第３回　コンピュータ科学基礎論　資料　　2017-10-06 山内

今回は、MIMD（マルチプロセッサ）について、問題点を議論してみたい。

(1)　MIMD（マルチプロセッサ）が得意とすることは何だろうか？　苦手とすることは何だろうか？

(2)　ハードウェア構成として、共有メモリ型と分散メモリ型に分類されることがあるが、それぞれの良い点、問題となる点を比較してみよう。
（ア）処理性能の得やすさ　～　構成上何がボトルネックになるか
（イ）拡張性　（スケーラビリティ）
（ウ）プログラムの移行性、考えやすさ

(3)　分散メモリ型がスケーラビリティで優れているとして、「京」のような大型のスーパーコンピュータは分散メモリ型にしている　（正確には、メモリ共有型構成をしたノードを結合ネットワークで結合している）。 結合ネットワークの設計は、すべてのノード間を結合すれば（フルコネクション）１ホップで到達できるが結合本数が多くなり各ノードの出入りロインターフェースが増えたり配線スペースが多くなったり故障が増えたりする。逆に結合を少なくする代わりに直接到達できないノードへはマルチホップで行き着くようなネットワーク（非常に多数の結合法が提案されている）は、結合線の数が減る代わりに到達時間が長くなる。 新しいネットワークトポロジーの提案をしている資料（論文）を探して読んでみると面白い。 （たとえば、「スーパーコンピュータ「京」のインターコネクト Tofu 」http://www.fujitsu.com/downloads/JP/archive/imgjp/jmag/vol63-3/paper05.pdf や 「革新的な「６次元メッシュ/トーラス」ネットワーク技術」 http://www.fujitsu.com/jp/about/businesspolicy/tech/k/whatis/network/ ）

(4)　NUMA (Non-Uniform Memory Architecture、メモリアクセスが一様でない) 構成のシステムを使う時、ベストの性能を出すためには何を考える必要があるだろうか？
同様に、データ転送時間（通信時間）が相手によって異なるとき （たとえば遠いノードへのアクセスがマルチホップになっているので時間がかかる等） ベストの性能を出すためには何を考える必要があるだろうか？

(3)　直列プログラムを MIMD 型のコンピュータに移して並列化しようとする時、何が起こるか？　そもそも、直列プログラムを移すということができるだろうか、それとも新しく計算手順を考え直す必要があるだろうか？