

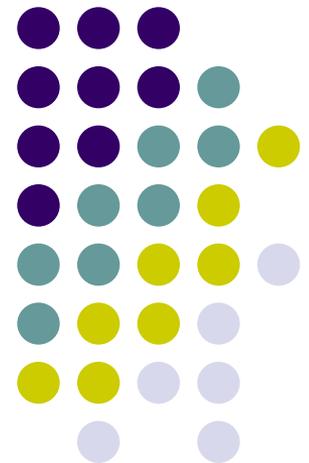
2013年度 実践的並列コンピューティング 第2回

並列アーキテクチャと
並列プログラムモデル

遠藤 敏夫

endo@is.titech.ac.jp

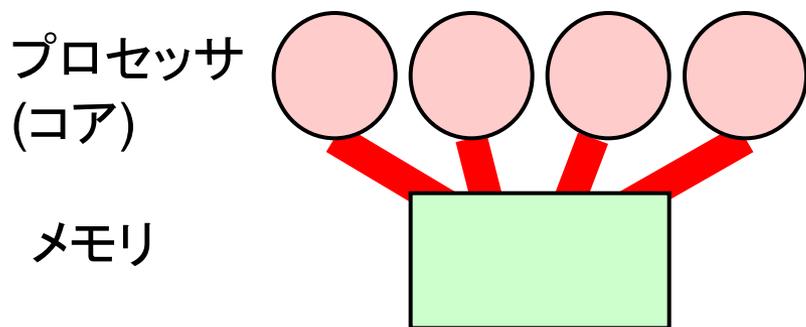
2013年4月15日



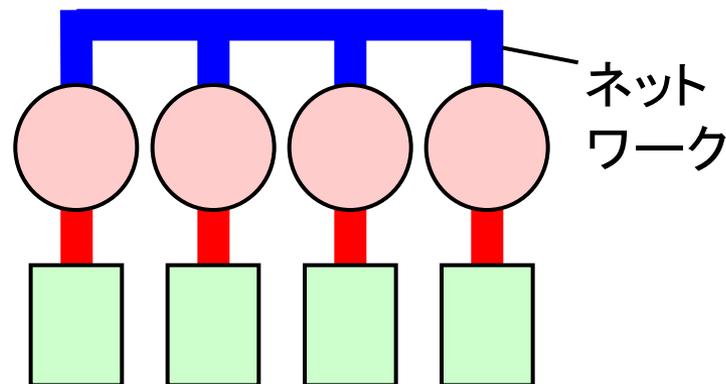
並列アーキテクチャの分類



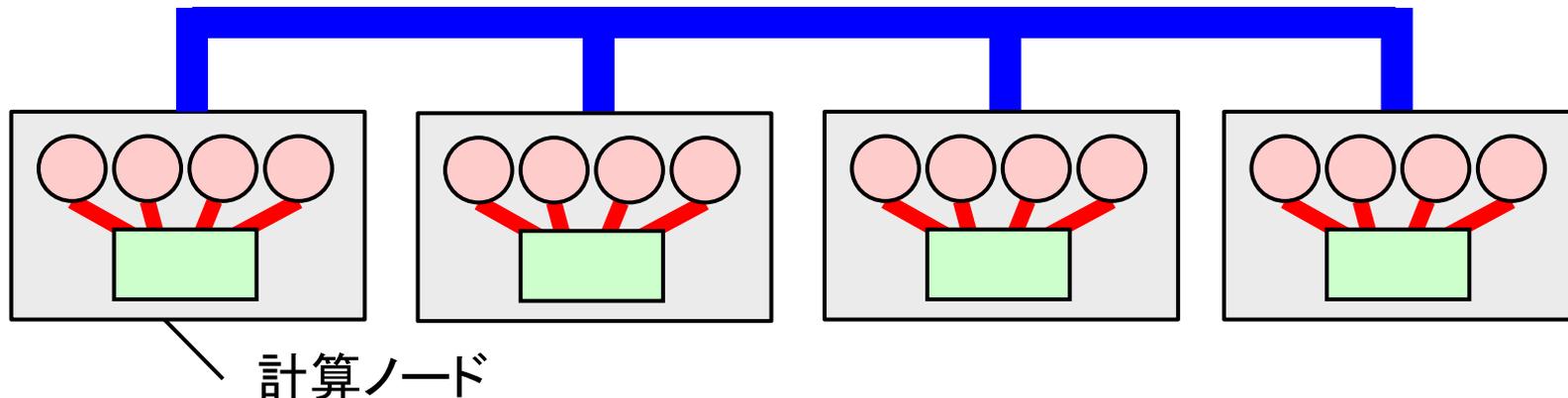
共有メモリアーキテクチャ



分散メモリアーキテクチャ



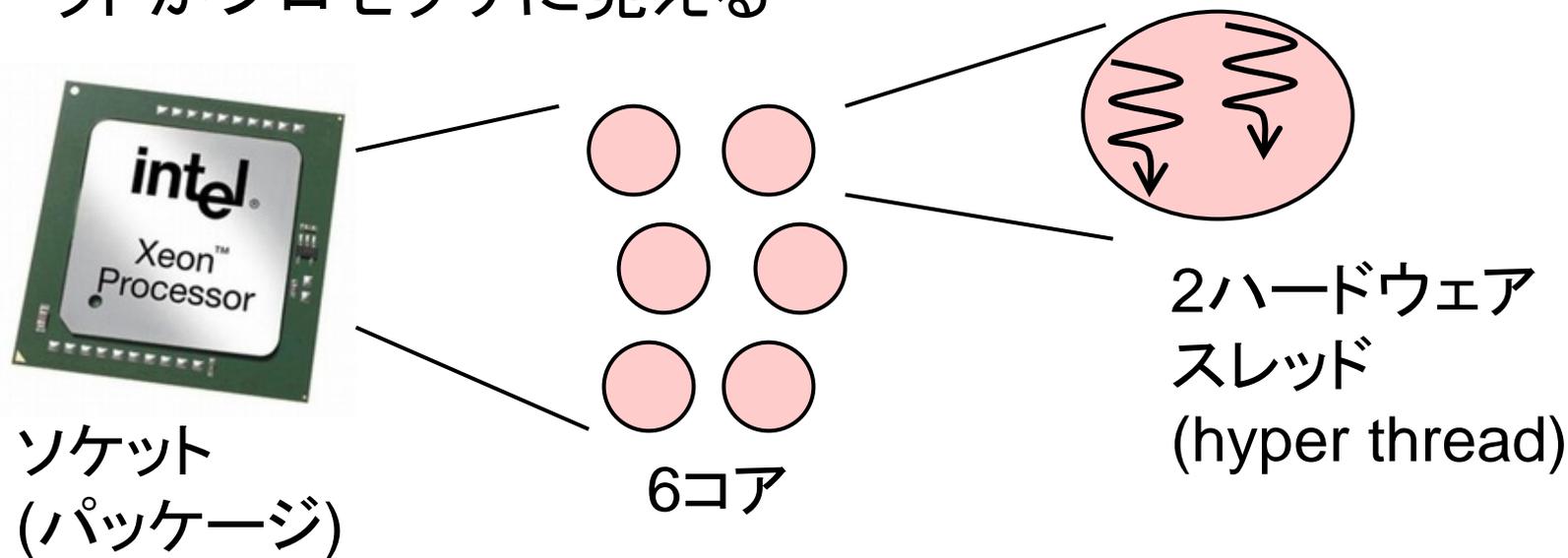
近年のスパコンのほとんど全ては共有と分散の組み合わせ
計算ノード内は共有メモリ, 計算ノード間は分散メモリ





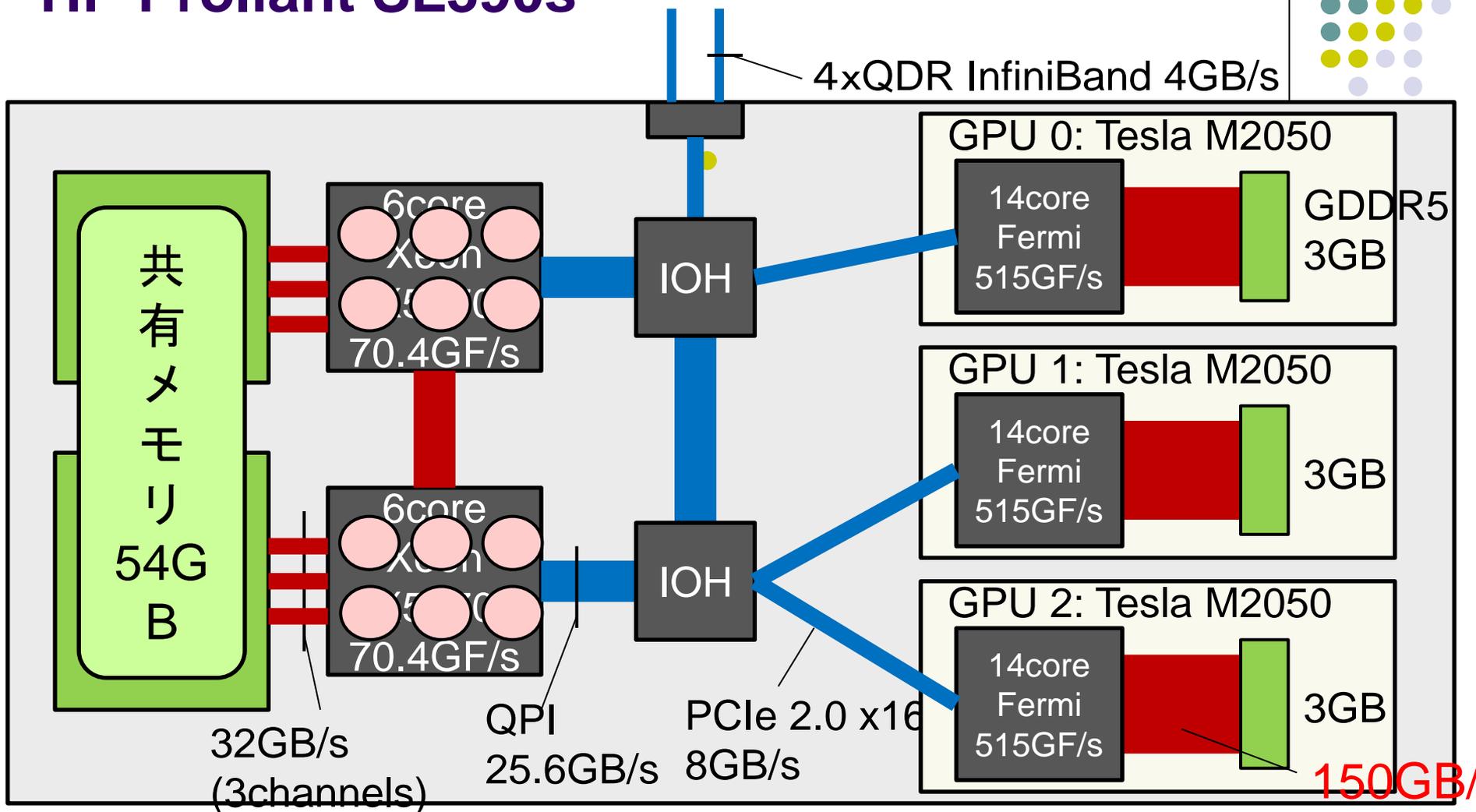
用語:ソケットやコア

- マルチコア時代になり、プロセッサの定義が複雑に
 - プロセッサの機能を持つ「コア/CPUコア」を複数、パッケージに詰め込むようになった
 - HyperThreadingによりさらに複雑に. 1コアを2つのハードウェアスレッドが共有する. OSからはハードウェアスレッドがプロセッサに見える



TSUBAME2のノードアーキテクチャ

HP Proliant SL390s



HyperThreadingのため、2ソケット×6コア×2HT=24プロセッサに見える。/proc/cpuinfoファイルの内容を参照

TSUBAME2のネットワーク構成

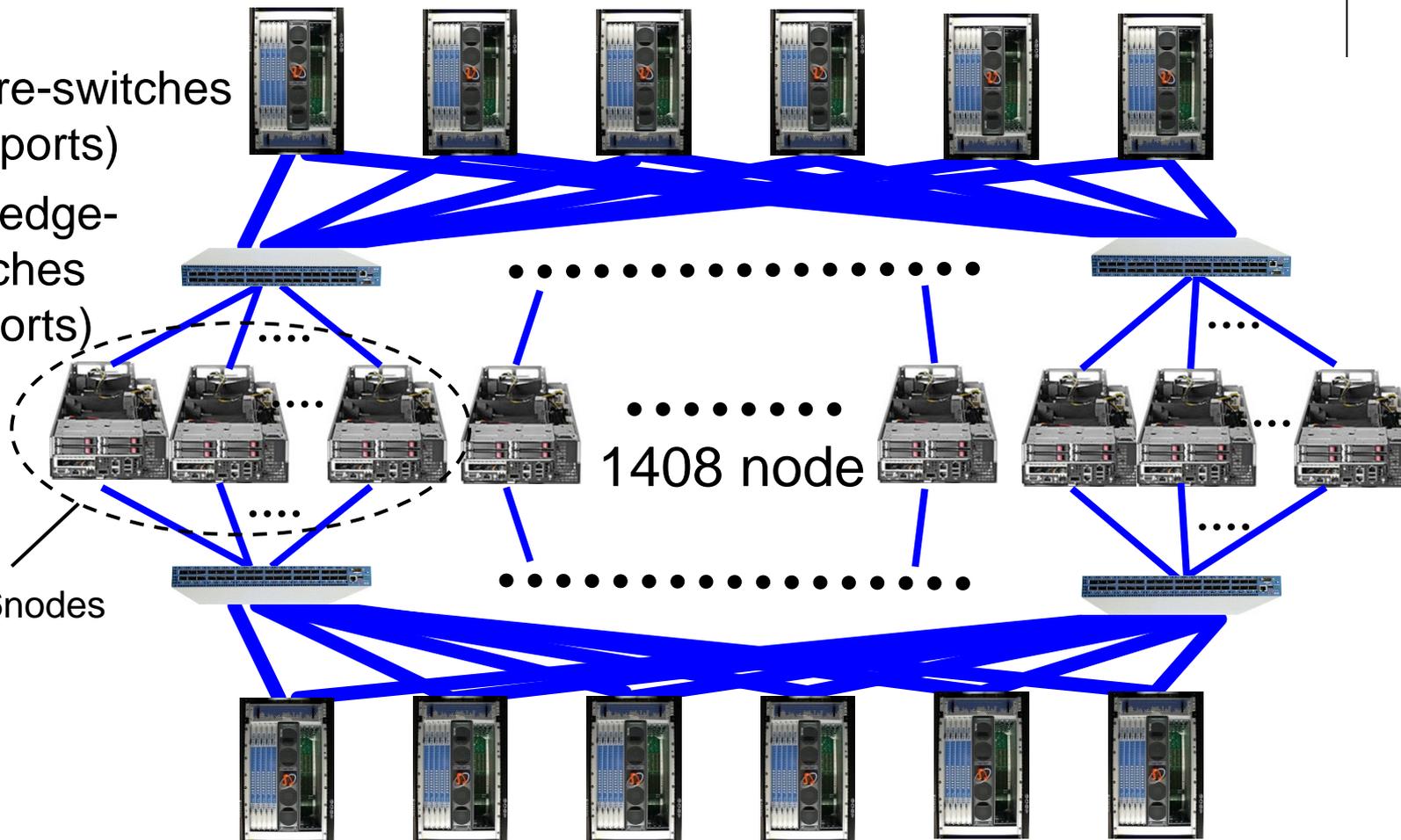
Dual-rail Full-bisection Fat-tree topology



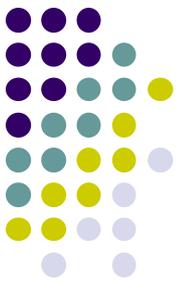
6 core-switches
(324ports)

~90 edge-
switches
(36ports)

14~16nodes



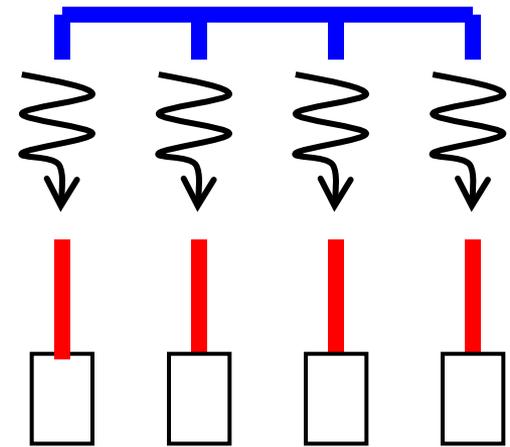
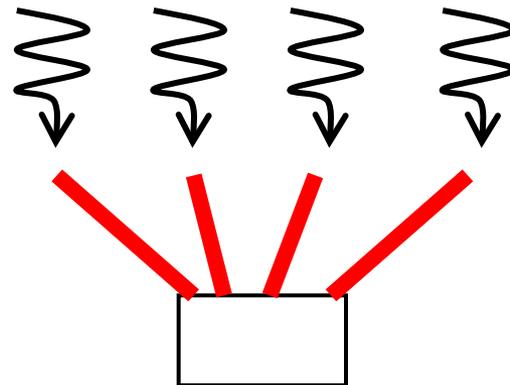
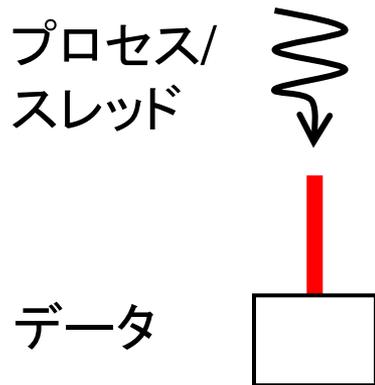
並列プログラミングモデルの、 メモリモデルによる分類



逐次

共有メモリモデル

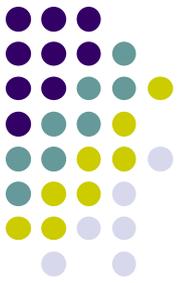
分散メモリモデル



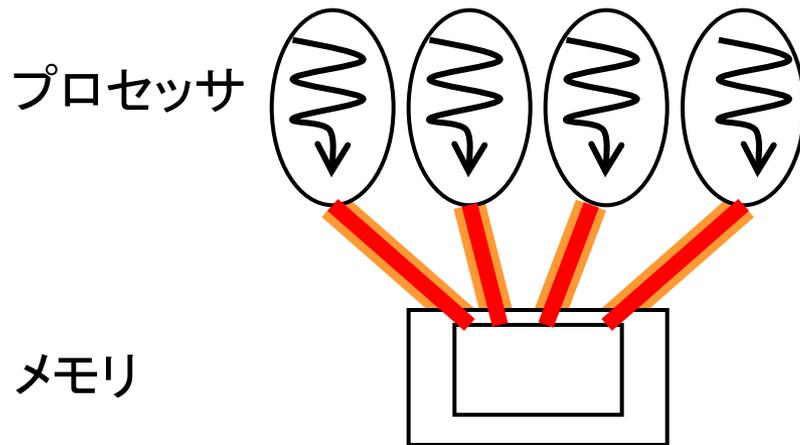
- スレッド達が共通のデータにアクセス可能
- **OpenMP**(言語拡張)
 - pthread(ライブラリ)
 - Cilk (言語拡張)

- プロセス間では通信が必要
- **MPI** (ライブラリ)
 - socket (ライブラリ)
 - UPC, HPF(言語拡張)
 - MapReduce

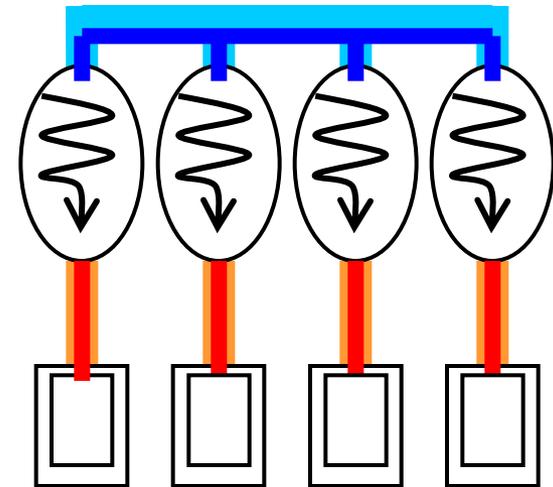
プログラミングモデルとアーキテクチャ (1)



共有メモリアーキテクチャ

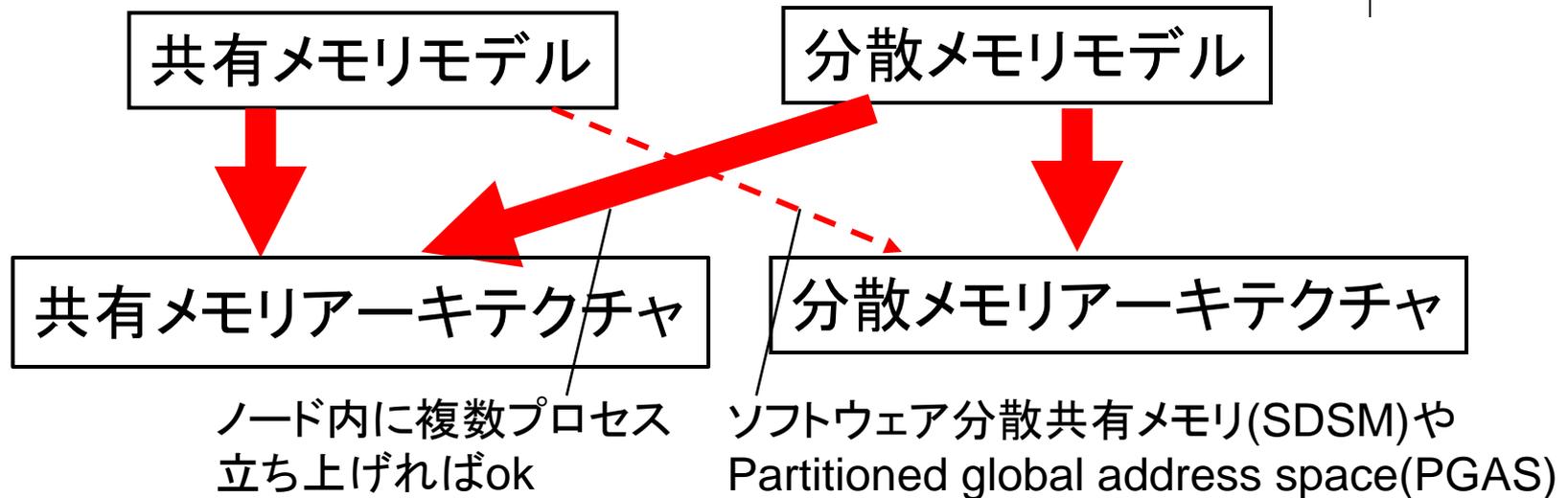
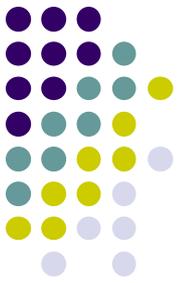


分散メモリアーキテクチャ



必ずしもモデルとアーキテクチャが一对一の対応ではない

プログラミングモデルとアーキテクチャ(2)



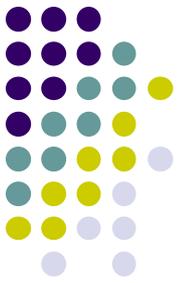
TSUBAME2の場合:

- 共有メモリモデル(OpenMPなど)で、1ノード内の12コアまで利用できる
- ノードをまたぐ、大規模並列を行いたい場合、分散メモリモデル(MPIなど)でソフトウェアを作る必要
 - 効率向上のために、MPIとOpenMP両方使う場合も



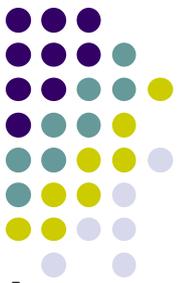
次回: OpenMPとは

- 共有メモリモデルによる並列プログラミングAPI
 - Pthread, Java threadより組むのがちょっと楽
- C言語, C++, Fortranに対応
- 並列化のための指示文や, ライブラリ関数
 - 指示文: #pragma omp ~
- 基本はFork-Joinモデル
- 変数は基本的にスレッド間で共有
- OpenMPで明示的に記述する必要のあること:
 - タスク分割
 - スレッド間同期
 - 変数の共有・プライベートの区別



参考文献

- OpenMP official site
 - <http://www.openmp.org>
- OpenMPチュートリアル
 - http://www.cc.u-tokyo.ac.jp/support/kosyu/03/kosyu-openmp_c.pdf
 - <http://www.nag-j.co.jp/openMP/>
- 牛島省, 「OpenMPによる並列プログラミングと数値計算法」, 丸善, 2006
 - Red-black SOR, LU分解, FFTなど
 - FORTRAN言語による



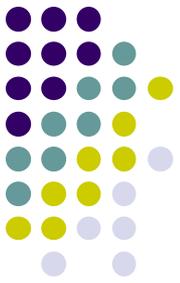
About Account

- TSUBAME2のアカウントができたなら、連絡してください。
授業用のTSUBAMEグループへ登録します。

Subject: TSUBAME2 ppcomp account

To: endo@is.titech.ac.jp

- 専攻・研究室
- 学年
- 氏名
- アカウント名



次回/Next Lecture

- 4/22(Mon)
- OpenMPによる共有メモリ並列プログラミング(1)