インテル® Xeon® プロセッサー E7 ファミリー時代のマルチプロセッサー環境での高速化のアプローチと留意点

パフォーマンスのボトルネック解消法

インテル® Xeon®
プロセッサー E7 ファミリー

テクニカルホワイトペーパー

目次

マルチプロセッサー環境の高速化

インテル® Xeon®
プロセッサー E7 ファミリー

はじめに

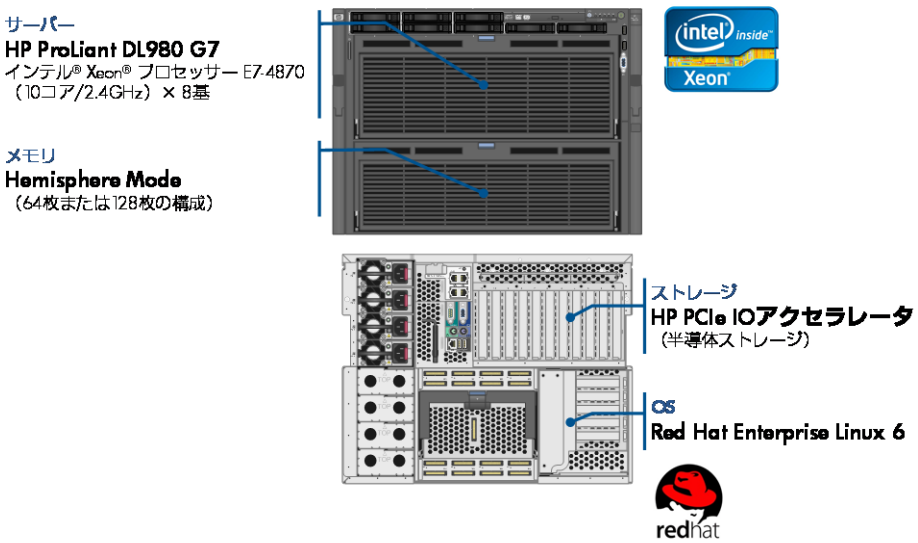
近年、ネットワークを介したビジネスやコミュニケーションが日常化したことで、企業が扱うデータは爆発的に増加しています。IT の成熟期を迎えた今日では、ビジネスを IT 化するだけに留まらず、従来の IT インフラの効率化や高速化が求められ、その要求を解決するための様々なアプローチが取られています。

本書では、高速化のアプローチのひとつであるマルチプロセッサー環境での「スケールアップ」について説明します。さらに、インテル、レッドハット、日本 HP の 3 社にて実施しました共同検証を例に、その際の問題点と解決策をご紹介いたします。

本書での、マルチプロセッサー環境での高速化のアプローチを実現するための推奨構成は以下の通りです（図 1）。

- サーバー ..... HP ProLiant DL980 G7
- CPU ..... インテル® Xeon® プロセッサー E7-4870（10 コア/2.4GHz）
- メモリ ..... Hemisphere Mode（64 枚または 128 枚の構成）
- ストレージ ..... HP PCIe IO アクセラレータ（半導体ストレージ）
- OS ..... Red Hat Enterprise Linux 6

図 1: 高速化のアプローチを実現するための推奨構成



高速化のアプローチの推奨マルチプロセッサー環境

インテル® Xeon®
プロセッサー E7 ファミリー

それぞれの製品概要と特長は以下の通りです。

HP ProLiant DL980 G7

HP ProLiant DL980 G7 はインテル® Xeon® プロセッサー E7 ファミリーを最大 8 基搭載し、80 コア/160 スレッドまでの拡張性を実現するラックマウント型サーバーです。

HP ProLiant DL980 G7 は、最大 2TB 搭載できる 128 のメモリスロット、16 基の PCI-Express スロット（フルハイト×11 基、ハーフハイト×5 基）を内蔵しています。また、HP が開発した数々の独自技術を組み合わせた「PREMA アーキテクチャ」を採用している製品です。

インテル® **Xeon®** プロセッサー **E7** ファミリー

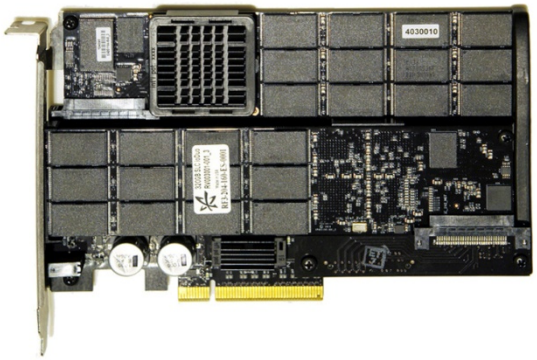
最高水準のインテル® Xeon® プロセッサー E7 ファミリーは、優れた拡張性と大容量メモリおよび I/O により、極めて大量のデータ処理を伴うワークロードに最適な性能を提供します。短期的なビジネスニーズの変化に容易に対応し、長期的なビジネスの成長にも対処できます。先進の信頼性とセキュリティー機能により、データの完全性確保や暗号化処理の高速化に加え、重要なアプリケーションで最大限の可用性を実現します。

HP PCIe IO アクセラレータ

HP PCIe IO アクセラレータは、不揮発性メモリ回路のフラッシュメモリを使用した直接接続型の PCIe カードベースの製品です（写真 1）。

優れたデータ読み取り/書き込み速度により、アプリケーションのパフォーマンス向上に寄与します。関連するアプリケーションのパフォーマンスが向上すると、ビジネス成果によい影響がもたらされ、迅速な意思決定が可能になり、コストと時間が大幅に節約されます。

写真 1: HP PCIe IO アクセラレータ



Red Hat Enterprise Linux 6

Red Hat Enterprise Linux 6（RHEL6）は、2010 年 11 月にリリースされた Red Hat 社の最新のバージョンの企業向け Linux ディストリビューションです。

RHEL6 では、OS が制御可能なシステム上の限界値が大きく向上しました。取り扱える CPU の最大数は 4096（CPU 論理コア数）、メモリは最大 64TB まで利用可能です。これらのリソースを効率よく利用する仕組みとして「cgroup」やプログラムを自動並列化する「OpenMP」、仮想化機能の「KVM（Kernel-based Virtual Machines）」が、RHEL6 では用意されています。

インテル® Xeon®
プロセッサー E7 ファミリー

パフォーマンスのボトルネック解消法

インテル® Xeon®
プロセッサー E7 ファミリー

パフォーマンスのボトルネックとなる問題点

一般的に、パフォーマンスのボトルネックは、CPU 使用率やメモリの使用量と思われがちですが、実際には様々な要因があります。その他の主な要因として、ストレージ I/O のボトルネックと、アプリケーションのボトルネックがあります。ここでは、それぞれの問題点の解消法をご紹介します。

- CPU/メモリ
- ストレージ I/O
- アプリケーション

CPU/メモリのボトルネック解消法

CPU やメモリがパフォーマンスのボトルネックとなっている場合、効果的なアプローチのひとつとして、マルチプロセッサー環境での「スケールアップ」があります。ただし、「スケールアップ」の場合、リニアにパフォーマンスが向上する適切なシステムを選択する必要があります。

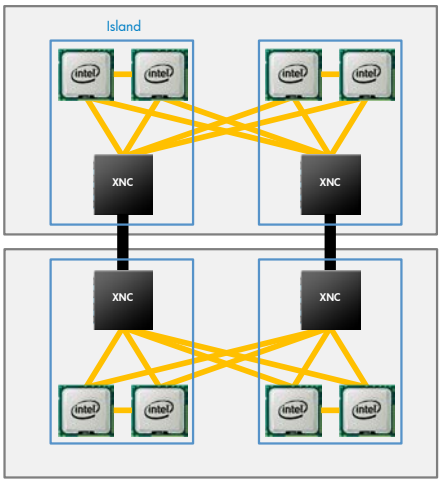
CPU：ポイントツーポイント接続

マルチプロセッサーのサーバー選択上、考慮すべき点はプロセッサー間がポイントツーポイント接続であることです。インテル® Xeon® プロセッサー E7 ファミリーは、グルーレス構成で 2 ソケット、4 ソケット、そして 8 ソケット構成をサポートできるサーバー向けのマルチプロセッサー（MP）ですが、8 ソケット構成時にはポイントツーポイント接続されていません。そのため、4 ソケット構成まではプロセッサー数に比例したスケールアップ性能を示しますが、8 ソケット構成ではプロセッサー数に比例したスケールアップ性能を発揮しません。

HP ProLiant DL980 G7 は、独自開発したノードコントローラ（XNC：eXternal Node Controller）を搭載することで、8 ソケット構成でそれぞれのプロセッサーがポイントツーポイント接続できるよう設計しています（図 2）。インテル® Xeon® プロセッサー E7 ファミリーを 2 つで「Island」と呼ばれる単位で構成し、XNC にポイントツーポイント接続しています。この「Island」を XNC 経由で 4 つ組み合わせ 8 ソケット構成のサーバーデザインにしています。さらに XNC 間は QPI（Quick Path Interconnect）ではなく HP 専用の高速シリアルバスで接続した結果、4 ソケットに比べ 1.8 倍程度のスケールアップ性能が向上するようになっています。

インテル® Xeon®
プロセッサー E7 ファミリー

図 2: HP ProLiant DL980 G7 のブロック図



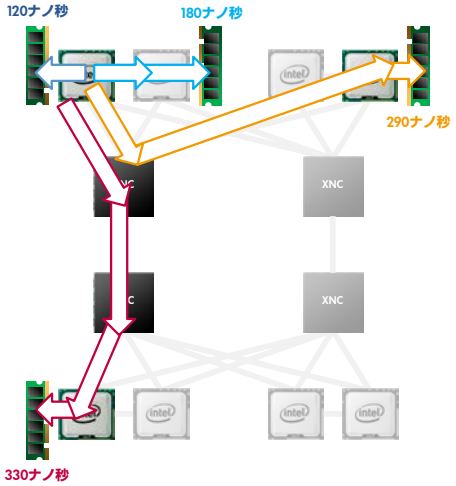
メモリ：キャッシュコヒーレンシー処理

メインメモリのアクセスを高速化させるには、インテル® Xeon® プロセッサー E7 ファミリーに内蔵されるメモリコントローラからのメモリ帯域を全て活用する Hemisphere 構成にする必要があります。つまり、メモリ帯域を全て活用するためには 1 プロセッサー当たり 8 枚または 16 枚の RDIMM、8 ソケット構成では 64 枚または 128 枚の RDIMM で構成することでメモリ帯域性能を最大限にすることができます。

また、8 ソケット構成時、ローカルメモリのアクセスレイテンシーは 120 ナノ秒ですが、物理的に一番遠い位置となるリモートメモリのアクセスレイテンシーは 330 ナノ秒となります。そのためメモリアクセスレイテンシーが低いローカルメモリを優先して動作させることが必要となります。これは OS がメモリの物理的な位置を理解して動作する NUMA（Non Uniformed Memory Access）対応することでローカルメモリを優先させる動作をすることができます（図 3）。

しかし、OS が NUMA 対応してもインテル® Xeon® プロセッサー E7 ファミリーのマルチプロセッサー構成ではローカルメモリのデータ読み込みと同時に搭載されている全てのインテル® Xeon® プロセッサー E7 ファミリーにキャッシュスヌープを発行するため、このキャッシュスヌープが完了しなければローカルメモリからの読み込みを完了させることができません ※1。

図 3: HP ProLiant DL980 G7 の NUMA 対応



インテル® Xeon®
プロセッサー E7 ファミリー

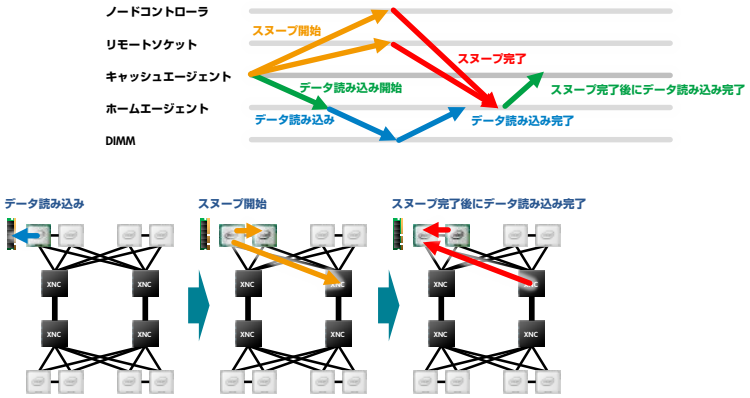
※1：キャッシュスヌープを高速処理させるには“そのローカルメモリのデータが他のインテル® Xeon® プロセッサー E7 ファミリーのキャッシュに保持されていない”ことが前提となるためで、OS とアプリケーションが NUMA 対応であることが必須になります。NUMA 対応であればローカルメモリに保存されているデータを優先して動作するため、ローカルメモリのデータが別のプロセッサーに書き換えられている確率を低く抑えさえられ、キャッシュコヒーレンシー処理を短時間で完了できます。

インテル® Xeon® プロセッサー E7 ファミリーには、グルーレス構成でローカルメモリの読み込み性能を向上させる DAS（Directory Assisted Snoopy）機能があります。HP ProLiant DL980 G7 では、よりキャッシュの応答を早くするために本機能を使用せず、キャッシュスヌープをインテル® Xeon® プロセッサー E7 ファミリーとポイントツーポイント接続している XNC が応答することで完了させ、ローカルメモリの読み込み速度を高速する方式「スマート CPU キャッシング」で高速処理させています（図 4）。スマートCPUキャッシングは一般的にはスヌープフィルタリングと呼ばれています。またXNCはスマートフィルタリング機能付きノードコントローラと呼ばれています。

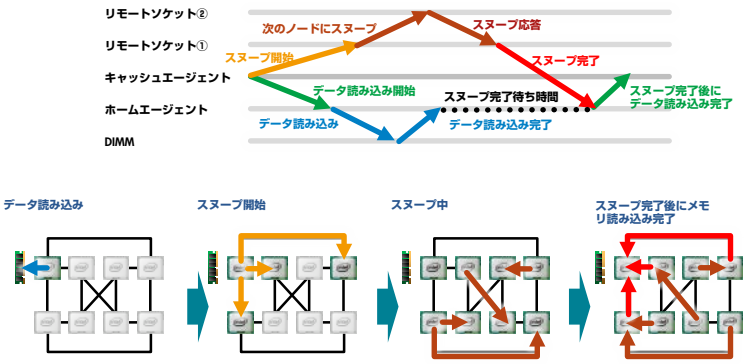
インテル® Xeon®
プロセッサー E7 ファミリー

図 4: スマート CPU キャッシング

スマート CPU キャッシング



グルーレス構成



ストレージI/Oのボトルネック解消法

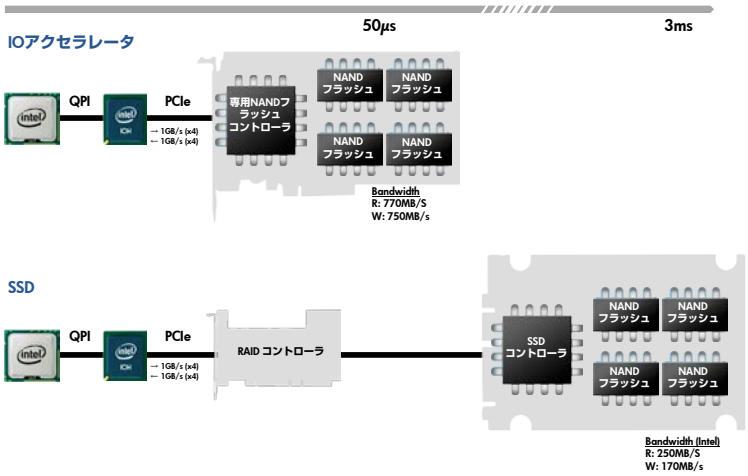
マルチコア化やメモリの大容量化などが進み、CPU/メモリは容量や性能が大幅に向上していますが、最も一般的な外部記憶装置である HDD は、大容量化は進んでいますがアクセスのスピードはそれに追いついていないのが現状です。この性能のギャップがストレージ I/O のボトルネックとなります。

このボトルネックを解消する方法としては、高速な外部ストレージシステムを利用することが一般的となっていますが、外部ストレージほどの容量が必要でなくサーバー単体でストレージ I/O のボトルネックを解消したい場合は、内蔵の HDD を SSD（Solid State Drive） にするという選択があります。

インテル® Xeon®
プロセッサー E7 ファミリー

更にディスクアクセス性能が求められる場合には、フラッシュメモリを搭載する半導体ストレージ「HP PCIe IO アクセラレータ」によって解決できます。HP PCIe IO アクセラレータは PCI Express スロットに空きがあれば搭載することができ、SAS/SATA が内部的に使用しているレガシーな SCSI/IDE プロトコルへの変換処理を介さずホスト側のドライバーミドルウェアでフラッシュメモリを直接ハンドリングすることにより、50 マイクロ秒以下と従来ストレージとの比較で 2～3 桁低いレイテンシを実現するほか、コンシューマ向け SSD と比較して読み込み 3 倍・書き込み 6 倍の帯域幅を誇り、IOPS 性能および帯域幅を高めることも可能です。実際、IO アクセラレータのスループットは、読み込み最大 700MB/秒および書き込み最大 600MB/秒で SSD よりも高い性能を発揮しています（図 5）。

図 5: HP PCIe IO アクセラレータと SSD の比較



アプリケーションのボトルネック解消法

CPU、メモリ、ストレージ I/O 以外のパフォーマンスのボトルネックに、アプリケーションのボトルネックがあります。アプリケーションによっては、「スケールアップ」の場合にスレッド性能の限界に達することがあるため、その限界を超えたスレッド以上は、性能が劣化します。

Red Hat Enterprise Linux 6（RHEL6）は、スケーラビリティを高めるために、OS の中核であるカーネルのスケジューラー部分の変更を加え、マルチコア環境に最適化された新スケジューラーの「CFS（Completely Fair Scheduler）」を採用しています。CFS は、タスクが CPU を利用してよい時間の公平性（バランス）を保つように、タスクの待ち時間をコントロールする仕組みを持つ特長があります。

また、マルチコア環境のリソースを効率よく利用する仕組みとして「cgroup（Control Groups）」やプログラムを自動並列化する「OpenMP」、仮想化機能の「KVM（Kernel-based Virtual Machines）」が、用意されています。

インテル® Xeon®
プロセッサー E7 ファミリー

処理性能劣化を抑える「**cgroup**」

cgroup は、厳密には性能を上げるための機能ではなく、性能劣化を抑える仕組みです。この仕組みは商用 UNIX では実装されていた仕組みで、HP-UX では「HP-UX Process Resource Manager（PRM）」にあたります。cgroup を使うことによって、マルチスレッドアプリケーションの性能劣化が起きにくいようにアプリケーションを改変することなくリソースの割り当てを行うことができます。

リソースを分割する手段のひとつに仮想化技術がありますが、サーバー仮想化時にはオーバーヘッドが生じます。cgroup では仮想化せずに物理サーバー上の OS だけで実現します。OS 上で動くアプリケーションに対して使用するリソースを指定するだけでオーバーヘッドは生じません。

ベンチマークによるボトルネック解消の実例

インテル® Xeon®
プロセッサー E7 ファミリー

ベンチマークの目的

インテル、レッドハット、日本 HP の 3 社にて共同検証を行いました。この検証では、HP ProLiant DL980 G7 の 80 コア/160 スレッドまでの拡張性、HDD と HP PCIe IO アクセラレータとの性能比較、Red Hat Enterprise Linux 6（RHEL6）の拡張性の測定を目的としています。

検証環境

この検証では、オープンソースソフトウェアの OLTP ベンチマークツール「sysbench」を使用し、1 秒あたりのトランザクション数を測定しました。検証環境は、以下の通りです。

サーバー

サーバー	HP ProLiant DL980 G7
プロセッサー	インテル® Xeon® プロセッサー E7- 8870（動作周波数 2.4GHz）
	ソケット数 8 ソケット（80 コア） メモリ 128GB
OS	Red Hat Enterprise Linux Server release 6.1
データベース	MySQL-server-5.5.17-1.el6.x86_64
ストレージ	HDD 146GBx2 RAID1、146GBx6 RAID5
	HP PCIe IO アクセラレータ

クライアント

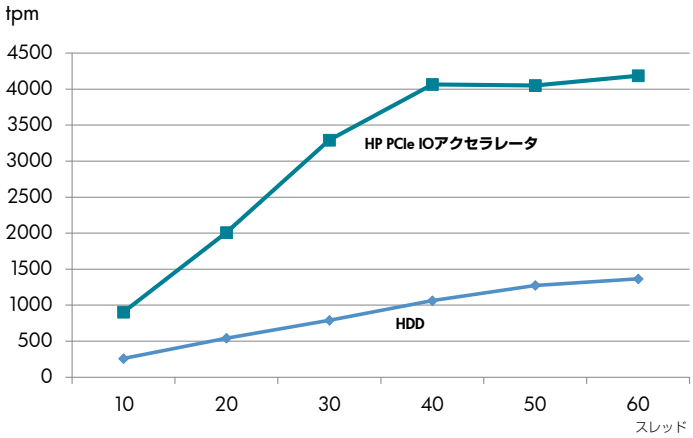
サーバー	HP ProLiant BL680c G7
プロセッサー	インテル® Xeon® プロセッサー X6550（動作周波数 2.0GHz）
	ソケット数 4 ソケット（32 コア） メモリ 128GB
OS	Red Hat Enterprise Linux Server release 6.1
データベース	MySQL-client-5.5.17-1.el6.x86_64
ストレージ	HDD 146GBx2 RAID1

HDDとHP PCIe IOアクセラレータとの性能比較

まず、HDD と HP PCIe IO アクセラレータとの性能比較を実施しました。SELECT/INSERT/UPDATE の処理によるスケールアップ性能比較です。40 スレッドの負荷を与えた時には HDD と比較すると約 4 倍の性能差という結果となりましたが、50 スレッド以降は性能向上が見られませんでした（図 6）。ここから、HP PCIe IO アクセラレータのスケールアップ性能の実証と、何らかの原因でスケールアップ性能が向上していないことが分かります。

インテル® Xeon®
プロセッサー E7 ファミリー

図 6: OLTP ベンチマーク性能（スレッド数増加による性能）



非 **NUMA** 対応のアプリケーション

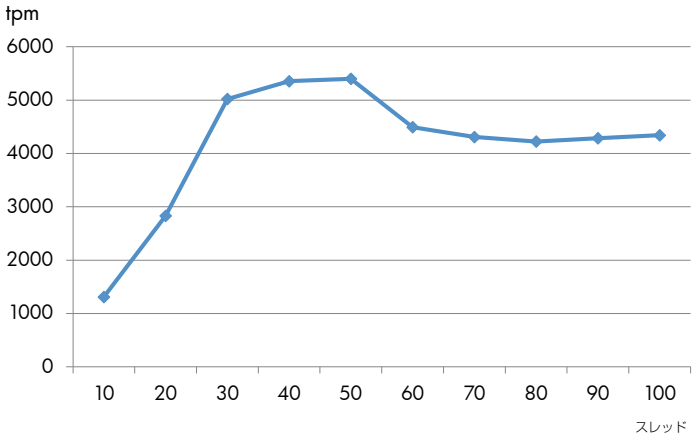
MySQL 5.5 はマルチプロセッサーに対応しているアプリケーションですが、対称型マルチプロセッシング（SMP）を前提としております。NUMA の命令や libnuma ライブラリーなどの NUMA 処理を支援するライブラリーは使われていない、つまり NUMA 環境を考慮されていないアプリケーションです。

インテル® Xeon® プロセッサー E7 はメモリコントローラを実装しているため、マルチプロセッサー構成ではローカルメモリとリモートメモリのレイテンシに最大で 3 倍程度の違いがあります。そのため、NUMA 対応の OS はローカルメモリを優先活用すれば高いメモリアクセス性能を発揮することができます。

この場合、Linux カーネル側である程度同じ NUMA ノードのメモリを使おうと努力します。しかしながら、NUMA ノードを跨ぐメモリアクセスの場合、アクセスする場合の効率が悪いために極力 NUMA ノードを跨がない CPU に処理が割り振られてしまいます。そのため、50 スレッド以降は性能が向上していません。

HP PCIe IO アクセラレータにより Read の I/O ボトルネックを解消させた状態で、SELECT の処理のみ行ったベンチマークを実施したところ、同様に 40 スレッドから 50 スレッドでスケールアップ性能が発揮できなくなりました（図 7）。

図 7: OLTP ベンチマーク性能/SELECT のみ（スレッド数増加による性能）



アプリケーションのボトルネック解消

これまでのベンチマークの結果より、MySQL 5.5 の場合は 40 スレッド～50 スレッドが分割点と判断します。そこで、Red Hat Enterprise Linux 6 からの新機能の cgroup により、2 つのインスタンスをそれぞれに HP ProLiant DL980 G7 上で 40 コアの 2 つでグルーピングさせ、リソースの利用権を 2 つに分割させます。

cgroups の定義に使った設定ファイル

今回、cpu0-39（HP ProLiant DL980 G7 の物理上の上段にあたる 40 コア）を mysql1 に、cpu40-79（同じく下段の 40 コア）を mysql2 に割り当てました（図 8）。

cgroups の定義に使った設定ファイルは次のとおりです。

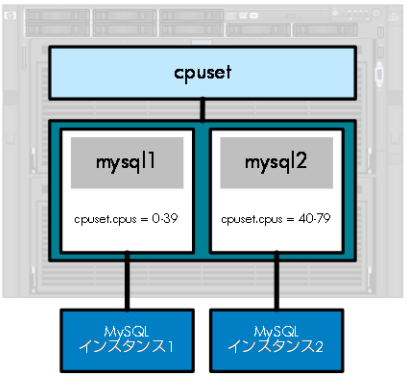
設定ファイル file:/etc/cgconfig.conf

```
mount {
        cpuset  = /cgroup/cpuset;
        cpu     = /cgroup/cpu;
        cpuacct = /cgroup/cpuacct;
        memory  = /cgroup/memory;
        devices = /cgroup/devices;
        freezer = /cgroup/freezer;
        net_cls = /cgroup/net_cls;
        blkio   = /cgroup/blkio;
}

group mysql1 {
        perm {
                task {
                        uid = root;
                        gid = root;
                }
```

インテル® Xeon®
プロセッサー E7 ファミリー

```
                admin {
                        uid = root;
                        gid = root;
                }
        }
        cpuset {
                cpuset.cpus = 0-39;
                cpuset.mems = 0;
        }
}

group mysql2 {
        perm {
                task {
                        uid = root;
                        gid = root;
                }
                admin {
                        uid = root;
                        gid = root;
                }
        }
        cpuset {
                cpuset.cpus = 40-79;
                cpuset.mems = 0;
        }
}
```

図 8: cgroup によるリソースの利用権の分割



すでに実行している MySQL のインスタンスを cgroup の中に参加させるには、各インスタンスのプロセス ID を調べた後に、cgclassify コマンドで任意の cgroup の中に参加させます。

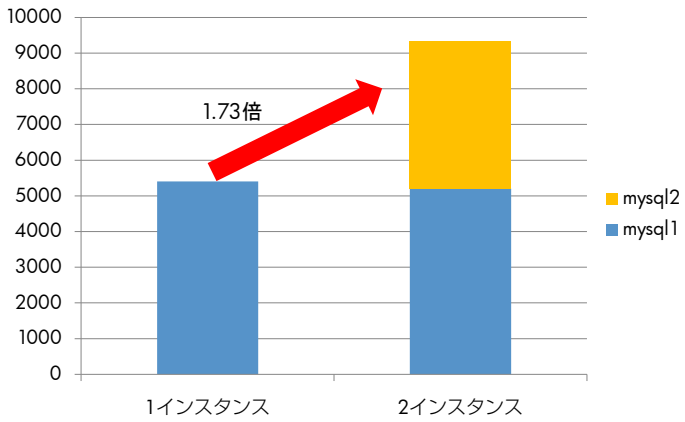
```
# cgclassify -g cpuset:mysql1 <mysql1インスタンスのmysqldのプロセスID>
# cgclassify -g cpuset:mysql2 <mysql2インスタンスのmysqldのプロセスID>
```

ボトルネック解消後のベンチマーク結果

このように複数インスタンスで起動した MySQL 5.5 を各インスタンスが相互に干渉しないように cgroup 内で動作させることで、全体性能としてスケールアップさせることができます。

2 つのインスタンスに対してロードジェネレーター側の sysbench も 2 つ同時のプロセスを起動し、ベンチマークを行なった結果が次のグラフになります。MySQL 5.5 のように非 NUMA 対応のアプリケーションであっても、cgroup により適切にグルーピングすることで、1.73 倍のスケールアップ性能を引き出す結果となりました（図 9）。

図 9: OLTP ベンチマーク性能（マルチインスタンスによる性能向上）



参考資料

インテル® Xeon®
プロセッサー E7 ファミリー

最後に、マルチプロセッサー環境の高速化に関連するその他のリソースを紹介します。

HP ProLiant DL980 G7 データシート
http://h20195.www2.hp.com/v2/GetPDF.aspx/4AA1-5671JPN.pdf

Red Hat Enterprise Linux と **HP ProLiant DL980 G7** が実現するコスト、パフォーマンス、アドバンテージ
http://www.jp.redhat.com/promo/WP/rh_hp_0726_DL980-RHEL6_WP.pdf

製品およびキャンペーンに関するお問い合わせ

カスタマー・インフォメーションセンター

03-6416-6512

月～金 9:00～19:00　土 10:00～17:00
（日、祝祭日、年末年始および 5/1 を除く）
www.hp.com/jp/proliant

日本ヒューレット・パッカード株式会社
〒136-8711 東京都江東区大島二丁目 2 番 1 号

redhat.