計算機システムについて

○広崎結里、福田優子、谷口麻梨香、斉藤昌樹、長友英夫、西原功修

大阪大学　レーザーエネルギー学研究センター

概要

大阪大学レーザーエネルギー学研究センター（以下、「本センター」と略す）では、様々なレーザー装置を用いたレーザープラズマに関する実験、および理論研究を行っている。研究に必要な計算ニーズはレーザープラズマ実験解析処理と多次元シミュレーションに大別できる。本システムは、前者のレーザープラズマ実験解析処理に対応するものである。レーザープラズマ実験解析処理の概要は、a)レーザープラズマ実験の実験管理、b)ターゲット製作、レーザー装置、実験計測装置、計算機シミュレーションなどで発生する各種データの転送、蓄積、検索、表示などのデータベース構築、c)各種実験計測データ解析処理、シミュレーションデータ解析処理、および処理データの可視化（グラフィック）処理、d)レーザープラズマ実験に必要なパラメータ設定を行う１次元シミュレーション、e)各種多次元シミュレーションプログラム、超高密度プラズマ状態方程式関連プログラム、各種計測処理プログラムなどの開発、f)レーザー装置、各種計測装置の開発設計計算である。平成17年3月、本センターでは、日立：SR8000コンパクトモデルを中心としたシステムからNEC：SX-8モデル6Aを中心としたシステムに更新を行う。本センターでは20年以上にわたり、比較的小人数による大規模シミュレーション研究を主体とする計算機システムの運用を行い、様々なノウハウを蓄積してきた。本論文では、主にシミュレーションを支援するホストコンピュータ関連のシステムについて説明する。

1　システム構成

1.1　全体構成

システムの基本的な構成は旧システムと同じ考え方で、機能階層別構成となっている。（参考文献[1]）。図１は新システムの構成を示しており、ホストコンピュータ SX-8 モデル 6A（96GFLOPS,64GB）、ファイルサーバー、プログラミングサーバー、グラフィックサーバー、端末、実験データサーバー、レーザーデータサーバー、ディリースケジュールサーバーなどから構成される。ホストコンピュータ、ファイルサーバー、プログラミングサーバーはホストコンピュータのシミュレーションを行うための環境を提供している。

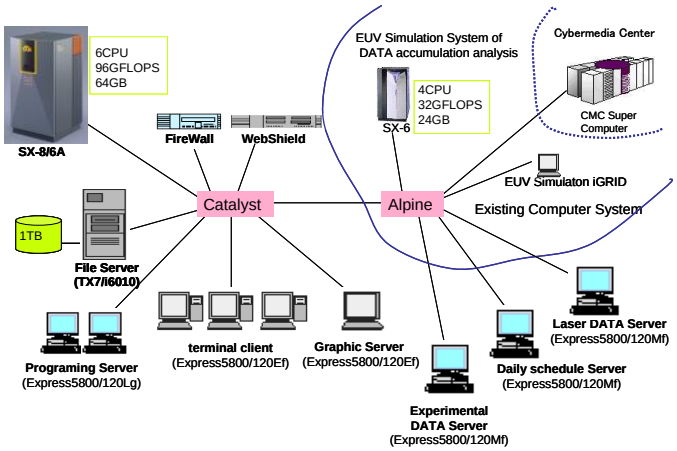

図１　システム概要図

1.2　ホストコンピュータとプログラミングサーバー

ホストコンピュータはシミュレーションの高速実行に専念し、プログラミングサーバーは利用者が直接ログインして、プログラム開発、デバッグ、コンパイル、ジョブ投入、結果確認などの作業を行うという役割を担っている。そのため、ホストコンピュータにはユーザーの直接ログインは許さずバッチによるジョブ実行のみを行い、プログラム開発やコンパイルなどの作業はすべてプログラミングサーバーで行う完全なクロス環境システムを採用し、システム全体の効率を重視している。

本センターのプログラムは、ほとんど研究者による自作の FORTRAN プログラムであり、多くのプログラムはベクトル化による高速化のための最適化がなされている。高速化については、ベクトル化と並列化を基本としているが、ベクトル化効率をあげるというプログラミングは一般に研究者にとって非常に容易であり、比較的簡単に高速化を達成することができる。シンプルなプログラミングで高速化を達成できることは、実際にプログラムを開発しながら研究を行う研究者にとって非常に重要なことである。

1.3　NFS によるファイル共有と領域制限

本センターの NFS によるディスク共有の概念図を図２に示す。ファイルサーバー上に、ユーザーのホーム領域やシミュレーションの生データを保存する領域を用意し一箇所に統一することで、ユーザーはどのマシンにログインしても、同じファイルが見えるため自分のファイルの管理が容易となっている。本センターでは 20 年以上前より、永久にファイルを保存しシステムとしてセーブもとるが容量制限の厳しい「ホーム領域」、シミュレーションからの膨大なデータを保存するための「一時保存領域」の 2 つの領域を用意してきた。ホーム領域にはプログラムソース、シミュレーション実行に必要な入力ファイルやバッチ投入のための NQS ファイルなど、シミュレーション再実行に必要な容量の小さいファイルを保存し、シミュレーションから出力される大容量の生データは「一時保存領域」に保存する。シミュレーションから出力される生データは膨大であり、領域制限を行っているディスクにデータを出力するとその制限のためにシミュレーションがアボートする可能性があるので、「一時保存領域」にはできるだけ領域制限はかけたくないが、不注意によりディスクをオーバーフローさせると他の利用者にも迷惑をかけることになるため、一人あたりの制限としては、領域の半分程度の大きな制限値を設定するようにしている。この「一時保存領域」にあるデータは、最終アクセス日より 3 週間程度でファイルを消すという運用を行っているが、論文が完成するまで、あるいは解析が終了するまで、もう少し長期にわたってデータを保存したいという強い要望があり、新システムではセーブはとらないが、永久に保存し、領域制限値もホーム領域よりは大きく、一時保存領域よりは制限の厳しい「永久領域」を用意する。本センター固有の領域は、各マシンに固有の UINX の一般的なディレクトリ（/etc, /var, /home など）と区別するために、/ile というディレクトリで用意してきたが、どのマシンに属するディスクであるかを簡単に判別できるようにするために、図２で示すように、/ホスト名で始まる領域を用意することとした。本センターのシミュレーションは 10 日以上もかかるような長時間ジョブも多いため、ホストコンピュータで実行中のジョブは、保守のために停止が必要な場合などにチェックポイントリスタートにより中断し、システム再開時には中断点よ

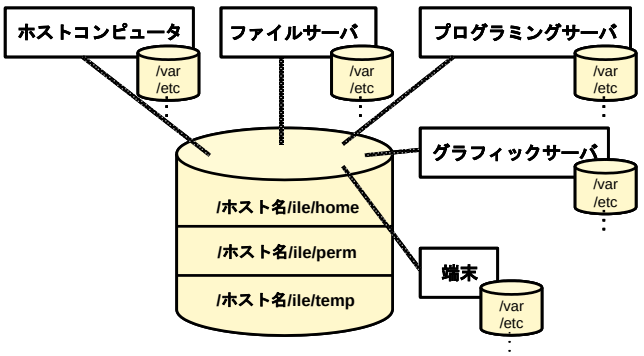

図２　ディスク共有の概念図

り継続して実行するという運用を行っている。このようなジョブの中断は、システム停止時だけでなく、システムを占有するような大規模なジョブの実行の際にも必要であるが、NFS でマウントした領域にネットワーク越しにシミュレーションからアクセスしている場合には、チェックポイント採取に失敗することがある。本センターのプログラムはほとんどが CPU バウンドであり入出力は少ないものが多いが、入出力が多い場合は NFS のディスクをアクセスするよりローカルのディスクの方が好ましい。以上のような観点から、何日にもわたる長時間ジョブや入出力の多い利用者には、計算を実行しているホストコンピュータのローカルのディスクを利用するように指導している。そのため利用者が容易に、どのホストのディスクを使用しているかを判断できるように、ホスト名で始めるというルールを採用することとした。

1.4　異なる OS により構成されたシステム

基本的な OS は UNIX であるが、現在のシステムでは、ファイルサーバーとプログラミングサーバーは HP 社製の HP-UX、グラフィックサーバーは SGI 社製の IRIX、端末は Redhat リナックスと異なる OS で運用していた。様々なソフトウェアをインストールできるなど利用者には評判がよかったが、運用面では様々な苦労があった。OS の違いによるコマンドパス、マニュアルのパスの違いやホスト固有の設定（インストールしたソフトウェアの環境設定など）は、利用者むけに標準で用意している設定ファイルで吸収してきた。しかし､Tex の環境が Redhat より VINE の方がいいという利用者の要望により一台の端末のみ OS を変更したり、新規機能のテストなどのために一部の端末のみバージョンアップを行うなどした場合に問題が発生した。リモートからログインする場合は問題ないが、コンソールからログインすると、それぞれの OS がデスクトップ環境のためのファイル（SGI : .desktop-host やリナックスでは.gnome など）を各個人のホームに勝手に作成する。リナックスとは言っても、異なるバージョンの Redhat をインストールした端末にコンソールからログインしてしまうと、一部同じ名前のファイルが上書きされ、古いバージョンの端末ではコンソールからログインできなくなるというトラブルが何回かおこり、異なる OS のマシンを同じ環境で利用するには充分な注意が必要であることがわかった。新システムでは Redhat リナックスに統一されているため、一部のバージョンアップなどは行わないようにすることで当面は回避する予定である。別にリナックスを用いた端末環境を構築するときには、ホーム領域をそれぞれのローカル（/home/user）に作成するということで、この問題を回避している。

2　運用の工夫

2.1　運用の自動化

UNIX をベースとした複数のサーバーや端末により構成されるシステムであり、極力運用の自動化が行えるように工夫をこらしてきた（参考文献[2][4]）。ツール類は、UNIX の標準のシェルや cron を組み合わせて作成しており、新システムでも、その仕組みを引き継ぐ。定型処理として自動的に採取している情報としては、1)稼動情報、２) ディスク使用量、３) ログイン情報、４) CPU/SWAP 情報、５) NQS 利用情報などである。それぞれ日次的、あるいは月次的に情報を採取し、利用状況の確認やトラブル時の確認に役立てている。それぞれの情報採取用ツール、および採取したログは、ファイルサーバー上で一元管理している。ファイルサーバーからリモートシェルを使用してツール実行を行うことで、マシン毎に設定を行う手間を省くとともに、全てのマシンにログインしてチェックしなくても、ファイルサーバーにログインするだけで自動監視している全てのマシンをチェックすることができ、トラブルの早期発見に役立っている。

2.2　ジョブ運用

本研究センターのシミュレーションプログラムは、小さなメモリしか必要としないものからシステムを占

有するような大きなメモリを必要とするもの、数分で終了するものから１０日間以上もかかるような長時間ジョブまで様々な種類のジョブがある。これらのジョブを効率的に運用するための工夫をこらしたジョブ管理形態（参考文献[3]）を引き継ぎ、各ジョブの特性（CPU 時間、メモリサイズなど）に合わせたキュー設定とスケジューリング、ジョブコントロールなどを状況に応じて行っている。本センター固有の管理ツールを作成するのではなく、OS に付随する標準の NQS の機能をカスタマイズすることにより機能を実現する。設定を変更するだけで、システムの状況に応じて様々な場合に対応することが容易であり、コストが安い、バージョンアップに対応できるなどのメリットがある。最も運用の難しい大容量メモリ、長時間ジョブは大阪大学サイバーメディアセンターのスーパーコンピュータを利用することで効率のよいシミュレーションが可能となっている。利用者は、端末からのコマンドで最新のキューの時間制限、メモリ制限、多重度などの設定情報を参照することができるようにしている。

実行ジョブのログについては、NQS 利用情報を月ごとに集計し、利用状況を確認するだけでなく、システム全体のジョブの状況を 5 分間隔で採取するようにしている。通常システムが用意しているログからこのような情報を得るのは手間がかかる場合が多いが、夜間にトラブルが発生した場合でも、発生時間の絞込みや、その時点で実行されていたジョブについての情報を簡単に得ることができる。

3　終わりに

以上、本研究センターのシミュレーションを中心としたシステムについて説明した。異なるベンダーのマシンへの移行経験は何度かあるが、最も大事なことはコミュニケーションであると感じている。メーカーによって、同じ用語でも異なる意味であったり、同じことを異なる用語で呼ぶなど、気をつけないとお互いに思うことが伝わらないということはよくあることである。システム導入にあたっては、導入するシステムだけではなく、既存のネットワークやシステムとの関係を含めたシステム構成図、利用者向けシステム構成図や使い方などのテキストを早期に作成し、導入業者に開示するなど、どのようにシステムを利用したいと考えているかなどの意思疎通を図ることが重要であると思う。

参考文献

[1]　大橋裕子、福田優子、斉藤昌樹、広瀬華子、長友英夫、西原功修”SR8000 システム導入について” 平成 12 年度 東北大学技術研究会報告集、186 頁～188 頁

[2]　田村篤和、岡本匡代、福田優子、斉藤昌樹 ”分散処理システムにおける運用の自動化の試み” 1998 年度 高エネルギー加速研究機構技術研究会報告集、279 頁～282 頁

[3]　岡本匡代、福田優子、島田京子、直江正美、和田幸裕、田村篤和、西原功修”SX4-2C システム導入について” 1996 年度 技術研究会東京分科会 1996.9.19-20

[4]　福田優子、澤井和美、藤井丈暢、安井秀一 ” 計算機システム運用の自動化(ILE オペレータ)”技術研究会報告集 113 頁～114 頁(1990.12)岡崎国立共同研究機構 分子科学研究所