HITACHI SR16000/M1 チューニング講座
1. ハードウェア概要

大島聡史
東京大学情報基盤センター 助教

1 はじめに

本稿ではHITACHI SR16000 モデルM1（以下SR16000）のハードウェア概要について解説します。プログラムを最適化するためには対象ハードウェアについて理解することが非常に重要です。正しく理解したうえで最適化プログラミングに挑んでください。

2 全体構成

図1にSR16000の全体構成図を示します。また表1にはSR16000の性能諸元を旧機種であるHITACHI SR11000/J2（以下SR11000）と比較して示します。SR16000は56の計算ノード、556TBのストレージ、高速な内部ネットワーク、そしてログインノードや各種管理用ノードから構成される計算機システムです。特に計算ノードとログインノードは高密度に搭載されており、計算ノード56ノードとログインノード2ノードが水冷2ラックに収められています。

表 1　SR16000 と SR11000 の比較

	SR16000	SR11000（旧システム）
CPU	Power7 3.83 GHz	Power5+ 2.30 GHz
ノード数	56	128
コア数/計算ノード	32	16
理論演算性能/コア	30.64 GFLOPS	9.2 GFLOPS
理論演算性能/計算ノード	980.48 GFLOPS	147.2 GFLOPS
理論演算性能/全計算ノード	54906.88 GFLOPS	18841.6 GFLOPS
主記憶容量/計算ノード	200 GB	128 GB
主記憶容量/全計算ノード	11200 GB	16384 GB
Byte/FLOPS 値	0.52	1.39
SMT 機能	最大 4 スレッド/コア	非対応
計算ノード間 ネットワーク構成	階層型完全結合	3 段クロスバー
計算ノード間転送性能	96GB/s（単方向）× 双方向	12GB/s（単方向）× 双方向
ストレージ容量	556 TB	94.2 TB
LINPACK 性能値	0.8075 TFLOPS （1 計算ノード） 6.46 TFLOPS （8 計算ノード）	15.81 TFLOPS （全計算ノード）

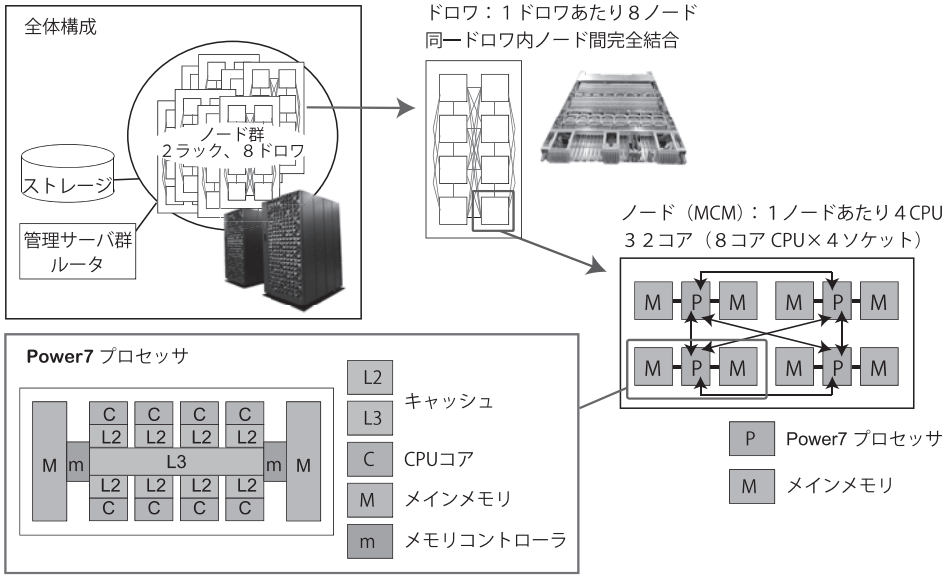

図1　SR16000 構成図

3　CPU構成

SR16000は計算ノードのCPUとしてPower7を搭載しています。旧機種であるSR11000がPower5+を搭載していたため、基本的なアーキテクチャはPower系で引き継がれています。プログラムの作成方法・最適化方法については本連載中に別の記事で解説しますが、SR16000ではSR11000向けに作成したプログラムを基本的には再コンパイルするのみで動作させることができます。

Power7の構成を図1の左下部に示しています。Power7は8つのコアによって構成されているマルチコアプロセッサであり、1ノードにはPower7 CPUが4基搭載されています。ノードを構成する4CPUはMulti Chip Module (MCM)とも呼ばれます。ラックへの搭載については、8ノードからなるドロワ（図1を参照、1ドロワは畳ほどの大きさである）を1単位として行われています。

SR16000に搭載されているPower7の動作周波数は3.83GHzであり、理論倍精度浮動小数点演算性能は1コアあたり30.64GFLOPS[*1]、1CPUあたり30.64GFLOPS×8コア=245.12GFLOPS、1ノードあたり245.12GFLOPS×4CPU=980.48GFLOPS、計算ノード群全体では980.48GFLOPS×56ノード=54.90688TFLOPSです。SR11000と比較すると、ノードあたりでは約6.7倍、計算ノード群全体では約2.9倍の演算性能を備えています。またキャッシュについては、L1キャッシュをデータと命令それぞれにコアごとに32KB、L2キャッシュをコアごとに256KB、L3キャッシュを1CPUごとに32MB搭載しており、L1キャッシュはパリティ、L2およびL3キャッシュはECCによって保護されています。

[*1] (乗算 2FLOP+ 加算 2FLOP)×2 演算器 ×3.83GHz=30.64GFLOPS

Power7はSMT(Simultaneous Multi Threading)に対応しているため、状況に応じて最大で1コアあたり4スレッドを同時に処理することが可能です。ただしSMTは数を増やすほど高い性能が得られるとは限らないため、本システムではSMT数を2に設定して運用しています。

CPUの演算性能が主に問われるLINPACKベンチマーク[*2]による性能測定では、1ノードで0.8075 TFLOPS、8ノードで6.46 TFLOPSの性能値が得られています。理論演算性能に対する割合は1ノード・8ノードともに82.4%です。

4　メモリ構成

Power7 CPUとメインメモリ（主記憶）の接続については、図1中に示すように各CPUに搭載されたメモリコントローラを介して接続されています。そのためCPUコアがメモリアクセスを行う際に、対象のメモリがどこに存在するか、すなわちローカルなCPUに接続されたメモリなのか他のCPUに接続されたメモリなのかによってメモリアクセス性能に差が生じます。このようなアーキテクチャをNUMA(Non-Uniform Memory Access)アーキテクチャと呼びます。SR16000におけるプログラム最適化・性能チューニングの際には、SR16000はSR11000とは異なりNUMA構成であることを考慮する必要があります。

SR16000は計算ノード1ノードあたり200GB、計算ノード群全体では11200GBのメインメモリを搭載しています。メモリ容量については表1に示したように、SR11000と比べると、計算ノード1ノードあたりのメモリ量は128GBから200GBへと増加していますが、計算ノード群全体のメモリ量は16384GBから11200GBへ、また1コアあたりでは8GBから6.25GBへと減少しています。SMTを用いる場合はコア数が2倍となるためさらに半減します。なお、一部のメモリはシステムが占有するため、これらのメモリを利用者が全て自由に使えるわけではありません。実際に利用者が使えるメモリは、SR11000では112GB、SR16000では170GBです。

メインメモリの種別については、DDR3 SDRAMメモリが搭載されています。また計算ノード上の各CPUコアと主記憶間の物理転送性能の合計値は512GByte/秒です。計算ノード1ノードあたりの性能としてはSR11000の204.6GB/sから大きく向上しています。一方で1ノードあたりの演算FLOPS値あたりメモリ性能GByte/s値（B/F値）については、SR11000が1.39であったのに対してSR16000では0.52（SMTについて考慮しない場合）へと減少しています。B/F値の観点からは、SR16000はSR11000と比べると計算インテンシブなアプリケーションに適したシステムであると考えることができます。なお、SR16000のB/F値はSR11000と比べると低い値ですが、HA8000（32GBメモリノード）のB/F値0.28と比べると高い値であることがわかります。

メモリの性能を測定するSTREAMベンチマーク[*3]の結果としては、計算ノード1ノードあたり217GByte/秒の値が得られています。

5　ノード間ネットワーク構成

SR16000のノード間ネットワーク構成の概要を図2に示します。SR16000のネットワーク構成は、1ドロワ内の8ノードが完全結合しており、さらにドロワ単位でも完全結合しているという階層型の完全結合となっています。そのため計算ノード群から任意の複数ノードを抽出して

[*2] http://www.netlib.org/benchmark/hpl/
[*3] http://www.cs.virginia.edu/stream/

通信を行うと、ノードの組み合わせによっては完全結合ではない組み合わせとなり、理論上は通信性能が低下する可能性があります。実際には性能への影響はほぼ無いとされていますが、今後の性能調査等において影響が明らかになった際にはスパコンニュース記事等にて改めて報告致します。

ノード間ネットワークの性能については、96GB/秒(単方向)×双方向です。

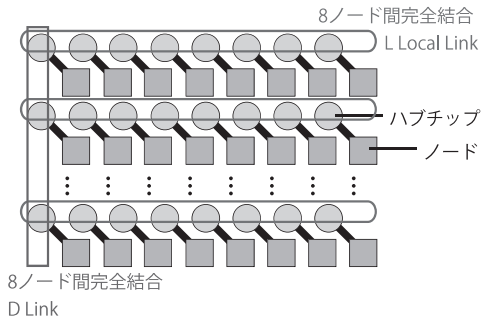

図 2　ネットワーク構成

6　ストレージ構成

SR16000は16台のHitachi Adaptable Modular Storage 2500からなる556TByteの共有ストレージを備えています。ファイルシステムについてはGeneral Parallel File System for AIX (GPFS)を採用しています。計算ノード群とストレージは64本の8Gbps Fibre Channelで接続されています。全ての計算ノードおよびログインノードからストレージに対して均一にアクセスすることが可能であり、総物理転送性能としては読込み40GByte/秒、書込み16GByte/秒の性能を持ちます。

以上、本稿ではSR16000のハードウェア概要について、一部ベンチマークの結果等を交えて紹介しました。さらに詳しい情報やベンチマークの結果等については、本連載における他の記事や、利用者向けに公開されている資料等[*4] を参照してください。

[*4] https://yayoi-man.cc.u-tokyo.ac.jp/manual-j/index.html