば良いでしょうか？PC自体の持つ電源ユニットは300W超など余裕を持って大きな物が付いています。しかし、これは内部にいろいろ増設した時に必要な容量です。もしこれを台数分倍した容量を確保するととんでもない容量が必要ということなってしまいます。これは、導入予定機で実測するのが一番手っ取り早く、かつ確実です。クランプタイプの電流計(図4参照:安いものですと、秋葉原等で1万円程度で入手出来ます。)とOAタップを改良して用意すると簡単に測定できるようになります(図5参照)。測定は、かならずPCに負荷をかけた状態で測定してください。搭載CPU全て100%の利用率にする事はもちろん、CD-ROM、ハードディスク、ネットワークインターフェースなど各種ドライブにも負荷をかける事を忘れてはいけません。(数値計算を実行しながら、ハードディスクにアクセスし、CD-ROMからデータを読み出しながら、ネットワーク上の他のマシンとの間にファイル転送をする....といった具合になります。)ネットワーク機器であるスイッチングハブも意外に電気を食いますので注意が必要です。

クランプ電流計はその他にもマシンルームの電源容量の余裕の測定にも使いました。また、クラスタ据付時の電源投入時にも利用しました。これを利用してブレーカーが落ちないことを確認しながら1台ずつ電源を投入しました。

3.11. 付属品の保管

数十台規模のPCクラスタを構築すると、その付属品、廃棄物の物量もあなどれません。

返却を前提としたリース購入の場合、あるいは将来のデスクトップPCへの転用などを考えている場合、付属品も無下に捨てるわけには行きません。特にリースの場合は返却時に揃っていなければいけない物品についてリース業者と細かく詰めておいた方が良いでしょう。付属品の中には、ソフトウェアのライセンス証書など、第3者に偶然取得されては問題のあるものも混じっていますので、こういったものの管理、廃棄にも注意が必要です。

何を捨てるか、何を残しておくかが決まったら、どのように捨てるか、保存するかを考えねばなりません。捨てるものに関しては、あらかじめ発注時の仕様に不要物の廃棄も含めておき、据付完了後ベンダーに回収してもらうと良いでしょう。

30台規模のクラスタでこれだけの廃棄物の量となります。これでも、梱包材などを各箱に詰め込んで整理し終わった状態です。
据付当日の立会作業は据付作業中の業者の横でこれらゴミの整理をして過ごす事になりました…

図6：GNF-Jクラスタ導入時のゴミの山

壁面キャビネット一つを占有してしまいました。キーボード/マウスで半分以上を占めています。

図7：GNF-Jクラスタ導入時の付属品

4. ソフトウェアの選択

4.1. OS

PCクラスタのOSとしては、UNIX系のOSが使われる事が多いです。特に最近はLinuxで構築する事が多いようです。これは、各種のオープンウェアがUNIXを中心に開発されてきた事と、Beowulf型と呼ばれるPCクラスタの成功の影響が大きい為だと思います。Linuxと言っても現状幾つかのディストリビューションが存在します。その中からどれを選択するかは、組み合わせて利用するコンパイラやバッチ処理システムなどのミドルウェアのサポートディストリビューションから決めると良いでしょう。

クラスタ構築用の一通りのコンパイラ、ミドルウェアと、クラスタ用の特別なドライバ、効率の良いインストールツールをセットにしたクラスタ専用ディストリビューションとでも言うべき、SCOREを利用する方法もあります。（http://www.pccluster.org/ ）

Windows 系のOSでの構築例はまだ少ないようです。PCクラスタを利用する為に必要な、特に並列計算サポート用の各種ミドルウェアが不足している、あるいは存在していてもあまり価格がこなれていない事も原因の一つですが、最も大きな理由は、Linuxベースのクラスタが着実に実績を重ねている同時期に主流であったWindows NT 4.0が非常に不安定であった事が原因です。（「Appendix A」参照）

どのようなOSを採用したとしても、使用する予定のハードウェアがそのOSで利用可能かどうか良く調査しておく必要があります。Linuxの場合は対応ハードウェアは多いですが、たまに「動作はするがパフォーマンスが低い」場合がありますので注意が必要です。(GNF-Jでもパイロットクラスタ構築時にNICではまりかけました。)運が良ければ、インターネット上から高速なドライバ入手したり、最新版カーネルへのアップデートで性能が改善します。事前に指定出来るならば、実績のある高性能なハードウェアを選択すべきです。

4.2. コンパイラ/デバッガ

フリーで利用できるgccベースのコンパイラでも殆どのプログラムはコンパイル出来ますが、最適化が不十分なのであまり性能が出ない事が多いようです。

Linuxで使えるFortranコンパイラには

- PGI (http://www.pgroup.com)
- Intel (http://developer.intel.com/software/products/compilers/)
- 富士通(http://www.fqs.fujitsu.com/fort-c/)

などがあります。

製品によっては一定期間試用できるものもありますので、実際に移植しようと思うプログラムがうまくコンパイル、実行出来るか試してみると良いでしょう。

PGIや富士通のコンパイラは自動並列化などの機能を持っています。gccやIntelコンパイラの様にコンパイラ自体が自動並列化などの機能を持っていなくてもMPIやPVM等のライブラリをコンパイルしてそれらを利用するようにコーディングすれば、並列処理は可能です。

大抵のコンパイラには原始的なデバッガが付いて来ますので、簡単なデバッグでしたらそれを利用します。並列処理を効率よくデバッグする為のデバッガ製品としては、Totalview(http://www.etnus.com/Products/TotalView/index.html)という製品があります。スーパーコンピュータを使ったことのある人なら、聞き覚えのある名前かもしれません。Linux上では、gcc, PGI, Lahey/Fujitsu Fortran 90/95などのコンパイラをサポートしています。

4.3. 並列計算/分散処理用ミドルウェア

ただPCクラスタを組み上げてOSをインストールしただけでは、効率よく計算には利用出来ません。並列計算、あるいは分散処理を行うためのミドルウェアが必要です。

並列計算では、MPIを使ってプログラミングをおこないます。(昔はPVMを使う場合が多かったのですが、最近では殆どの場合MPIを使います。既存のPVM対応プログラムを利用する目的以外で

は新たに利用する必要は無いかと思います。)Linuxで使えるフリーのMPI実装として、MPICHがあります。最新版は1.2.4であり、2002年の5月にリリースされました。(http://www-unix.mcs.anl.gov/mpi/mpich/) RPMでパッケージングされたものを使うか、ソースからコンパイルする事になります。

並列処理あるいは分散処理をクラスタに行わせる為には、バッチ処理システムを導入しないと、ジョブの管理が大変面倒になってしまいます。

商用のバッチ処理システムとしては、LSFが有名ですが比較的高価です。PCクラスタ上では、フリーのバッチ処理システムを利用する事が多いようです。バッチ処理システムについては、あまり詳しくは知らないので、知っている範囲の情報に留めます。

- OpenPBS (http://www.openpbs.com/)　良く言えばシンプル。悪く言えば低機能。シングルプロセッサ上で動作するジョブだけではなく、MPIベースの並列ジョブが扱える。障害時の予備のマスターノードへの自動切換(フェールオーバー機能)、ジョブの自動再実行(リカバリ機能)など複雑な事は苦手。機能を強化した商用版であるPBS Proもある。

- Grid Engine (http://gridengine.sunsource.net/)あるいは
  Sun Grid Engine (http://jp.sun.com/products/software/serverperf/gridware/index.html)
  Sunが開発している、グリッドコンピューティング用の処理システム。小規模向けのSun ONE Grid Engine は無償。 Grid EngineはSGEのオープンソース版。(フリーソフトとして、ソース公開の上開発されており、多数のプラットフォームに対応している。)MPI並列ジョブが扱える。マスターノード・フェールオーバー機能、ジョブの自動リカバリ機能などを備える。

ちなみに、GNF-JではOpenPBSしか経験がありません。将来的にSun Grid Engineを利用してグリッドコンピューティングに発展して行ければと思っています。

4.4.　故障対策

クラスタに限らず、計算機システムを設計する時には必ず故障すると考えて対策をしておく必要があります。同時に、想定される各種故障について、その発生頻度、どれくらいシステムがダウンしていても良いのか、どの程度過去の状態まで戻ってしまっても許容されるのかを、を良く考える必要があります。

例えば、銀行のオンライン業務のようにミッションクリティカルと言われるシステムは、少しでも停止しようものなら、会社の経営に与えるインパクトは多大です。(みずほ銀行のトラブルが記憶に新しいところです。)このようなシステムは非常に高信頼性のマシンと高性能なスタンバイ機への自動切換え機能を利用して構築しますので、システム価格も維持管理コストも高価になります。

エンジニアリング用途のPCクラスタであれば、それほどシビアな条件は必要ないでしょうが、そ

れでも一週間も使えなかったり、大切なデータが失われたりすれば損失は莫大です。研究用途のクラスタであれば、それでも平気な場合もあるでしょう。

数時間の停止が許容されるなら、十分な備えがあれば人手でシステムの復旧が可能でしょうし、数分程度しか許容されないならば、高度な自動復旧の仕組みが必要になります。

自分が構築しようとしているシステムに要求される条件を良く見極め、限られた予算を「性能」と「信頼性」にどのように配分するか決めなければいけません。

PCクラスタで故障について考えるとき、次のような対策が有効となります。

- ハードウェアの信頼性を高める
- ソフトウェアの信頼性を高める
- 故障を早期に発見、修正できる体制を整える
- ある程度の故障を許容する

1点目はお金をかけて必要なレベルを確保するのが基本です。2点目はあまり冒険をせずに堅実なソフトを選択することです。3点目は、システム監視ソフトの類を利用すると良いでしょう。

4点目は、ハードウェアの側面から見れば、RAIDで多重化されたドライブ、二重化された電源やファン等が該当します。ディスクドライブを含めたファイルサーバの故障に備えてバックアップを日常的に取っておくことも大切です。バックアップから復旧する手順、所要時間が予定の範囲に納まるかどうかの見極めも必要です。障害発生後どこまで過去の状態まで戻る事が許容されるかで、バックアップの頻度が決まってきます。その他の機器の故障、例えばスイッチングハブの故障などにどうするか事前に考えておきます。

ソフトウェアの側面から見た場合、Sun Grid Engineであれば、フェールオーバー機能でマスターサーバーの故障に備える事が出来ますし、計算ノードの故障に対してはジョブリカバリ機能で対応できます。

4.5. インストールを簡易化するツール

PCクラスタの台数が多くなると、OSのインストール自体が結構大変な作業となります。OSのインストールや細かい設定などは、経験のある方ならご存知だと思いますが、効率よくやらないとたった一台でも最低半日、長いと数日かかってしまいます。それが何十台もあるわけですから大変です。

GNF-Jでは、IBMのLUI（Linux Utility for Cluster Installation：http://oss.software.ibm.com/developerworks/projects/lui/ ）というツールを使って2,3日以内にインストールが完了しました。（LUIは、現在既に新規の開発は行われないメンテナンスモードに入っています。[10]）RedHat Linuxで

[10] **LUI:** IBMのLinux HPCプロジェクトそのものがOpen Cluster Group（http://www.openclustergroup.org/）の活動に移行しており、LUIの技術はその成果物であるOSCARパッケージ(RedHat Linux/Mandrake用のクラスタ構築パッケージ)中のSIS（http://sisuite.org/））コンポーネントに生かされています。

は、KickStartという方法を使って効率よく多数のマシンへのインストールが可能です。クラスタ専用ディストリビューションであるSCOREでは、EIT(http://www.pccluster.org/score/dist/score/html/ja/installation/eit.html)というツールを利用してマスターサーバ以外の計算サーバのインストールを簡略化する事が出来ます。

5.　GNF-Jで構築したクラスタ

GNF-Jでは、これまでにパイロットクラスタと実用本格クラスタを構築してきました。

5.1.　パイロットクラスタ

パイロットクラスタは、実用クラスタを導入するに先立って構築したものです。この上で様々なノウハウを蓄積する事が出来、実用クラスタの設計へとつなげて行けました。

パイロットは、Pentium III 886MHz Dual / Memory RDRAM 1GB / HDD Ultra 160/m SCSI 18GBといった構成のPCを6台で構成しました。キーボード/マウス付き、モニタレスとし、モニタケーブルを毎回手でつなぎかえて利用しました。

OSにはRedHat Linux 6.2Jを採用。コンパイラとしてPGI Compiler，デバッガとしてTotalview 、バッチ処理システムとしてOpenPBSを採用しました。これは、コンサルテーションを受けた会社の当時の標準メニューから選択したものです。

5.2.　実用クラスタ

GNF-Jでは、当時最高速であった1.7GHzのPentium4のワークステーションタイプPCを計算ノードとして採用しました。当時、Xeonはまだ出ていなかったので、マルチプロセッサはかなり早い時期に候補から外れていました。RDRAMタイプのメモリを512MB搭載しました。特に各計算ノードに大きな一時ファイルは作成せずに、すぐにファイルサーバの方に結果を書き戻すような計算が多いので、HDDは最低限の容量しか搭載していません。 インターフェースは、当時入手可能であった最高速度のSCSI HDDであったUltra 160/m SCSI を採用しました。

ノード間は100base-Tのスイッチングハブで接続しました。

データサーバとして、SunのEWSベースのNFSファイルサーバを設置しました。Compaq SANとSunのEWSを組み合わせたシステムを利用しています。Linuxをファイルサーバとしなかったのは、ロギング機能のよる障害時の高速復帰、ユーザの容量管理などを備えたファイルサーバは当時のLinuxでは構築が難しかった為です。データの保護は弊社のエンジニアリング業務の根幹に関わる重大な課題です。DiskのRAID化、UPSの採用、日常的なバックアップの実施と、それなりのコストをかけて性能と信頼性の確保に心がけました。全ノードからのアクセスが集中するので、ネットワークインターフェースは1Gbpsと高速なものを採用しています。

30台のPCのコンソールを切り替えるために、コンソール切替器を利用しました。キーボード/マウ

スが不要だったのですが、PC購入時に発注内容から外す事ができなかったので、大量に余ってしまいました。しかし、これも、キーボードの故障時用のスペアパーツとして役立っています。

マスターノードは、計算ノードと全く同一のものを利用しています。バッチ管理ソフトとして採用したOpenPBSには予備のマスターノードを構築する機能は持っていないので、頻繁にマスターノードのバックアップを作成し、マスターノードの故障時にはその内容を復旧するか、別の計算ノードのその内容をコピーして新たなマスターノードとして利用する事としました。実際には、稼動期間一年以上になりますが、マスターノードを含め故障したことは一台もありません。

OSにはRedHat Linux 7.2を利用しました。コンパイラにはPGIのコンパイラ、並列デバッガとしてTotalview，バッチ管理ソフトとしては、OpenPBS を用いました。

5.3. GNF-Jクラスタの性能

このようにして構築したPCクラスタでの性能測定値を紹介します。

まず、理論性能値は1.7G×30=51Gflopsという事になります。

世界のスーパーコンピュータのランキングを競うTop500[11]でも利用されている、HPLベンチマークでは24GFlopsを記録しました。これは、現在のトップ500位(136Gflops)に及ばない成績ですが、1994年のランキングではTop10、1997年でTop100、1999年でTop500という事になります。(最初に書いたとおり、)5年程前なら世界のトップクラスであったスーパーコンピュータの性能が非常に安価に入手出来るという事が分かると思います。

姫野ベンチマーク[12]では、8CPUで1830MFlops、30CPUで2560MFlopsを記録しました。

Bonnieというディスクアクセス性能を測定するベンチマークでNFSによるネットワークファイルアクセス速度を測定すると、読み出しでは約10MB/secと100Mbpsというネットワーク速度に見合った性能が出ましたが、書き込みの方では約5MB/sec程度の性能しか出ませんでした。これは、Solaris NFSサーバの実装とLinux NFSクライアントの間であまり性能が出ないという現象の様です。

正直なところ、100Mbpsのネットワークなのでやはりこの程度かという気持ちと、ネットワーク周りを簡単に構築した割にはかなり性能が出るものだという気持ちが半々です。

使ってみて分かった事は、並列処理に比べて分散処理が非常に効果があるという事です。アムダールの法則などがありますので、並列プログラムはCPUを沢山並べて効果を得るためにはかなりの努力が必要です。一方、パラメトリックサーベイなど多数のジョブを発生する解析にバッチ処理システムを利用した分散処理を採用すると台数が線形に効いて非常に効率よく計算時間を短縮出来ます。見方によっては、分散処理は非常に効率の良い並列処理といえます。一般に、プログラムを

---

[11] **Top500**：(http://www.top500.org/)世界のスーパーコンピュータを(行列のLU分解を行う)LINPACKベンチマークの性能でランキングしたサイト。現在、日本の地球シミュレータが一位である。HPL(http://www.netlib.org/benchmark/hpl/)を利用すると簡単に測定出来る。

[12] **姫野ベンチマーク**：理化学研究所の姫野龍太郎氏が開発した非圧縮性流体のソルバーによる、ベンチマークテストである。並列計算機のベンチマークの一つとして広く利用されている。(http://w3cic.riken.go.jp/HPC/HimenoBMT/index.html)

局所的に並列化するよりも大局的に実施した方が並列化の効果が出やすいので、さらにこの大局化を進めたものと見るわけです。

この様な書き方をすると、これから並列プログラミングをがんばるぞという気持ちで本特集を読まれている方に水を指すような話になるかもしれませんが、無理に並列化する事を考える前に、解析業務そのものを大きく捕らえ並列性を見出し、分散処理で解析業務を並列化する事が効率的です。そして、並列化の努力は本当に並列化すべき大きなプログラムに集中投資して高い並列化効率を達成します。そうすると、分散処理と並列処理でクラスタのCPUを効率よく使い続ける事になり、投資効果を最大限に高める事が出来ます。

5.4. まとめ

最近「ナノテク、バイオ、地球環境シミュレーションの各分野では、実験や理論研究といったものと同列に計算機科学が位置し、テラフロップス級(ギガフロップスの1000倍の性能)の計算機を駆使することでこれら最先端の技術開発を推進している」というようなことをよく耳にします。そして、計算機ハードの進歩ではなく、逆に利用する側、あるいはアプリケーション開発側のニーズがこの種の研究を牽引しているとも言われています。地球シミュレータ[13]もその一例と言えるでしょう。そして、次のペタフロップス級(テラフロップス級のさらに1000倍の性能)の計算機ニーズさえもこれら研究者の間からは聞こえて来ます。こういった目標に向かって世界中に分散した計算機資源をフルに活用しようという、GRIDコンピューティングの概念研究が進められていることもご存知の方は多いと思います。

我々、基盤エネルギーを支える原子力に携わるものとして、もし計算機パワー不足で何か技術的に停滞している部分が少しでもあるとするならば、誰もが気軽に使えるHPCを積極的に活用することを考えてもいい時だと思います。そうすれば新たなステップが切り開けるかもしれません。地球シミュレータならぬ究極の原子炉シミュレータができたら、それに見合った究極の炉設計ができるかもしれません。個人的には、各サイトのクラスターを相互乗り入れし、GRIDコンピューティングを実現できたらなどと思いを巡らしています。あるいは、2001年に開始された、ITBLプロジェクト[14]が究極にはそのクラスの計算が出来る規模になれば面白いと感じています。

---

[13] **地球シミュレータ:** 海洋技術センターがNECと共同開発。現在世界最速スパコンとして公認されている。LINKPACKベンチマークで約36Tflopsを叩き出した。2002年3月完成。ベクトル・並列計算機のハイブリッド計算機である。約400億円。

[14] **ITBL:** (Information Technology Based Laboratoryの略) IT技術を活用して仮想的な共同研究環境を実現するプロジェクト。国内の100台以上のスーパーコンピュータを接続して共有、共同利用することを目指してい。http://www.itbl.riken.go.jp/

Appendix　A：PCの故障率とクラスタの故障率の関係

平均的に時間間隔Tに一度故障するような機器はMTBF(平均故障間隔時間)がTであると言います。

例：半年に一度故障する場合　T=365/2 (day)

この機器の故障率λはMTBFの逆数として定義されます。(例ですと、$\lambda = 5.48 \times 10^{-3}$となります。)

この機器が一定期間無故障で動作する確率を信頼度といい、故障の発生が偶発的で故障の発生件数がポアソン分布に従う場合$R = e^{-\lambda t}$と計算されます。(例ですと、一年間の信頼度は13.5%となります。)

さて、かつてWindows NT 4.0は一週間程度も連続運転出来ないといわれていました。もし、MTBFが一週間のOSで30個の計算ノードからなるPCクラスタを構成するとどうなるでしょうか？一台も故障を許さないとすると、そのたった一日の連続運転の信頼度はたった1.4%となります。これが、PCクラスタが普及した頃にWindowsベースのクラスタが広まらなかった大きな原因の一つと考えられます。表 2にクラスタ30台運用時の機器MTBFと信頼度の関係を示します。MTBFが一年以内のOSでは一週間の安定稼動もままならないということが分かります。

表 2:クラスタ30台運用時の機器MTBFと信頼度の関係

MTBF(day)	一週	一ヶ月	半年	1年	2年	10年
1day 信頼度	0.014	0.380	0.848	0.921	0.960	0.992
1week 信頼度	0.000	0.001	0.316	0.563	0.750	0.944
1month 信頼度	0.000	0.000	0.006	0.078	0.280	0.775
1year 信頼度	0.000	0.000	0.000	0.000	0.000	0.050

Appendix　B:CPUの価格性能比について

一般に、「PCは価格性能比が他のEWSに比べて良い」といわれますが、それでは一体どれほど安いのでしょうか？人によっては10倍違うという人も言えば、その他の性能を考えれば大して差は無いという人まで様々です。実は決まった結論というものはありません。実際にプログラムを走らせるまで分からないというのが実情です。というのも、計算機システム上で行う計算は千差万別なので、結論も千差万別、整数演算と浮動小数点演算の割合、コンパイラの最適化の効き具合もプログラム、計算の内容によって随分と異なってきます。

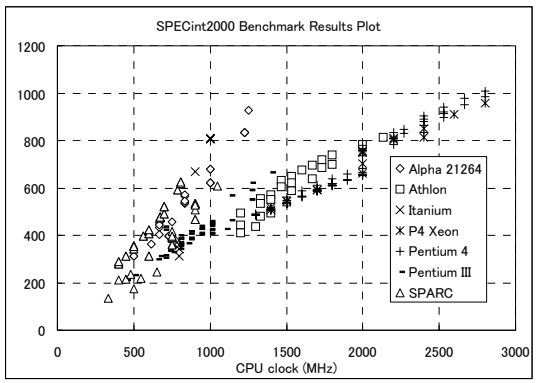

図8:SPECint2000のCPU毎の性能比較

しかし、可能性のある機器を全て試せるわけではありませんので、何らかの目安が必要です。こういう時に便利なのが、SPEC社が公開しているSPEC CPU2000ベンチマークの結果[15]と各社のインターネットショッピングサイトです。

単純にSPEC CPU2000のベンチマークの結果をダウンロードし、クロックと性能関係をCPUの種類毎にプロットすると、PCクラスタに関する情報以外にも様々な情報が得られます。例えば、整数演算性能(SPECint2000)の結果(図8参照)より、

- おおむね、同一種類のCPUの性能はクロックに比例する。(実際はだんだん寝てきます)
- EWS用のCPUはPC用のCPUの約1.5倍
- しかし、CPUの最高クロック値が倍程度あるので、最高値はPentium4(2.8GHz)。
- 2GHz以下ではPentium4は同一クロックのPentium3, Athlonに負ける傾向があるが、2GHz超において、クロックあたりの性能が向上して遜色無くなっている。さらに詳しく結果を見ると、WillametteコアからNorthwoodコアに切り替わっている事が分かる。2次キャッシュの容量が256KBから512KBに大きくなった点が効いているようであり、同様の傾向がPentium III,他のCPUにおいても見られる。

一方、浮動小数点演算性能においては、(図9参照)

- おおむね、同一種類のCPUの性能はクロックに比例する

---

[15] **SPEC :** http://www.spec.org/ SPEC社はコンピュータシステムに関する性能を測定する各種ベンチマークテストプログラムをとりまとめており、その結果がWebを通じて公開されている。ここでは、CPU/Memory/コンパイラの総合性能を測るSPEC CPU2000の結果に着目している。

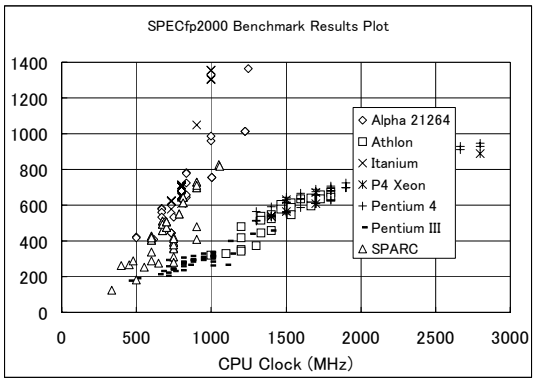

図9:SPECfp2000のCPU毎の性能比較

- クロック当たりの浮動小数点演算性能(SPECfp2000)で見た場合、EWS用のCPUはPC用のCPUの約2～3.5倍(整数演算性能に比べて開きが大きくなる。)
- 現在主流の1GHz程度のEWS用CPUだとその性能はPentium4の2.8GHzと同程度
- SPECint2000のPentium4で見られたようなキャッシュ容量による不連続性が見られない。
- 最高性能はES45という機種に搭載されたAlphaチップである。
- Itanium2のクロックあたりの性能は非常に高く、1GHzで1.25GHzのAlphaとほぼ同等の性能を出している。

などという事が分かります。

このように、このグラフを眺めていると、世の中で売っているPC/EWSの性能が概ね分かる様になってきます。

この結果をふまえて、次に価格の比較を行っていきます。このとき、大切な事は値段を調べるマシンのスペックを実際に実行予定のプログラムのスペックに揃えるという事です。具体的にはメモリ、ディスクなどのスペックを全て揃えます。

また、マルチプロセッサの機種は、とりあえずは単純にシングルプロセッサの性能値を搭載CPU数倍にすることにします。(SPEC fp/int 2000 rateの結果がもっと多くのCPUで得られればそちらを参考にする事も考えられますが、現在ではあまりシングルプロセッサのSPEC fp/int 2000 rateの結果が得られないので、このような方法を取っています。)

そして、各社のWebショッピングサイトで目標スペックのPCあるいはEWSの価格を調べると、性能あたりの価格が分かります。今回の原稿を書くにあたり、以下の条件で調べてみた結果を例示します。

スペック

- メモリCPUあたり1GB
- ハードディスク 18GB以上
- ネットワークインターフェース 1000Base-Tを増設
- その他のスペックは最低限レベルとする。

調査結果

- WorkStationクラスのPCでSPECfp2000のスコア当たりの価格は300円程度。
- サーバクラスのPCでは価格の幅がかなりある。Dualで350円程度と1～2割増
- Xeon MPを利用した4CPUクラスのサーバは、1400円にもなる。
  最高クロックもPentium 4に比べて2/3程度。IA-32アーキティクチャのマルチプロセッサ性能はCPU負荷が高くなると落ちる傾向がある(Appendix　C参照)ので、計算用サーバとしてのコストメリットは低いと思われる。
- EWSを見ると、1600円から2500円程度となりました。平均値で約2000円というところでしょうか。

これらの結果の比を見ますと、PCとEWSの価格性能比は約6倍(SPECfp2000ベース)という事になります。また、同様の比較をSPECint2000について行うと、さらに差は広がり、約8倍となります。

同様の比較を独自に2年前に行ったときは、SPECfp95ベースで約3.5倍、SPECint95ベースで約5倍でしたので、その差は広がりつつあると感じました。(ただし、スペックの条件やベンチマークが異なりますので、単純には比較は出来ません。)

このPCとEWSの価格性能比を持って、単純にEWSは割高であると思わないで下さい。EWS用のハードウェアは一般にPC用のハードに比べて信頼性が高く、プロセッサ周りのバスの性能なども良くなっています。PCは単純にクロックをぶん回す事によってその不利な点を補っているわけです。ただ、それだけのハンディを補っても、やはり汎用品の割合が少なく、出荷台数の違いでスケールメリットがなかなかでないEWSは割高感を否めません。

このような状況の中で、IntelのItaniumやAMDのHammer、SUNのSPARCチップ、IBMのPower4等の64bitCPUが今後どのように展開していくか非常に興味のある点でもあります。これらのチップが価格性能比で既存のIA-32アーキティクチャの32bitCPUに追いついてきた時には、それらを利用してクラスタシステムを構築すると大きな性能の向上が期待できるからです。

Appendix　C：マルチプロセッサマシンでの計算性能について

「3.3計算ノードの選択」でも触れましたが、SMPタイプのマルチプロセッサのマシンを数値計算に利用しようとする場合、その性能がCPUの数に比例して向上しないので注意が必要です。これは、並列処理ではない、複数のシングルタスクを同時に実行する場合でも発生します。

もし、マルチプロセッサのマシンが手元にありましたら、なるべくメモリに負荷の高いプログラムを選んで、1つだけ実行した時と、複数同時に実行した場合を比較して見てください。(もし、無ければ姫野ベンチマークを入手、コンパイルして実行してみても良いでしょう。)デュアルプロセッサマシンでの計算性能はマシンによって、1.3～2近くまでいろいろと幅が出てきます。この現象は、メモリのバンド幅がCPUの処理速度に対して低すぎる為に発生します。CPUの内部クロックとメモリへのアクセス速度の比は広がる一方ですので、このような事が発生します。

では、このマルチプロセッサでの性能劣化はシステムによって異なってきます。そして、この傾向はSPEC CPU2000ベンチマークの結果、SPECint2000_rate及びSPECfp2000_rateを見ると分かります。rateの付いたベンチマークは63のタスクを生成しますので、およそ64CPUまでのマルチタスク性能を見ることが出来ます。(図10、図11参照)

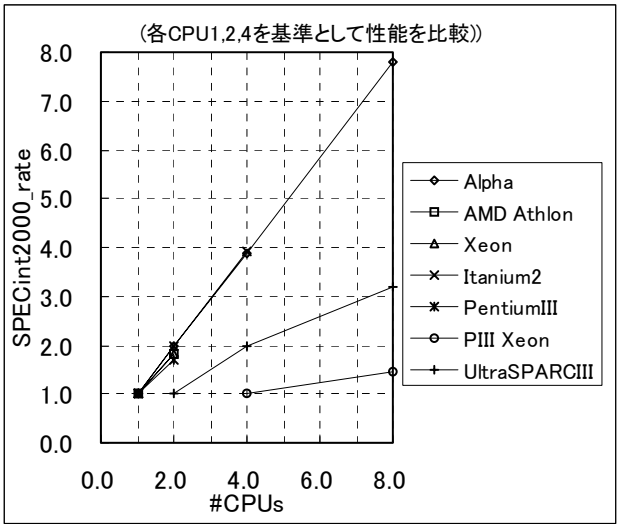

図10：CPUタイプ毎のマルチプロセッサ性能比較(SPECint2000_rate)

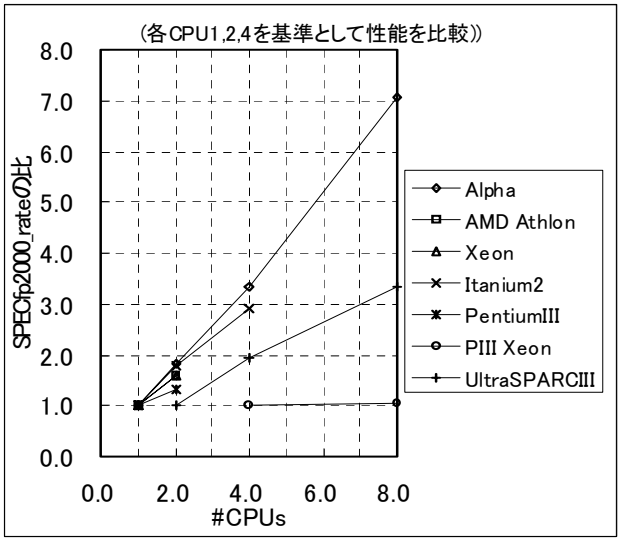

図11：CPUタイプ毎のマルチプロセッサ性能比較(SPECfp2000_rate)

IA-32アーキティクチャのCPU でシン1CPUと2CPUで実行した時のSPECfp200_rate性能の伸びは、Xeon(Pentium 4と同じコア)やAthlonで1.6程度、Pentium3では1.3と全体にふるいません。一方、EWS系のCPUを見ると、Alpha CPUは1.8、SPARC CPUは1CPUの結果が無いのですが、2CPU→4CPUで1.9と伸びていますので、1CPU→2CPUの性能の伸びはかなり2に近いと思われます。このようにEWS用のCPUではマルチプロセッサ環境での性能の劣化が比較的少なくなっています。こ

れは、数Mbytesクラスの大容量キャッシュやクラスバースイッチ等の高速な(そして高価な)周辺バス性能が効いているためです。また、Pentium III Xeonの4CPU→8CPUの性能向上はわずかに3%にとどまります。つまり、メモリに負荷の高い計算のためにPentium IIIを沢山積んだマシンを利用するのはほとんど意味が無いと言うことになります。

一方、SPECint2000_rateのベンチマークではマルチプロセッサでの性能劣化は若干軽減されます。Pentium 4 / Athlonで1.8、Pentium IIIで1.7となります。一方、EWS系のCPUでは2.0とほとんど劣化が見られなくなります。

これらの結果から、走らせるプログラムによってマルチプロセッサ環境での性能劣化度合いが異なるという事がわかります。PCクラスタにおける計算ノードの選定時には実際にプログラムを各種環境で実行してその性能を実測する事を強くお勧めします。