計算器アーキテクチャ

山本 考伸

平成 8 年 5 月 1 日

1　マルチプロセッサ概論

A　マルチプロセッサの分類

- プロセッサアーキテクチャ
  プロセッサのアーキテクチャとしての分類
  - スーパースカラ方式
  - VLIW 方式
  - スーパーパイプライン方式
  - マルチスレッド方式
- メモリアーキテクチャ
  メモリが共有か非共有か, 集中か分散かで分類される
- ネットワークアーキテクチャ

B　マルチプロセッサの基本方式

- 粒度問題
  プログラム分割の大きさにおける問題
- プログラム分割
  - メモリ共有方式
    命令形のみを分割する
  - メッセージ交換方式
    命令形とデータ全てを分割する方式

2　メモリ共有型マルチプロセッサ

最新のプロセッサは、キャッシュを内蔵している. このキャッシュを生かして、 (1) データの局所せいを生かすこと、 (2) 無駄な通信を極力削除することができる。

A　共有メモリの構成方式

- 集中共有メモリ
- 集中共有メモリ　(ローカルメモリ付)
- 集中共有メモリ　(キャッシュ付)
- 分散共有メモリ
- 分散共有メモリ　(キャッシュ付)

共有メモリの分類方式としては、次のようなものがある。

- UMA モデル
  共有メモリが、全てのプロセッサから等距離
- NUMA モデル
  共有メモリが、全てのプロセッサから等距離ではない

B　メモリコンシステンシモデル

メモリが、いつの時点で正しい状態を示しているのか、いつ正しい状態にすべきか, がメモリコンシステンシ問題である。
(1) 書き込みおいてけぼり制御、 (2) キャッシュメモリの更新などのとき, どのように処理すべきか,

- ノンコンシステンシモデル
- 逐次コンシステンシモデル
  (1) 各プロセッサの命令は、並べられた順に行う
  (2) 各々のプロセッサの先行命令でのメモリ操作が完了しないと、次の命令に移らない。
  実際は、臨海境域へのアクセスや、異なる領域へのアクセスにおいては、書き込み操作完了を待たない。この 2 つの方式を達成するモデルとして、ウィークコンシステンシモデル、リリースモデルも存在する。

3　キャッシュコヒーレンス

A　キャッシュコヒーレンスとメモリコンシステンシの関連

あるプロセッサにおいて、キャッシゅの変更があったとき, 同一ブロックをキャッシゅに保持する多のプロセッサのキヤッシゅを、どうするかには、(1) ノンコンシステンシ、 (2) 逐次コンシステンシ, (3) ウィーク ・リリースコンシステンシ, の 3 通りがある.
キャッシュコヒーレンスの諸方式として、次のようなものがある。

- ハードウェアによる方法
  (1) スヌープキャッシゅ方式
  (2) ディレクトリ方式

- ソフトウェアによる方法

B　スヌープキャッシュ法

- 単純な構成の場合
  各プロセッサにスヌープコントローラがあって、互いの書き込みアドレスを監視する。ストアスルーと、ストアインにわかれる。トラフィックが大きく高速処理困難。
- 各ブロックにタグをつける方式
  あるプロセッサで、書き込みが多数起こるブロックに、Dirty タグをつけ, システムないに、1 つしか存在しないようにする。ストアインの場合には、次の 4 通りにわかれる。
  - バークレイプロトコル (無効果、無書き戻し)
  - イリノイプロトコル (無効果、書き戻し)
  - ドラゴンプロトコル (更新、無書き戻し)
  - ファイアフライプロトコル (更新、書き戻し)

C　ディレクトリ方式

メモリないの各ブロックごとに、そのブロックがどのプロセッサにキャッシュされてるかを示すディレクトリが用意されている。

- フルマップ方式
  メモリブロックは、N ビットのディレクトリを持って、ビットが 1 であれば、対応するプロセッサのキャッシュにそのブロックが存在する。
- リミテッド方式
- チェイン方式

更新方式では、共有するプロセッサ数が増大するので、無効果方式と比べて、ポインタ数、チェイン数が増える
ブロックサイズが大きいと、ディレクトリの保持ビット数は減るが、関係ないデータが、同じブロックにはいる可能性もある。

D　ソフトウェアによる方式

D.1　ソフトウェア方式に必要なこと

- ストアスルー方式
- 同期命令
- キャッシュ内容の選択的無効化

D.2　ソフトウェアによる諸方式

- キャッシュブロックの無効化命令を持つ方式
- マーキング法
- キャッシュ制御命令を拡張したもの
  キャッシュブロックの、選択的無効化, 全面的無効化、などがある。

E　キャッシュによる通信量の削減法

データ属性に応じて、プロトコルを切り替えする。

F　キャッシュオンリメモリ

プログラムやデータがシステムないのどこかのキャッシュに、存在する。メモリは存在しない。

4　ＪＵＭＰ－１

A　ＪＵＭＰ－１の特徴

- キャッシュコヒーレンス機構を内蔵したメモリ共有方式
  スヌープ方式と、疑似フルマップ方式を採用して, データごとに, コヒーレンスプロトコルの変更を可能にしている。

- 通信のオーバヘッドを削減
  (1) レイテンシを演算時間とオーバーラップする
  (2) 各データに、同期ビットを付与したり、ＦＩＦＯなどを工夫して、無駄な通信を省いてる。

- 階層トーラス網
  相互通信網として階層トーラス網を採用。

- 高速な入出力機構

B　メモリアーキテクチャ

ＪＵＭＰ－１では, 演算処理を実行するプロセッサの他にメモリアクセス、通信を分担する知的なメモリ処理機構を導入して、細粒度並列処理を実現してる

(A) 疑似フルマップ方式

各クラスタが、2 進木構造でコヒーレンス制御される。各ディレクトリは、2 進木の階層に対応したエントリを持って、そのエントリが、１であるときはその 2 進木ノードの枝につながるクラスタに少なくとも１つのコピーが存在することを示す。

(B) リミテッド、チェイン方式、と比較した疑似フルマップ方式の評価

- ディレクトリ容量
  共有度が大きいと疑似フルマップが、優位である。

- ネットワークを流れるトラフィック量
  遠距離に少数の共有クラスタが存在すると、無駄な転送が多くなる。

C　高速同期、通信機構

C.1　キャッシュによって実現される機構

(1) プリフェッチと注入
(2) 更新データのマージ
(3) 早期共有解除

C.2　キャッシュコヒーレンスプロトコルの動的切り替え

データ属性による最適プロトコル選択は、コンパイル時に行う必要があるが、ＪＵＭＰ－１では、実行時にも動的に実行可能である。

C.3　１対１通信における高速化

同期ビットと、データを不可分なものにするこれを利用して、様々な通信モデルが高速化されている。

参考文献

[1] 富田 眞治. 並列計算機構成論. 昭晃堂, 1986.

[2] 福島さんのプリント