

C91 Intel XeonとCore i7の違いについて

目次

第一章 本書の概要

- (ア) はじめに
- (イ) 比較早見表

第二章 メモリ編

- (ア) メモリ転送速度(連続アクセス)
- (イ) メモリレイテンシ(ランダムアクセス)
- (ウ) 測定方法

第三章 コア編

- (ア) 1スレッドあたりの性能
- (イ) 並列度

C91 Intel XeonとCore i7の違いについて

第一章 本書の概要

(ア) はじめに

いわゆるサーバ用プロセッサである Xeon と、いわゆるデスクトップ用プロセッサである Core i7 の違いを**性能の観点**からまとめました。性能とは例えば速さなどの数値のことで、機能(できること)の違いについては立ち入りません。

Xeon の方がなんとなくよさそうだけどよく分からない人、具体的には以下のような人の助けになることを期待しています。

- Xeon と Core i7 のどちらを買うべきか分からない
- Xeon より Core i7 の方が速いのでは??と経験から薄々思っている

また本書は**基礎的な知識を普及**し皆様にコンピュータを楽しく正しく使ってもらうことを目的としています。従ってこれを読めば即目の前のプログラムを高速化できる!といった内容にはなっていません。むしろ、これを読んで様々な性能の観点を頭に入れた状態で課題に取り組んだときに、ふと以前とは違った発想で何か考えることができれば、と思っています。

(イ) 比較早見表

まずは簡単に比較のみ示します。各項目は二章以降で原理も含め詳しく説明します。

注目すべき点としては、**全ての項目について一概にサーバ用の方が高性能ではない**という点です。従って実行したいタスクや使い方に合わせて適切な方を選択する必要があります。

| 性能要因 | 比較 |
|--------------------|----------------------|
| メモリ転送速度(連続アクセス) | サーバ用の方が2倍高い |
| メモリレイテンシ(ランダムアクセス) | デスクトップ用の方が10~15ns 速い |
| 1スレッドあたりの性能 | デスクトップ用の方が高い |
| 並列度 | サーバ用の方がコア数/スレッド数が多い |

C91 Intel XeonとCore i7の違いについて

第二章 メモリ編

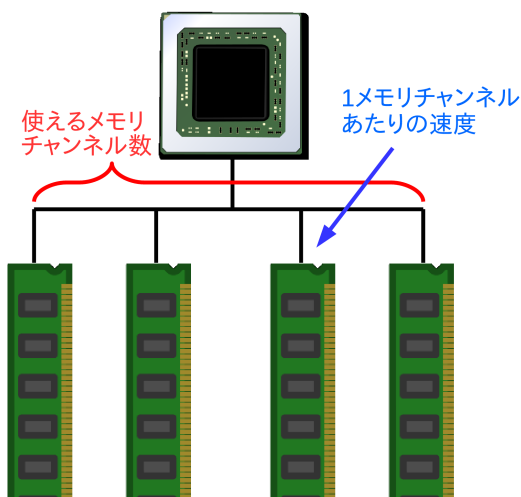
(ア) メモリ転送速度(連続アクセス)

メモリ転送速度とはメモリに連続(シーケンシャル)にアクセスする場合の最大スループットです。ランダムアクセスする場合の遅延とは全く別の話で、これについては次の章で扱います。

メモリ転送速度は、

1 メモリチャンネルあたりの速度 × 使えるメモリチャンネル数

で決まります。大まかな数値はサーバ用で **40GB/s** から **80GB/s**、デスクトップ用で **20GB/s** から **40GB/s** 程度であり、サーバ用のほうが2倍速くなっています。



1メモリチャンネルあたりの速度は、使うメモリモジュールによって決まります。例えばPC3-12800の12800が転送速度(MB/s)を表し、このメモリモジュールは最大12.8GB/sで転送できることが分かります。PCX-YYYYという表記とDDRX-ZZZZという表記は相互に互換で、変換方法については別途グーグル先生に聞いてください。

使えるメモリチャンネル数は、複数のメモリモジュールに何並列で読み書きできるか

C91 Intel XeonとCore i7の違いについて

を表します。これはCPU内部のメモリコントローラのチャンネル数で決まり、一概にサーバ用/デスクトップ用で決まるものではありません。ただし、一般に2016年現在ではサーバ用が4チャンネル、デスクトップ用が2チャンネルのものが多くあります。下位モデルではこれより少なかったり、最上位モデルではデスクトップ用でも4チャンネルありたりします。例えばXeon E3では2チャンネル、モバイル向けのAtom等では1チャンネル、Core i7-5960X Extreme Editionでは4チャンネルなどです。

CPUの型番と使えるメモリモジュール、メモリチャンネル数の対応はIntel公式のARKというページで調べることができます。例えばXeon E5-2699 v4のスペックは<http://ark.intel.com/ja/products/91317/>にあり、以下のようになっています(スクリーンショットは前述のURLの2016年12月24日時点の状態を引用)。この図では、使えるメモリの種類が最高でDDR4 2400(PC4-19200と同値)、使えるメモリチャンネル数が4なので $19.2\text{GB/s} \times 4 = 76.8\text{GB/s}$ がこのCPUの最大メモリ転送速度となります(転送速度と帯域幅は同じ意味)。

| 仕様 | |
|------------------------|--|
| 基本仕様 | |
| プロセッサ・ナンバー | E5-2699V4 |
| ステータス | Launched |
| 発売日 | Q1'16 |
| リソグラフィ | 14 nm |
| 希望カスタマー価格 | \$4115.00 |
| パフォーマンス | |
| 補足情報 | |
| メモリの仕様 | |
| 最大メモリーサイズ (メモリーの種類に依存) | 1.54 TB |
| メモリーの種類 | DDR4 1600/1866/2133/2400 |
| 最大メモリーチャンネル数 | 4 |
| 最大メモリー帯域幅 | 76.8 GB/s |
| 物理アドレス拡張 | 46-bit |
| ECC メモリー対応* | <input checked="" type="checkbox"/> はい |

注意として、使えるメモリチャンネル数はマザーボードにあるメモリスロットの数と

C91 Intel XeonとCore i7の違いについて

同じではありません。メモリスロットが4本や8本あっても、CPUまたはマザーボードのいずれかが2チャンネルしか持っていなければ使えるメモリチャンネル数は2となります。2016年12月現在、一般向けに販売されているマザーボードではX99チップセットを搭載したもの以外4チャンネル対応のものはありません。

結論として、XeonとCore i7のメモリ転送速度はXeonの方が約2倍速く、その理由は使えるメモリチャンネル数がXeonでは4、Core i7では2でXeonの方が2倍多いからと言えます。

(イ) メモリレイテンシ(ランダムアクセス)

メモリレイテンシとはメモリにランダムアクセスした場合のアクセス遅延のことです。(ア)では連続アクセスした場合の帯域(スループット)を扱いましたが、ここでは遅延を扱います。

メモリレイテンシは、

CPUがメモリアクセスを処理する時間 + メモリがメモリアクセスを処理する時間

で決まります。具体的には、アイドル状態の場合でデスクトップ用で約50ns～、サーバ用で65ns～程度です。注意としてはこれは何も無い状態からあるアドレスにランダムアクセスした場合にデータが返ってくるまでの時間であり、全てのアクセスにこの時間がかかるわけではないことです。つまりその次のアドレスに続けてアクセスする際にはキャッシュやプリフェッチが有効に働くので、遅延はもっと短くなります。

(i) CPUがメモリアクセスを処理する時間

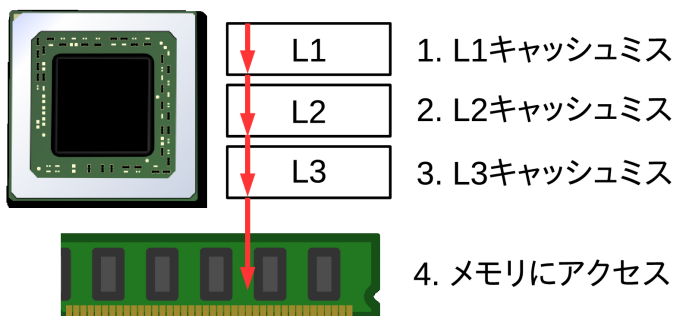
「CPUがメモリアクセスを処理する時間」とは、CPUキャッシュをミスするのにかかる時間で、これがデスクトップ用CPUのほう10nsから15ns程度短いことで上記のサーバ用とデスクトップ用のメモリレイテンシの差が現れています。

ではキャッシュ「ミス」にかかる時間とは何でしょうか。キャッシュとはCPU内にもうけられた高速なメモリのようなもので、CPUがメモリにアクセスする前にまずキャッシュに欲しいデータがあるかどうか確認します。キャッシュにはL1, L2, L3と階層があり、数字が大きくなるに従って容量が大きい代わりに少しずつ遅くなっていきます。従って、メモリアクセスを要求された時のCPUの動作としては以下ようになります。

C91 Intel XeonとCore i7の違いについて

1. L1 キャッシュを見る。目的のデータがあればラッキー
2. L1 キャッシュに目的のデータがなければ、L2 キャッシュを見る。あればラッキー
3. L2 キャッシュにもなければ、L3 キャッシュを見る。あればラッキー
4. L3 キャッシュにもなければ、仕方がないのでメモリを読みに行く

* L1, L2, L3は実際にはCPUに内蔵



つまり、メモリに実際にアクセスする前に、キャッシュに目的のデータがあるかどうかをL1, L2, L3と順に調べていくのに時間がかかるというわけです。さてこの時間は前述のようにデスクトップ用CPUの方が速くなっています。その理由はデスクトップ用CPUの方がサーバ用CPUよりもキャッシュの動作周波数が速いからです。動作周波数の違いについてはコア編で説明しますが、一般にデスクトップ用CPUの方が高く設定されています(ここでの話はキャッシュの動作周波数、コア編での話は演算器の動作周波数の話ですが、ほぼ同じことが成り立ちます)。

ここであるタスクにかかる「サイクル数」という概念を導入すると話が分かりやすくなります。あるタスクにXサイクルかかるとは、クロックがX回まわるとその仕事が終わることを意味します。従って同一のタスクを終えるのにかかる「実」時間は1秒あたりのサイクル数が大きい方が短くなります。

L1からL3までキャッシュを終えるのにかかるサイクル数は、CPUの世代が進むと増えたり減ったりしますが(一方的に減るわけではない)、同じ世代のCPUならばサーバ用もデスクトップ用も同じです(具体的な値はIntelの数千ページあるマニュアルを見れば書いてあります)。一方、「1秒あたりにまわるサイクル数」、すなわちキャッシュの動作周波数はデスクトップ用で4GHz超、サーバ用だと3GHz以下程度です。これはコアの周波数より少し高くなっていますが、コアにTurbo Boost(余裕があるときに周波

C91 Intel XeonとCore i7の違いについて

数をあげる機能)がかかった時に調度良くなるように調整されているようです。従ってCPUがメモリアクセスを処理する「実」時間はクロックの高いデスクトップ用CPUの方が短いと言えます。

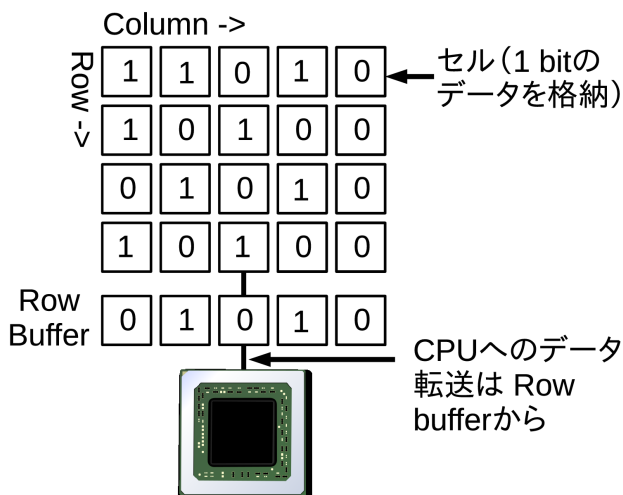
(ii) メモリモジュールがメモリアクセスを処理する時間

CPUからアクセス要求を受けた後にメモリモジュールがメモリアクセスを処理する時間は、どのメモリモジュール(DDR3 1033, DDR4 2133など)を使うかによって決まり、CPU自体は基本的に関係ありません。そこで以下ではCPUには直接関係ありませんがメモリアクセスの遅延も一様ではなく色々考えるべきことがあるんだという話を記します。

メモリからCPUにデータを転送するには、以下の2つのステップが必要です。

1. Precharge(メモリへの電荷のチャージ)
2. Row(行)アクセス
3. Column(列)アクセス

ここではRowとColumnについて説明します。Prechargeはさらに細かいメモリ内の仕組み(Bank, Rankなど)が必要なるため本書では触れません。



Row(行)アクセス

メモリの中では図のようにメモリセル(小さなコンデンサ)がRowとColumnに並んでおり、またRowを一行分格納できるRow Bufferというものがあります。CPUにデータを転送できるのはRow Bufferからのみのため、まず目的のデータが入っている

C91 Intel Xeonと Core i7の違いについて

Row を Row Buffer に持つてくる必要があります。これを Row アクセスといい、これにかかる遅延が約 13ns と仕様で定められています。

Column(列)アクセス

目的の Row を Row Buffer に持ってきたら、次は目的の Column を指定します。Column を指定してから実際にデータが出てくるまでの遅延がいわゆる CAS レイテンシというもので、モデルによって違いますが 10ns-20ns 程度です。なお、よく CL=12 などといった表示がありますが、これは 12ns の意味ではなくて 12「サイクル」、すなわちクロックが 12 進んだらアクセスが完了するという意味です。メモリの動作周波数は DDRX-YYYY の Y の半分なので、DDR4 2133 ならば 1066MHz で 12 サイクルだと約 11ns になります。この場合は 12 クロックと 11ns がほぼ同じで勘違いしがちですが、最近出てきつつある DDR4-4200 などではクロックが 2.1GHz なので実時間は CL の半分の値になります。

Row Buffer ヒット

上記の CAS レイテンシは、簡略化して「メモリのレイテンシである」と言われることがよくありますがこれは一概には正しくありません。Row アクセスでは目的の Row を Row Buffer に持つてきましたが、連続した Column へのアクセスですでに Row Buffer に目的の Row が存在することになります。これを Row Buffer ヒットと呼び、この場合 Row アクセスの時間はかからないので CAS レイテンシ = メモリレイテンシで正しいです。しかし、Row Buffer がヒットしなかった場合 (Row Buffer ミス) では Row アクセスをする必要がありますその時間は無視できないほど大きいので、メモリレイテンシは CAS レイテンシとは大きく異なります。Row Buffer ヒット率を考慮していかにプログラムを上手く作るか・実行するかは進行中の研究課題です。

(ウ) 計測方法

(ア)、(イ)で示したメモリ転送速度、メモリレイテンシは Intel の出している Memory Latency Checker (mlc)というツールで簡単に測定することができます。ダウンロードは <https://software.intel.com/en-us/articles/intelr-memory-latency-checker> からできます。

mlc を実行すると次ページの図のようになり、枠で囲った部分がそれぞれメモリレイテンシとメモリ転送速度を現します。計測は Intel Core i5 6400, DDR4 2133 (PC4-1700), 2 メモリチャンネルで行いました。転送速度の理論値は $17.0\text{GB/s} \times 2 =$

C91 Intel XeonとCore i7の違いについて

34.0GB/sです。レイテンシは高速ですが転送速度がやや低いのはデスクトップとして利用していて同時に他のソフトが動いているからかもしれません。

```
Intel(R) Memory Latency Checker - v3.1a
Measuring idle latencies (in ns)...
      Memory node
Socket      0
      0      55.7

Measuring Peak Memory Bandwidths for the system
Bandwidths are in MB/sec (1 MB/sec = 1,000,000 Bytes/sec)
Using all the threads from each core if Hyper-threading is enabled
Using traffic with the following read-write ratios
ALL Reads      :      30429.7
3:1 Reads-Writes :      28360.0
2:1 Reads-Writes :      28085.8
1:1 Reads-Writes :      32351.7
Stream-triad like:      28155.1
```

転送速度(Peak Memory Bandwidth)を測定しているところの ALL Readsなどは、測定に用いるメモリアクセスパターンを現します。ALL Readsは読み込みのみ、N:M Reads-Writesは読み込みと書き込みをN:Mにした場合、Stream-triad likeはスーパーコンピュータなどの性能を測るのに用いられる stream triadというベンチマークを模したもので、 $a[i] = b[i] + \alpha c[i]$ という計算を用います。

C91 Intel XeonとCore i7の違いについて

第三章 コア編

(ア) 1スレッドあたり性能

実行したいプログラムが並列化できなくて1スレッドしか有効に使えない時、1スレッドあたりの性能が全体の性能に直結します。また理論的には並列化できる場合でも、スレッド間の同期やデータ交換が頻繁に発生する場合、対象のタスクが小さすぎる場合などにも少数のスレッドのみ使って実行した方が多くのスレッドを使うよりも高速になることが多々あります。

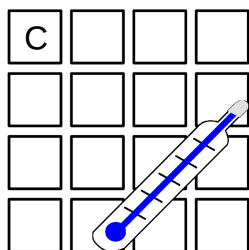
デスクトップ用CPUは、サーバ用CPUに比べて以下の特性があります。

1. 周波数が高い
2. コアの世代が新しい
3. キャッシュサイズが小さい

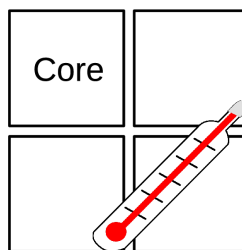
(i) 周波数

周波数がデスクトップ用CPUの方が高い理由は、コア数が少なく全体の発熱が小さいからです。近年の計算機はCPU・GPUのような単体デバイスからデータセンター全体に至るまで電力と発熱が一番大きな課題になっています。よって多くのタスクを並列実行するためにコア数が多く設定されているサーバ用CPUでは周波数があまり上げられません。従って、単純な演算(例えばレジスタに入った二つの値を足す)ではデスクトップ用CPUの方が高速であるといえます。

サーバ用CPU:
低周波数(== 低発熱)な
コアを多数搭載



デスクトップ用CPU:
高周波数(== 高発熱)な
コアを少数搭載



(ii) コアの世代

次にコアの世代ですが、製造の歩留まり向上やバグだしを目的としてまずローエンドな製品から新しいコアが導入されていくという戦略になっています。具体的には下の表

C91 Intel XeonとCore i7の違いについて

のように、同時期で比較するとデスクトップ用 CPUの方が1から2世代ほどコアの世代が新しいことが分かります(QNは第N四半期の意味)。

| コアの世代 | サーバ用(E7) | デスクトップ用(i7) | モバイル用(i7-U) |
|-----------------------|------------------|-------------|-------------|
| Broadwell(5世代) | 2016 Q2 発売 | 2015 Q2 発売 | 2015 Q1 発売 |
| Skylake(6世代) | 現在 E3(ローエンド)のみ発売 | 2015 Q3 発売 | 2015 Q3 発売 |
| Kabylake(7世代) | 未定 | 2017 Q1 発売? | 2016 Q3 発売 |

コアの世代が新しくなると、1クロックあたりに実行できる命令の数(IPS: Instructions Per Cycle)が上昇します。CPU内部では命令を1つずつ実行するわけではなく、もっと小さな命令(micro operations)に分割し並び替えたりパイプラインで実行したりと非常に複雑になっており、その実行効率がコアの世代が進むと向上するためです。

分かりやすい例としては分岐予測があります。CPUは条件付 jmp 命令(あるレジスタが0なら飛ぶ、など)があるとどちらに飛ぶかをあらかじめ予測して飛び先の命令を実際に飛ぶ前に実行開始します(これを投機的実行といいます)。予測がはずれると投機的実行は無駄になってしまうため、**コアの世代が進んで分岐予測の精度が上がるとIPSが上がる**こととなります。例えば2017年1月に発売予定である(Intelではありませんが)AMDのRyzenでは、分岐予測にディープラーニングで学習した結果を用いると話題になっています。

(iii) キャッシュサイズ

キャッシュとは、CPUについているメモリのデータを一時的に保存しておくための高速ストレージのことです。(今回は扱いませんが)メモリはCPUに比べて非常に遅いため、データが必要になるたびにいちいちメモリにアクセスしてはCPUの性能がまったく使いきれません。そこでよく使うデータをキャッシュに入れておいて高速にアクセスできるようにします。

具体的にあるプログラムが与えられた時にどのくらいキャッシュサイズがあれば十分か(== それ以上キャッシュサイズを増やしても性能が上がらないか)、あるいはキャッシュサイズをX KB減らすとどれくらい性能が下がるかは単純には分からず、研究の課題となっています。しかしとにかく言えることは、**一般にキャッシュのサイズが大きいほ**

C91 Intel XeonとCore i7の違いについて

ど今欲しいデータがキャッシュに乗っている確率が高くなるのでプログラムの性能が上がるということです(キャッシュサイズを小さくして損することはない)。

最近のIntelのCPUではキャッシュはL1, L2, L3と3レベルになっています。これはメモリとキャッシュの関係と同じく、高速な代わりに容量が小さいキャッシュと比較的低速な代わりに容量が大きいキャッシュの階層構造になっています(L1が最速で最低容量、L3が最も遅く最高容量)。サーバ用CPUとデスクトップ用CPUでは、L1, L2のサイズは同じですがL3のサイズが大きくなっています。具体的にはサーバ用では**10MBから多いもので30MB、デスクトップ用では数MB程度**です。

以上(i), (ii), (iii)をまとめると、1スレッドあたり性能ではデスクトップ用CPUの方が演算自体は高速、ただしメモリアクセスが多い場合にはキャッシュの多いサーバ用が有利な場合がある、と言えます。例えば並列化できずメモリアクセスもほとんどしないsuper PIのような純粋演算系のベンチマークではデスクトップ用の方が高い性能を記録するでしょう。

(イ) 並列度

(ア)では1スレッドあたりの性能を見たので、次にそのスレッドを並列に並べる場合を考えます。そのためにはCPUのコア数が多いほうが有利ですが、一般にコア数はサーバ用CPUの方がデスクトップ用CPUよりもかなり多く設定されています。2016年12月現在のサーバ用では最大24コアのものがあるのに対し、デスクトップ用では8コアが最大となります。(なおここでは物理コアのみ考え、Hyper Threadingによる論理コアは考えません。Hyper Threadingは扱いが難しく分析もしづらいため、性能測定などの際にはオフにすることをおすすめします。)

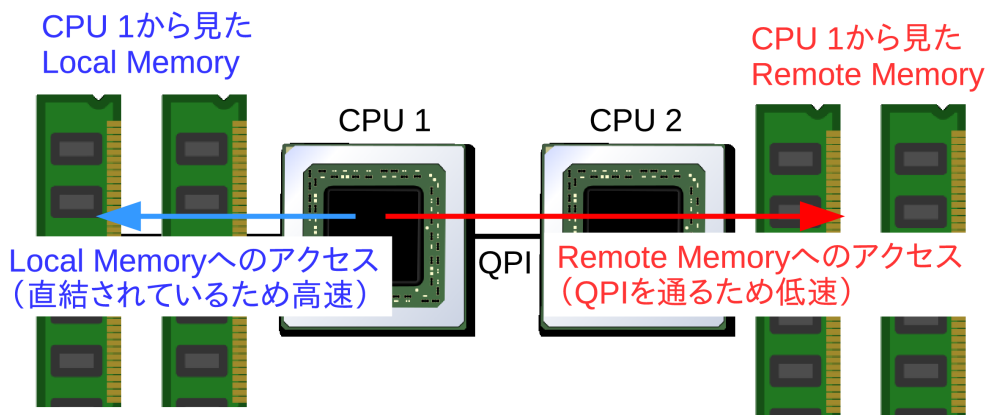
サーバ用CPUでは1CPUあたり最大24コアですが、これをさらに1台のマシンに複数搭載することができます。具体的にはXeon E5では4CPU、Xeon E7では8CPUまで1台のマシンに搭載できます。Xeon E7-8890 v4は24コアなのでこれを8個積みれば192コアになります。

1台のマシンにCPUを複数搭載した状態はNUMA (Non-Uniform Memory Access) と呼ばれ、次の二つの特性があります。

1. アプリケーションからは普通の1CPUマシンと同じに見える
2. ハード的にはCPU間の通信に時間がかかるため、性能面では同じではない

C91 Intel XeonとCore i7の違いについて

まずアプリケーションから見た場合ですが、1 CPUで192コアある場合と全く同じに使えます。どちらかのCPUでデータを更新したら、それがソフトウェアからは見えないレイヤーで自動的にもう片方のCPUにも通知されます(これをキャッシュコヒーレンシといいます)。つまり、通信などしなくても普通にスレッドを192個立てて共有メモリでデータを交換できます。また既存のマルチスレッドライブラリやランタイムも何の改変もせずに使えます(本書を読んだ方には分かっていただけだと思いますが、192倍速くなかどうかはまた別の話です)。



Non-Uniform Memory Access (NUMA)

次に性能面ですが、メモリアクセスの速さ(遅延・帯域ともに)が1 CPUの場合とは変わってきます。具体的には、例えばCPUが二つ(CPU 1, CPU 2)でメモリモジュールが4枚刺さっている場合を考えます(上図)。このとき2枚がCPU 1に直結されており、残り2枚がCPU 2に直結されています。あるCPUから見て、自分に直結されたメモリをLocal Memory、隣のCPUの直結されたCPUをRemote Memoryなどといいます。Remote Memoryにアクセスするためには、QPIというインターコネクと隣のCPUソケットを通じてアクセスする必要があります。この影響により、Local MemoryとRemote Memoryではメモリアクセス遅延が数十ns違い、帯域も倍程度異なります(QPIは2つのCPU間でL3キャッシュを同期するのにも使われるため、帯域はQPIの混み具合によっても変化します)。これがNon-Uniform Memory Accessという名前の由来です。メモリ編で見たようにメモリ帯域はサーバ用CPUがデスクトップ用CPUの約2倍、メモリレイテンシはサーバ用CPUの方が不利であったため、Remote Memoryにアクセスするとサーバ用CPUでもデスクトップ用CPUと同じかより遅くなってしまうことが分かりま

C91 Intel Xeon と Core i7 の違いについて

す。

通常は OS が両方のメモリを均等に使うように割り当てているため、性能的には Local Memory と Remote Memory の中間くらいの値になります(これを address interleaving といいます)。しかし、パフォーマンスを追及するのであればなるべく **Local Memory** にアクセスしたほうがよいということになります。これは Linux であれば numactl (NUMA ConTroL) コマンドなどで制御できます(コマンドの使い方などは本書の範囲外なので触れません)。

以上をまとめると、サーバ用の方がコア数が多いため並列化が有効なプログラムには有利、ただし複数 CPU を同時に搭載する NUMA 構成の場合はメモリアクセスの速度低下に注意すべき、といえます。

C91 Intel XeonとCore i7の違いについて

- ・本書の図は Openclipart (<https://openclipart.org/>) より引用しました。
- ・本書のカラー版 pdf を <http://www.soramichi.jp/pdf/C91.pdf> に用意しておりますので是非ご利用ください。アクセス制限/配布制限等ありません。
- ・本書はコミックマーケット C91 において無料頒布したものです(委託先:2日目東ト29a あいすまぐねっと)。