JAXA Supercomputer System(JSS)の初期性能評価

高木亮治，藤田直行，松尾裕一
宇宙航空研究開発機構

Preliminary Performance Assesment of JAXA Supercomputer System (JSS)

by
Ryoji Takaki, Naoyuki Fujita and Yuichi Matsuo (JAXA)

ABSTRACT

Japan Aerospace Exploration Agency (JAXA) has long recognized the importance of numerical simulations using high performance computer and has strongly promoted the installation and operation of large scale high performance computing system. Since Oct. 2002, JAXA had operated an SMP-cluster type large scale parallel computing system, called NS-III. In April 2009, it was replaced with a multicore based scalable parallel cluster with approximately 12,000 cores, peak performance of 120 Tflops and 94 TBytes of main memory. The new system is called JAXA Supercomputer System (JSS). In this paper, after reviewing the overview of the newly installed system and parallel programming model on the new system, we present the results of preliminary performance evaluation for our current aerospace CFD applications as well as Linpack HPL on JSS.

１．はじめに

宇宙航空研究開発機構（JAXA）は、前身の航空宇宙技術研究所（NAL）および宇宙科学研究所（ISAS）の時代から高性能計算機を用いた数値シミュレーション技術の重要性を認識し、高性能・高機能な大規模計算機システムの整備・運用を積極的に行ってきた。2002年10月から運用してきたNS-IIIシステムに換わるシステムとして2009年4月に新しい大規模並列計算機システムを導入した。新しく導入した計算機システムはJSS（JAXA Supercomputer System）と呼ばれ、JAXA統合後初めての導入となることから、これまで以上に宇宙開発等のJAXA事業への本格的な活用および宇宙三機関統合のシンボル的な位置づけ（One-JAXA）を意図して導入された。

JSSは図1で示す様に複数の計算機システムから構成される複合システムである。JSSの中で実際の計算の中核となるシステムは大規模並列計算機システムおよび共有メモリシステムであり、概要を表1に示す。大規模並列計算機システムはマルチコアCPUをベースにした富士通製FX1と呼ばれるスカラー超並列計算機で構成され、120TFlopsの演算性能と94TBytesの主記憶装置を有するM（メイン）システムと15TFlops、6TBytesのP（プロジェクト）システムから構成される。共有メモリシステムはA（アプリケーション）システムと呼ばれる1TBytesの共有メモリを有する富士通製SPARC Enterprise M9000とV（ベクトル）システムと呼ばれる、4.8TFlopsの演算性能と3TBytesのメモリを有するNEC製ベクトル計算機システムSX-9からなる。

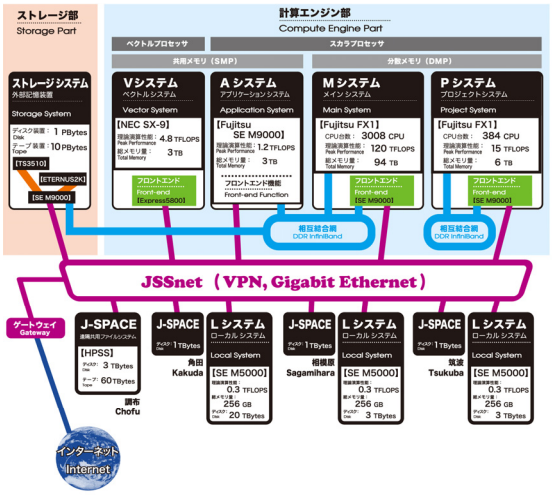


図1：JSSの概要[1)]

表1：JSSの主要計算機システム

名称	JSS-M/P	JSS-A	JSS-V
CPU	Scalar	Scalar	Vector
System	MPP	SMP	SMP
ノード数	3008/384	1	3
CPU/ノード	1	32	16
Core/CPU	4	4	1
論理性能[TFlops]	120/15	1.2	4.8
ノード性能[GFlops]	40	40	1,600
総メモリ[TBytes]	94/6	1	3
メーカー	富士通 FX1	富士通 SEM9000	NEC SX-9

本報告では、JSSにおいて計算の中核となる大規模並

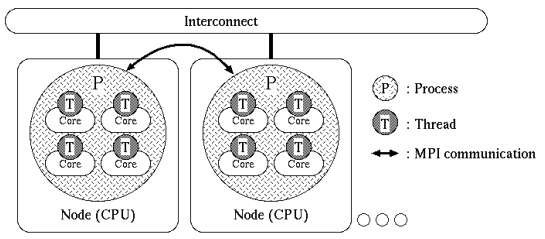
列計算機システム（JSS-M）に関して、新しい並列プログラミングモデルについて紹介する。またJAXAで利用されている代表的なCFDプログラムやベンチマークとして一般的なLinpackなどを用いた性能評価結果について報告する。

２．JSS-Mシステム

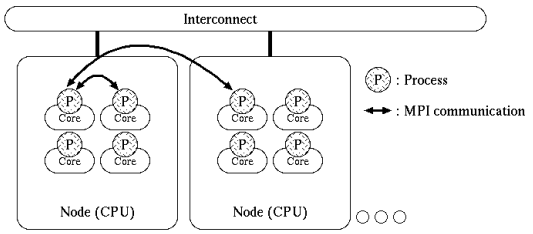
JSS-Mは1つのマルチコアCPUを1ノードとし、ノード間をInfiniband（トポロジーはFAT-Tree）で接合したMPP（Massively Parallel Processor）システムである。JSS-Mでは、並列プロセス数の増加に伴う並列オーバーヘッドとそれによるスケーラビリティ向上のボトルネック解消のために以下の2つの取り組みがなされている。1つ目は並列プロセス数の削減としてマルチコアの有効活用技術、2つ目はプロセス間通信のオーバーヘッド削減としての高機能スイッチの採用である。

2.1 マルチコアの有効活用技術

JSS-MのCPUはSPARC64VIIプロセッサで1個のCPUに4個のコアを搭載し、クロック周波数は2.5GHzとなる。プリフェッチ機能、アウトオブオーダー実行、4個の浮動小数点演算の同時実行機能を有し、1CPUあたりの理論ピーク性能は40GFlopsとなる。このCPUに搭載された4個のコアを利用する技術としてVirtual Single Processor by Integrated MultiCore Parallel ArchiTecture（VisIMPACT）と呼ばれるマルチコアの密結合性を活かした技術が採用されている。H/W技術としてコア間における共有L2キャッシュおよび高速バリアの実現、またS/W技術として自動細粒度並列化コンパイラ技術、更には高いメモリバンド幅（1Byte/Flops）を実現する専用チップセットの連携によりスレッド並列の高性能化を実現している。図2にJSS-Mにおける並列化モデルを示す。マルチコアの代表的な使い方として、VisIMPACTを利用したVisIMPACTモデルと各コアにMPIなどのプロセスを配置するフラットMPIモデル（XPFortranでも同様にVisIMPACTモデルを利用できる）があるが、通常はVisIMPACTモデルを推奨している。VisIMPACTモデルでは、各ノード（各CPU）にMPIなどのプロセスを配置し、ノード（CPU）内の各コアには自動並列によるスレッドを配置するいわゆるハイブリッド並列となる。



a）VisIMPACTモデル



b）フラットMPIモデル

図2：JSS-Mでの並列モデル

VisIMPACTモデルではプロセス（MPI/XPFortranなど）並列に関してはユーザーが明示的に並列プログラムを作成する必要があるが、スレッド並列に関してはコンパイラによる自動並列化が行われ、ユーザーが明示的に並列化を記述する必要はない。コンパイラによる自動並列化は従来も用いられていたが、以下で述べる理由のため性能的に満足のいくものではなかった。もともと自動スレッド並列ではループの並列化が基本となるが、図3で示すように、CFDなどで一般的な多重ループでは自動スレッド並列が対象とするループをどれにするかが問題となる。一般に最外ループでの並列化の方がオーバーヘッドが小さく、高い性能が期待できるため、できるだけ外側ループをスレッド並列の対象ループとするのが有利である。しかしながら実際問題としてはコンパイラの構文解析能力が十分ではなく、ループ構造が少し複雑になると最外ループでの並列化が実施できず、内側ループでの並列化を行う場合が多かった。内側ループでの並列化ではスレッド間の同期など並列化にともなうオーバーヘッドが大きくなってしまい、思ったようにスレッド並列の性能が出せないという問題があった。これに対してVisIMPACTでは、前述したコア間の共有L2キャッシュおよびコア間の高速H/Wバリアによりスレッド並列におけるオーバーヘッドを大きく削減することが可能となり、その結果内側ループのスレッド並列化でも十分な並列性能が出せるようになった。内側ループでのスレッド並列でも十分な性能が出せるようになった事およびコンパイラのデータ依存関係やループ構造などの解析能力の強化により従来よりも自動スレッド並列での性能向上が見込めるようになった。また、最内ループを対象に並列化を実施する事はベクトル化と同じ方法論となり、最内ループでの高効率な自動スレッド並列化の実現によりベクトル計算機向けに作成されたプログラムでも、そのままで自動スレッド並列での性能が期待できるようになった。ベクトル化の場合は、一部のデータ依存関係（逆方向依存）が存在する場合はベクトル化ができなかったが、VisIMPACTではその様な場合でも自動スレッド並列化が可能となり、より広い範囲に適用できることとなった。

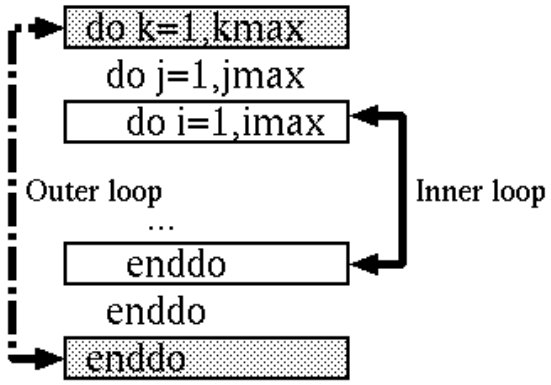


図3：自動スレッド並列化対象ループ

図4にVisIMPACTの有効性の一例を示す。この図ではJSS-Mの1CPU（VisIMPACTあり）とJSS-M（FX1)とほぼ同じスペックを有する富士通HX600（VisIMPACTなし）の1CPUの性能をベンチマーク（DAXPY）で比較したものである。HX600のCPUはAMD ShanghaiでJSS-MのCPUと同じCPUクロック（2.5GHz）、コア数(4)を有する。HX600ではVisIMPACTを使わない富士通の自動並列コンパイラによる自動スレッド並列を行っている。この図で横軸はループ長、縦軸は性能を表しているが、ループ長が短い細粒度並列においてVisIMPACTの有効性が確認できる。

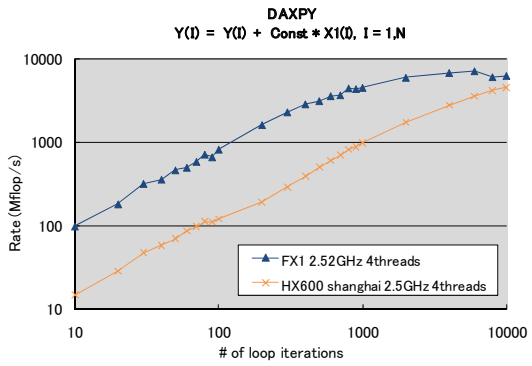


図4：VisIMPACTの有効性（DAXPYベンチ）

2.2 高機能スイッチ

プロセス数増加にともない、プロセス間での集合通信やバリア同期の高速化が必要となる。そのためにJSS-Mでは通常のプロセス間でのデータ通信を行うInfinibandのネットワークとは別に、バリア同期や集合通信を専門に行うネットワークおよび高機能スイッチを採用した。この高機能スイッチでは、プロセス間の集合通信やバリア同期機能を専用のH/Wを用いて実装しており、S/Wによる処理に比べてより高速な処理を実現した。高速なバリア同期機能はOSジッタの対策としても有効に機能し、リダクション演算が少ないアプリケーションでも、有効な機能である。図5に高機能スイッチの有無によるMPIバリア性能の測定結果を示す。

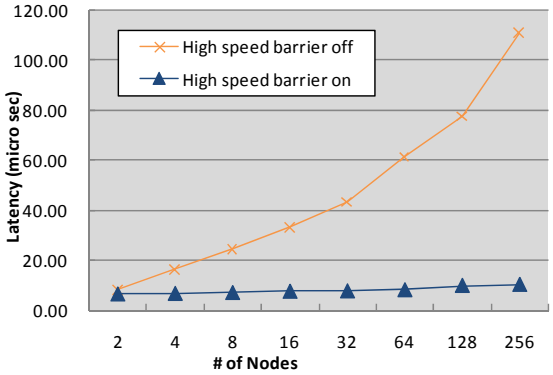


図5：高機能スイッチの性能（MPIバリア）

図では256ノードまでの測定結果が示されているが、ノード数が増えても高機能スイッチの効果でバリア同期のコストがほとんど増加しないことがわかる。一方、高機能スイッチを利用しない場合は、ノード数の増加にともない、飛躍的にバリア同期のコストが増加していることがわかる。高機能スイッチは最大768ノードまで結合することが可能であり、768ノード内ではここで測定されたレベルの高性能が期待できる。768ノードを超えた場合はS/Wによるバリア同期となるため、例えば1024ノードでは248.51マイクロ秒となり極端に性能が悪化する。768ノードを超えた大規模並列解析におけるバリア同期の高速化は今後の課題である。

3．性能評価

JSS-Mに関して一般的なベンチマークのLinpackや実際にJAXAで使われている実アプリケーションによる性能評価を行った。

3.1 Linpack HPL

高性能計算機の代表的なベンチマークであるLinpack HPL を用いてJSS-Mの性能評価を行った。JSS-M向けにチューニングを行い表2の結果を得た。

表2：Linpack HPL 実行結果

ノード数	3,008
コア数	12,032
Rmax	110.6TFlops
Rpeak	121.3TFlops
実行効率	91.19%
実行時間	60時間40分

測定の結果91.19%という高い実行効率を得たが、世界でもトップクラスの性能である。またLinpackは計算負荷が大きくシステムに大きな負担がかかる計算である。

Linpackを実行したのはJSS導入初期であるが、この時期にトラブルも発生せずに60時間強の計算実行を完遂できたことはシステムとしての高い安定性と信頼性を実証できたと考えられる。

3.2 JAXAアプリによる性能評価

JAXAの代表的なアプリケーション（JAXAベンチ）による性能評価を行った。表3にJAXAベンチの概要を示す。

表3：JAXAベンチ一覧

名称	適用先	計算手法	並列化
P1	燃焼	FDM+化学反応	MPI+VisIMPACT
P2	航空	FVM（構造）	MPI+VisIMPACT
P3	乱流	FDM+FFT	XPF+VisIMPACT
P4	プラズマ	PIC	MPI+VisIMPACT
P5	航空	FVM（非構造）	MPI+VisIMPACT

図6は測定したJAXAベンチの特性を示す。この図では計算負荷、メモリアクセス負荷、データ通信負荷のうち、どの特性が相対的に顕著かを示している。図で上部に位置するアプリケーションは相対的にデータ通信の割合が大きく、右領域に位置するアプリケーション程メモリアクセスの割合が大きいことを示している。左下に位置するアプリケーションは計算負荷が大きいことを示す。そのため、各アプリケーションの特性としてはP1とP4は演算負荷が大きいアプリケーション、P2とP5はメモリアクセス負荷が大きいアプリケーション、P3はネットワーク負荷が大きいアプリケーションである。

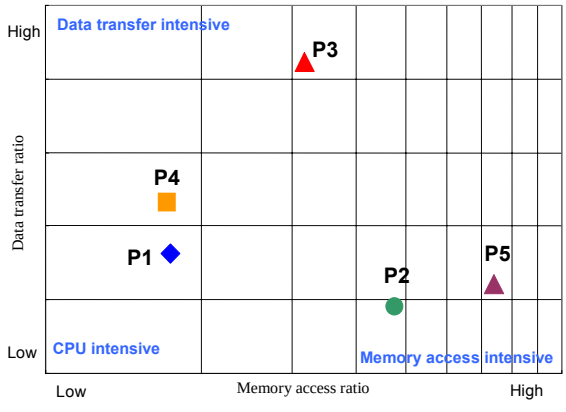


図6：JAXAベンチの特性

表4に測定結果を示す。この結果よりJAXAの実アプリケーションに対してJSS-MはCeNSS（以前のシステムであるNS-IIIの中核計算機）より平均で11倍以上高速であることがわかる。P2とP4の性能比が他よりも高いが、P2は自動スレッド並列コンパイラの性能向上によりスレッド並列の範囲が広がったためと考えられる。またP4はMPIの集合通信の改善が主な理由と考えられる。逆にP3が悪いのは他アプリケーションと異なり、演算負荷に対して相対的にデータ転送負荷が大きく経過時間ではほぼ同程度となる。JSS-Mでは通信性能比（対CeNSS比で2倍にしかならない）は演算性能比（周波数：2倍、コア数：4倍、その他）に比べて悪いため、全体の性能比が悪くなったと考えられる。

表4：JAXAベンチによる性能評価

名称	CPU数	CeNSS [sec]	JSS-M [sec]	性能比
P1	744	1380.4	143.3	9.63
P2	750	1468.6	91.5	16.05
P3	512	3517.0	491.7	7.15
P4	750	3061.7	193.0	15.86
P5	750	1447.2	181.6	8.13
平均	-			11.36

JAXAの代表的な実アプリの一つであるUPACS[2)]を用いてJSS-Mの性能評価を行った。UPACSは構造体、動的配列、ポインター、モジュールといったFortran90の機能を活用して作成された汎用的な圧縮性流体解析プログラムであり、3次元マルチブロック構造格子および重合格子を扱うことができる。UPACSを用いたスケールアップ評価を図7に示す。

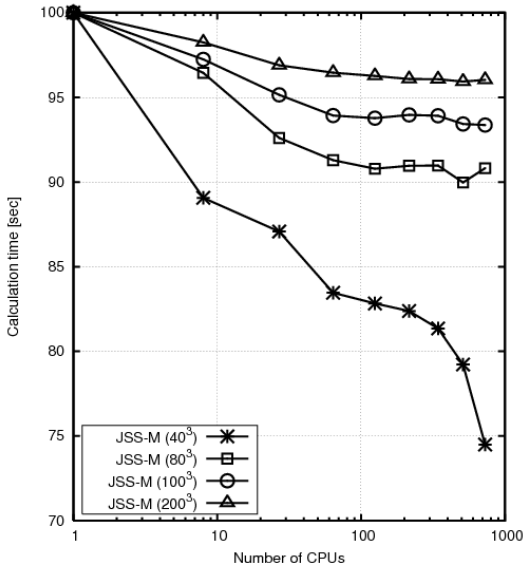


図7：UPACSによるスケールアップ評価

使用するCPUを増やす度に計算規模を増大させるスケールアップ評価では729CPUまで74%（ブロックサイズは403）～96%（ブロックサイズは2003）といった良い並

列効率を示した。ブロックサイズが大きくなると相対的に並列性能は良くなる。何故なら、ブロックの１辺をNとすると演算量はN3に対して通信量はN2に比例するため、Nが大きくなる、つまりブロックサイズが大きくなると相対的に通信によるオーバーヘッドが減少するからである。

図8に並列モデル（フラットMPIモデルとVisIMPACTモデル）の違いによる性能差を示す。

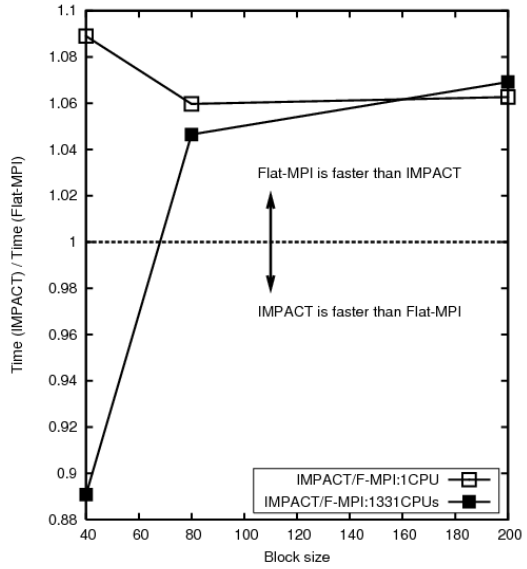


図8：並列モデルの比較

横軸はブロックサイズ、縦軸はVisIMPACTとフラットMPIとの計算時間の比を示している。図の上半分はフラットMPIの方がVisIMPACTよりも高速であることを示す。今回の計測条件ではVisIMPACTとフラットMPIの性能比は10%程度であることがわかる。VisIMPACTとフラットMPIで、どちらが良いかは色々な条件の影響を受ける。例えば、CPU内並列を考えた場合、VisIMPACTの並列性能は自動並列コンパイラの能力に依存するが、フラットMPIではユーザーが明示的に並列化を行うことで高い並列性能が期待できる。一方、プロセス間通信を考えた場合、VisIMPACTはプロセス数を減少させることができるので、高プロセス並列においても性能劣化が低いが、フラットMPIの場合はプロセス数がコア数倍増えるため、高プロセス並列においては性能劣化が大きいと考えられる。またアプリ側の問題として計算格子のブロックサイズが大きくなるとフラットMPIが有利となる。これは前述したようにブロックサイズが大きくなると通信のオーバーヘッドが小さくなるからである。この影響は図からも読み取れ、プロセス数を固定（1331）した場合、ブロックサイズが小さくなる（40）と、プロセス並列のオーバーヘッドが大きくなり、フラットMPIよりもVisIMPACTの方が有利となっている。ここではプロセス数が1331でしか計測できていないが、プロセス数が増えるとこの性能が逆転するブロック数は大きくなると考えられる。アプリケーションの特性に大きく依存するが、現状のJSSの規模ではVisIMPACTよりもフラットMPIの方が若干有利かもしれないが、今後更なるスケールアップ（コア数、ノード数）を考えた場合、フラットMPIの限界が見えてきたと考えている。

次にこれもJAXAの代表的なアプリの一つであるLANS3D[3]を用いてJSS-MおよびJSS-Vの評価を行った。LANS3Dは航空宇宙分野で比較的初期に開発された先駆的なCFDプログラムでFortran77をベースに構造化プログラミング的な考え方で設計され、主にベクトル計算機向けに実装された典型的な圧縮性流体解析プログラムである。並列化に関しては基本的に自動スレッド並列による並列化を行っている。プロセス並列化としてはMPIを用いた領域分割に一部対応しているが、多数プロセスによる並列計算は現実的でないためここでは最大8プロセスまでとした。このLANS3Dを用いてJSS-M/Vのスピードアップ評価を行った。

表5：LANS3Dによるスピードアップ評価

システム	プロセス	スレッド	(ブロックサイズ) ×ブロック数	並列手法
JSS-M (FX1)	1	4	(320x320x320) x1	VisIMPACT
	2	8	(320x320x160) x2	MPI + VisIMPACT
	4	16	(320x160x160) x4	
	8	32	(160x160x160) x8	
JSS-V (SX-9)	1	1,2,4,8,16	(320x320x320) x1 (160x160x160) x8	自動並列
SSS (SX-6)	1	1,2,4,8	(320x320x320) x1 (160x160x160) x8	

問題規模として約3300万点の計算格子を固定して、CPU数を増やすことによる計算速度の向上を評価した。表5に計算の概要を示す。JSS-MではVisIMPACTを用いて、プロセス数として1～8プロセスまで測定を行った。プロセス数に応じて計算格子をブロック分割し1CPUに1ブロックを割り当てた。JSS-VおよびSSS（現在も相模原で運用中のNEC製SX-6）ではノード内並列に限定し、自動スレッド並列を用いてスレッド数1～16（SSSは8まで）で計測を行った。またベクトル長の影響を見るため、320x320x320が1ブロックの計算と160x160x160が8ブロックの計算を行った。

測定結果を図9、10に示す。図11は計算時間を比較したもの、図12は相対的実行効率を比較したものである。ここで相対的実行効率は計算時間の逆数をCPUのピーク性能で割った値であり、単位ピーク性能あたりの計算速度を示す。そのためこれらの値の比較は実行効率の比較と同じ意味を持つ。相対的実行効率は通常の実行効率とは異なり、比較を行う場合同じ問題規模でなければ意味がない（もちろん、格子点あたりの値を計算することで異なる問題規模での比較もある程度可能であるが）など利用範囲が限定されるが、ユーザーの実感に近い、計測が容易といったメリットがあるため、ここでは相対的実行効率を比較した。図9のグラフより、JSS-M(FX1)、JSS-V(SX-9)、SSS(SX-6)とも並列化効率（スピードアップ性能）はほぼ同じである。図10よりSSS(SX-6)とJSS-M(FX1)では実行効率で3倍程度の違いが見られるが、JS-V(SX-9)とJSS-M(FX1)では約2.5倍程度の違いに縮小していることがわかる。またJSS-V(SX-9)はSSS(SX-6)に比べてベクトル長が短くなると性能が悪化することもわかる。

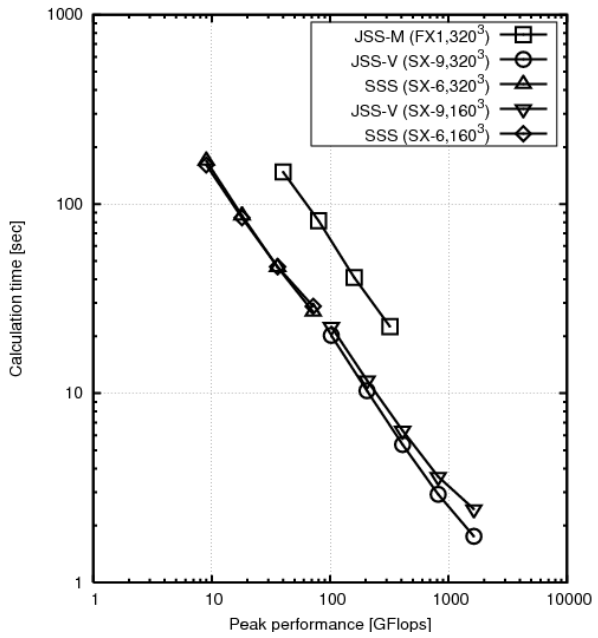


図9：計算時間の比較（LANS3D）

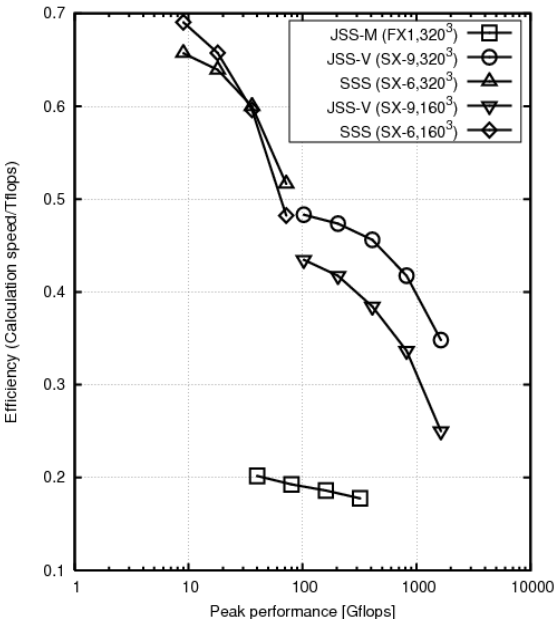


図10：相対的実行効率の比較（LANS3D）

４．おわりに

JAXAの新しい計算機システムJSSの中核となる大規模並列計算機システム(JSS-M)についてシステム概要、並列化手法について紹介した。また共有メモリシステム（JSS-V）も含めて性能評価結果を紹介した。今後も引き続き性能評価を予定している。

謝辞

本稿の執筆に際し、JSS運用チームおよび富士通株式会社の関係各位にご協力いただきました。ここに記して深く感謝いたします。

参考文献

1. 藤田直行、高木亮治、松尾裕一、“JAXA Supercomputer System (JSS)の構成と特徴”、第41回流体力学講演会/航空宇宙数値シミュレーション技術シンポジウム2009、2D1, 2009
2. Takaki, R., et al, “The Development of the UPACS CFD Environment”, Lecture Notes in Computer Science 2858 Springer (2003), pp307-319, 2003
3. Fujii, K. and Obayashi, S., “High-Resolution Upwind Scheme for Vortical-Flow Simulations”, Journal of Aircraft, Vol. 26, No.12, pp.1123-1129,1989