LinuxにおけるNUMAサポート

2003年10月10日

NEC

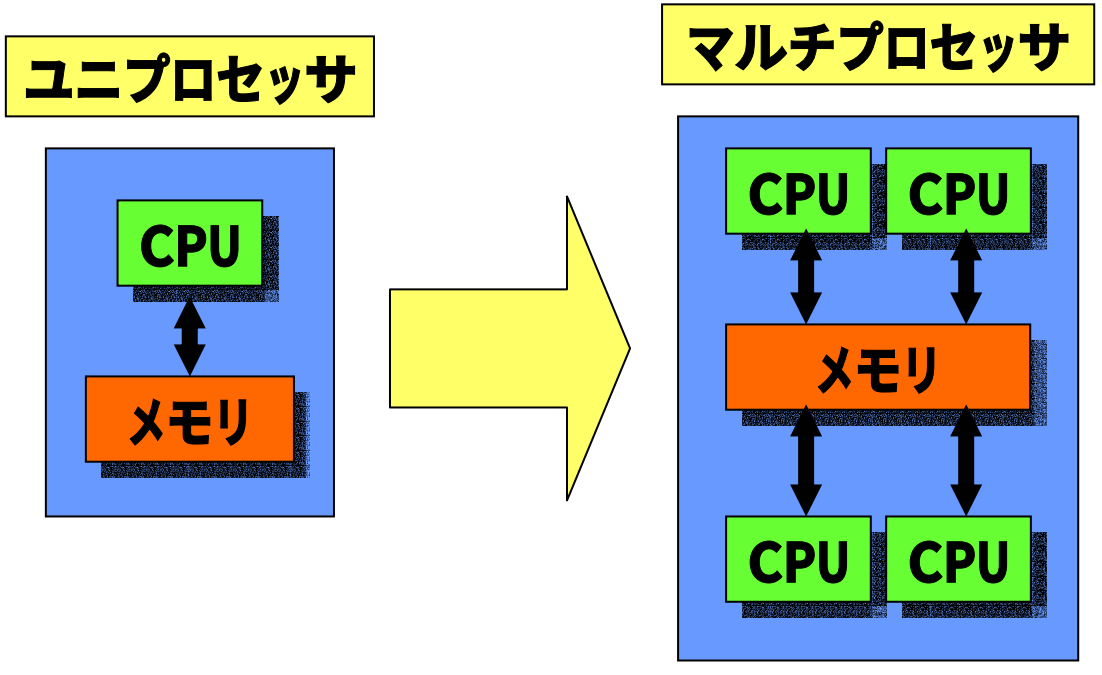
河内隆仁

目次

- NUMAアーキテクチャの概要
- カーネル2.6でのNUMAサポート
- 今後のNUMAサポート

NUMAアーキテクチャの概要

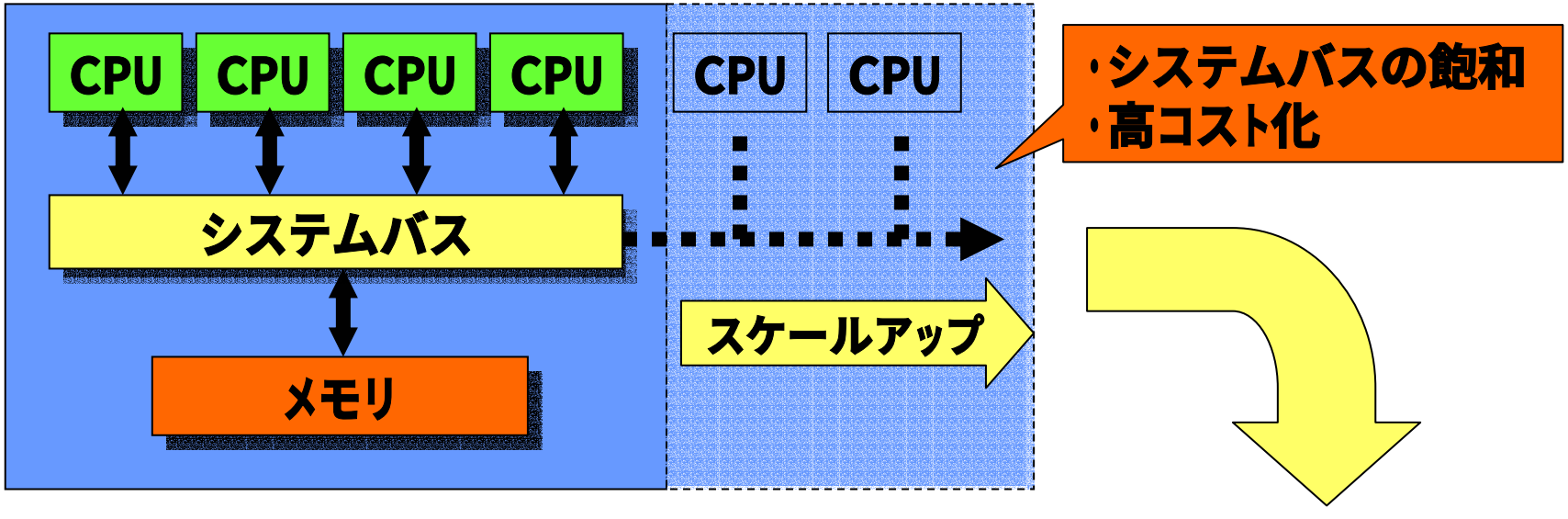
背景:マルチプロセッシングシステム



単体CPUの性能向上を上回る計算能力への要求に応えるため、数々のマルチプロセッシングシステムが考案されている。その中で最も多く利用されているものが「共有メモリ型マルチプロセッサ」

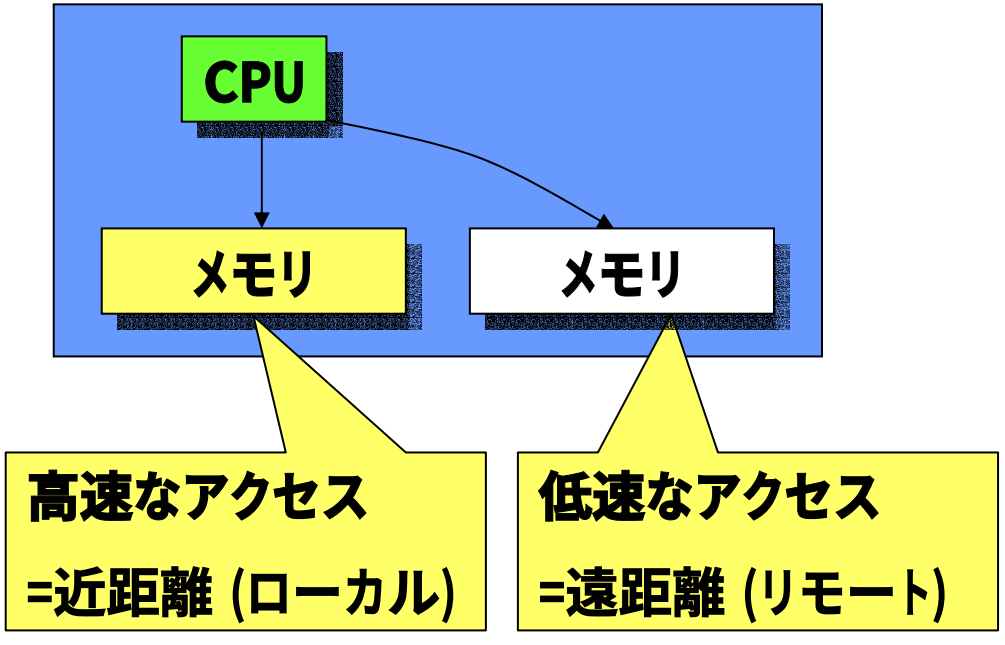
共有メモリ型マルチプロセッサの進化

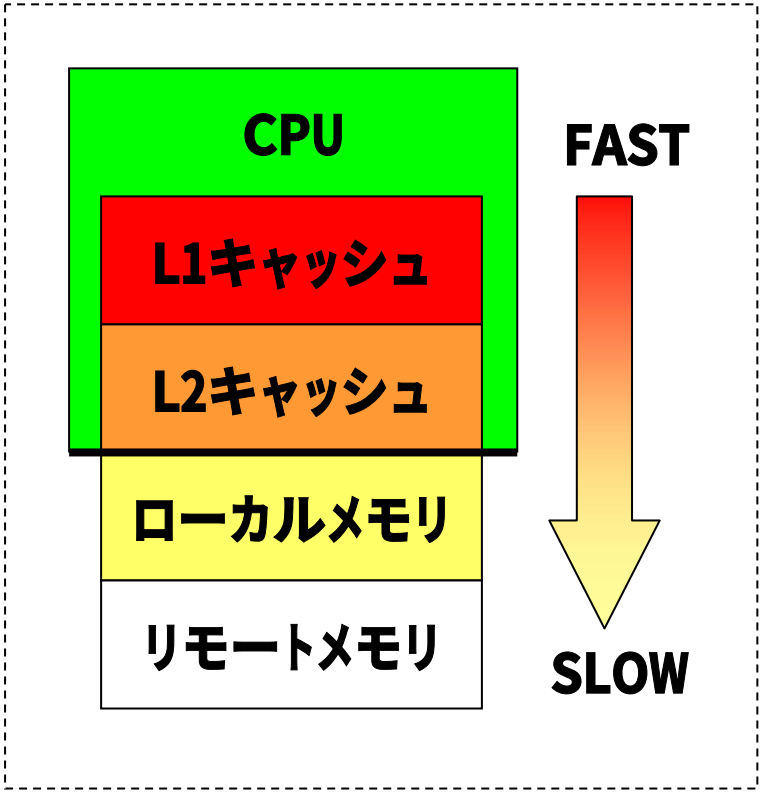
SMP (Symmetric Multi-Processors: 対称型マルチプロセッサ)



- NUMA (Non-Uniform Memory Access: 非均一なメモリアクセス)
- キャッシュの増量・多階層化
- オンチップマルチスレッド

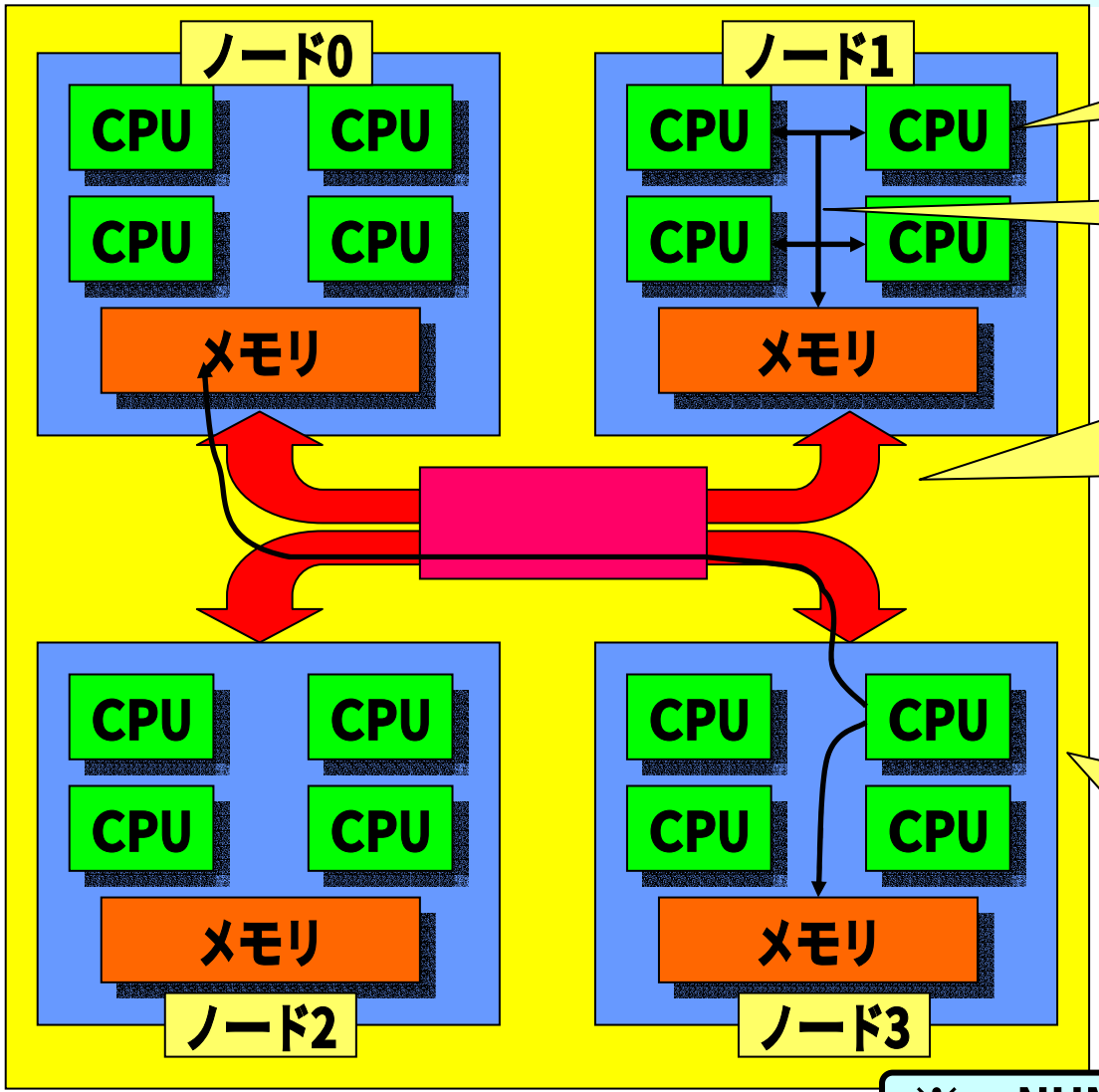
NUMA (Non-Uniform Memory Access)





NUMA: メモリの場所によって、CPUとメモリの間の距離 (レイテンシ) やバンド幅が異なる ⇔ UMA

NUMAアーキテクチャの例(1)



1ノード4CPU

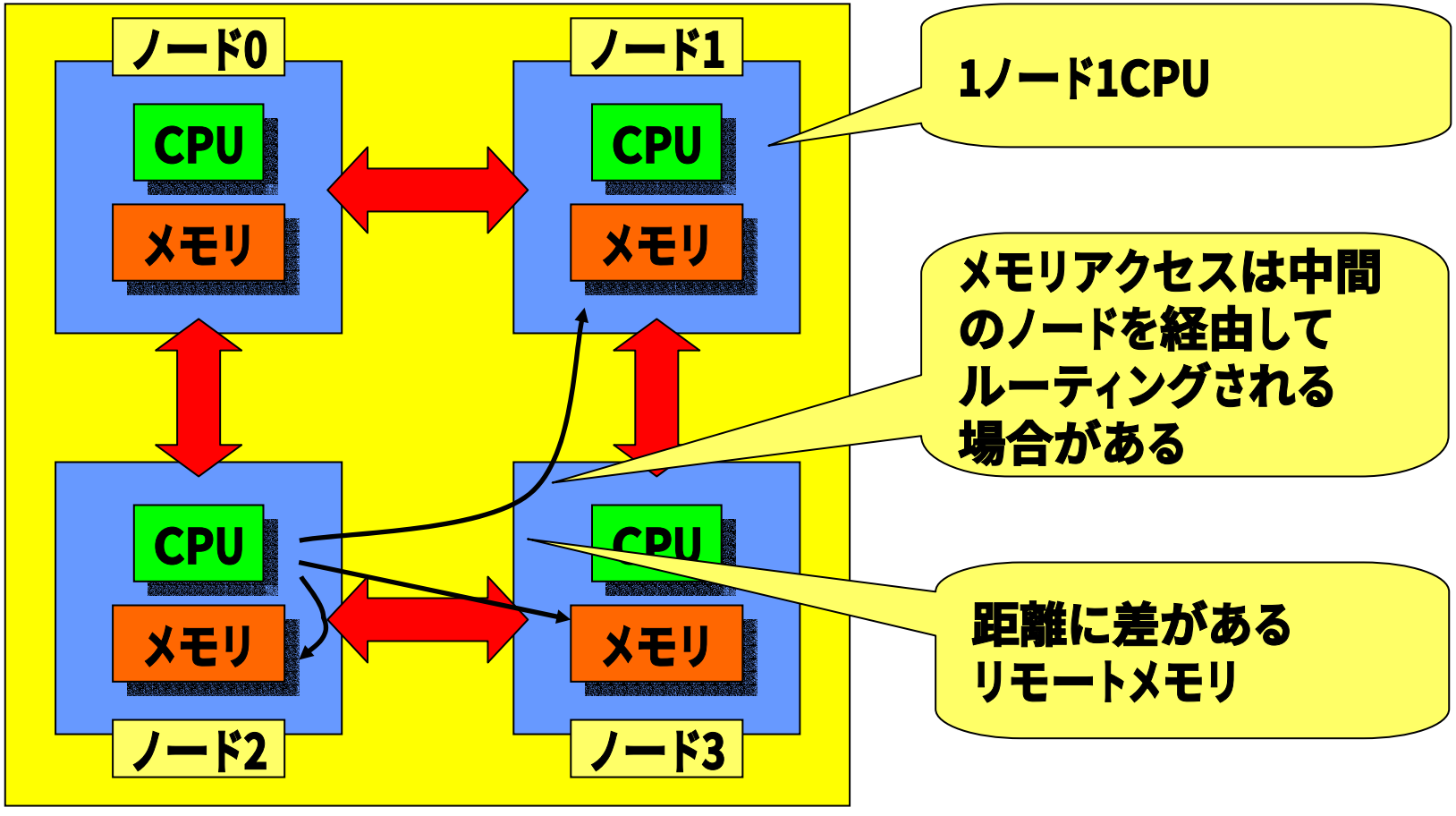
ノード内はUMA

全体で一つのメモリ空間を一つのカーネルで制御する

全てのプロセッサ間でキャッシュの一貫性は保たれる (ccNUMA)
⇒ SMPとしても扱える

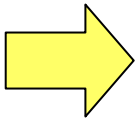
※ccNUMA : Cache-Coherent NUMA

NUMAアーキテクチャの例(2)



NUMAアーキテクチャの特徴

- 非均質なメモリアクセス
  - メモリの場所により異なるレイテンシ・バンド幅
- ローカル/リモートの区別
  - “ノード”, “ノード間の距離”
- SMPとのソフトウェア的な互換性
  - キャッシュの一貫性が保たれるため、ソフトウェアからはSMPに見える



このようなNUMAでよい性能を得るには…

メモリ参照をできるだけローカライズ(局所化)する

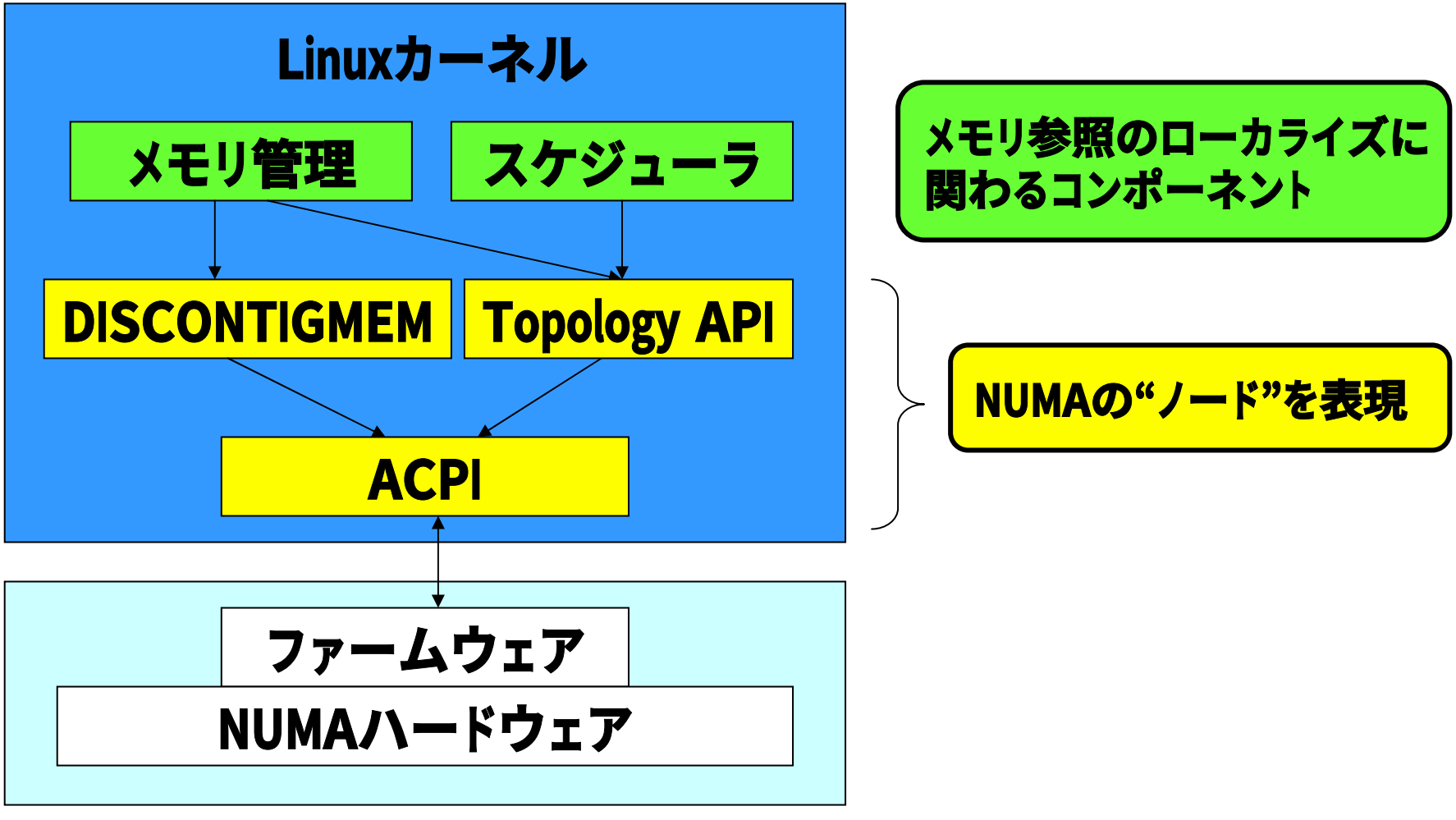
Linuxでサポートされるプラットフォーム

- alpha
- i386 (IBM NUMA-Q, IBM x440)
- ia64 (DIG64, SGI Altix)
- mips
- ppc64
- x86_64

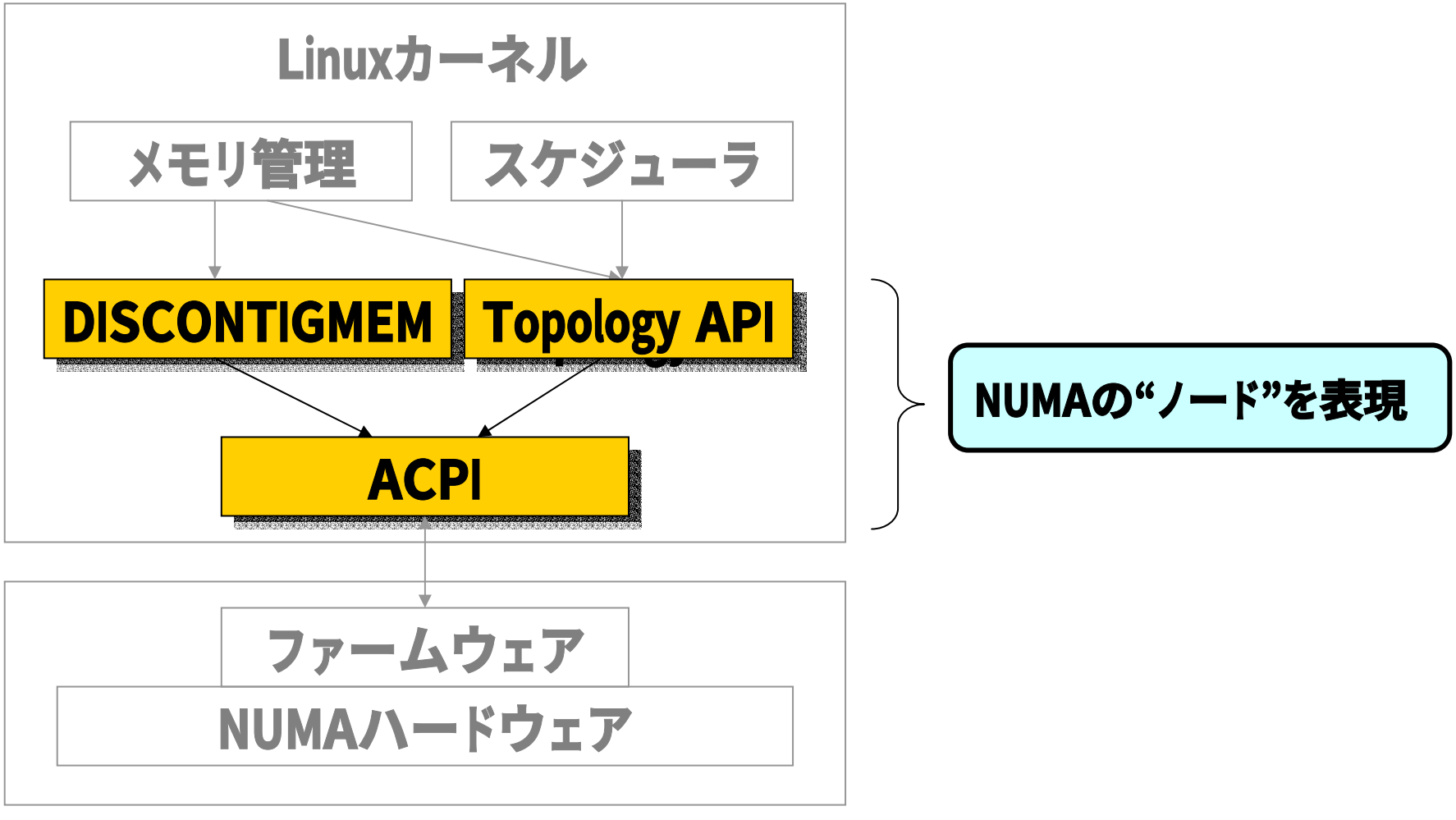
多様なプラットフォームがサポートされており、各アーキテクチャーのベンダーや技術者が協力して開発を行ってきた

カーネル2.6でのNUMAサポート

カーネル2.6でのNUMAサポート



カーネル2.6でのNUMAサポート



“ノード”の表現 (1/3)：ACPI

- ACPI NUMA support (2.5.33)
  - ACPIから以下の情報を取得
    - ノードに属する物理メモリアドレスの範囲
    - ノードに属するCPU
    - ノード間の距離

※ACPI: Advanced Configuration and Power Interface

“ノード”の表現 (2/3)：トポロジー

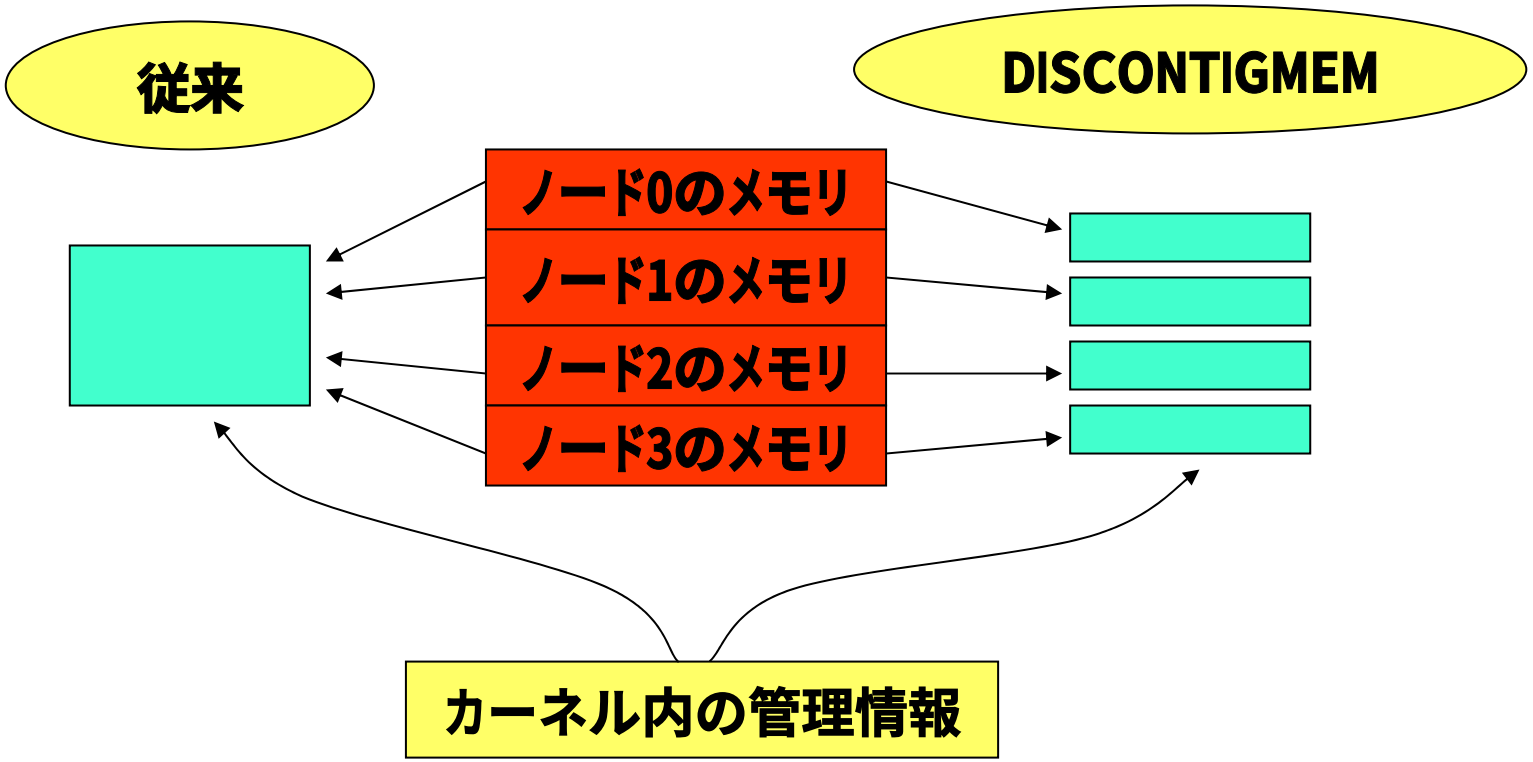
- Topology API (2.5.40)
  - **カーネル内部で**
    - **プロセッサの属しているノード**
    - **それぞれのノードに属するメモリの範囲**

  …等を取得するAPI群が提供されている

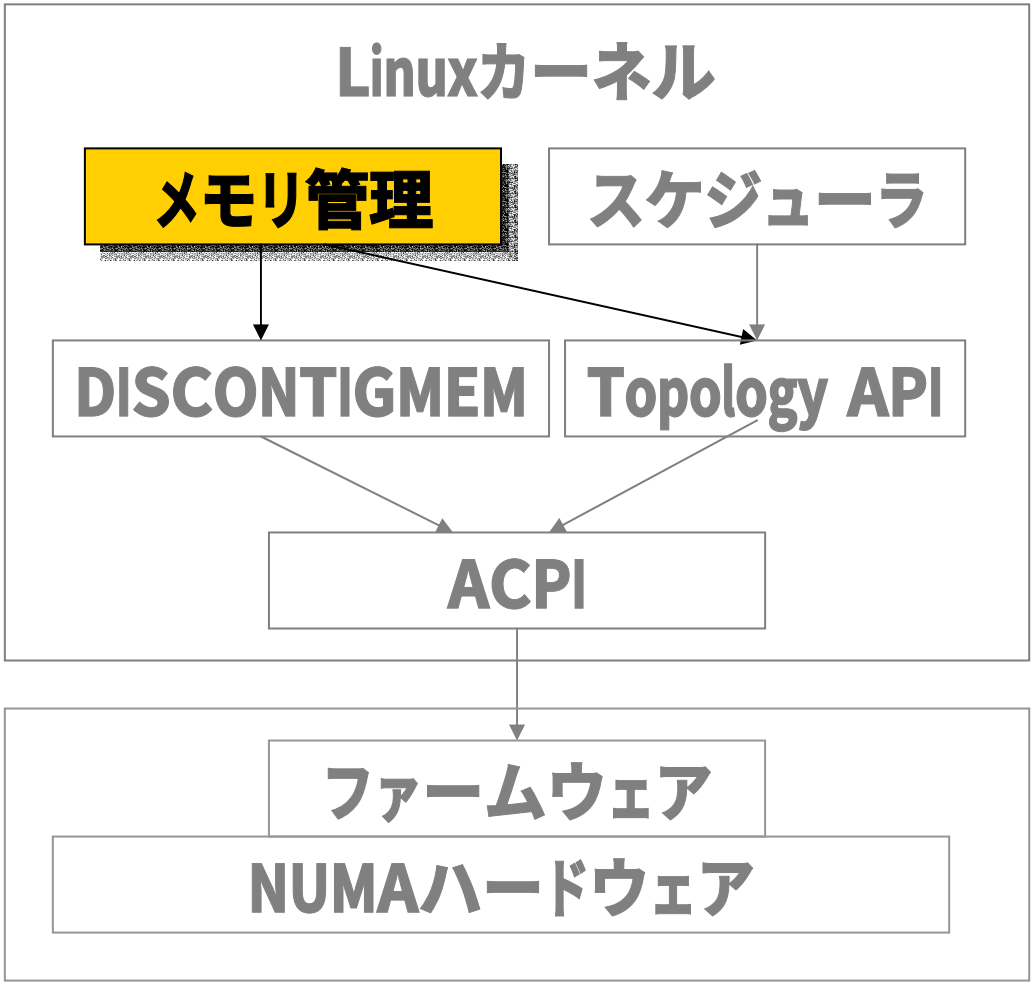
例:
cpu_to_node(cpu) : cpuの属するノード番号を返す

“ノード”の表現 (3/3)：DISCONTIGMEM

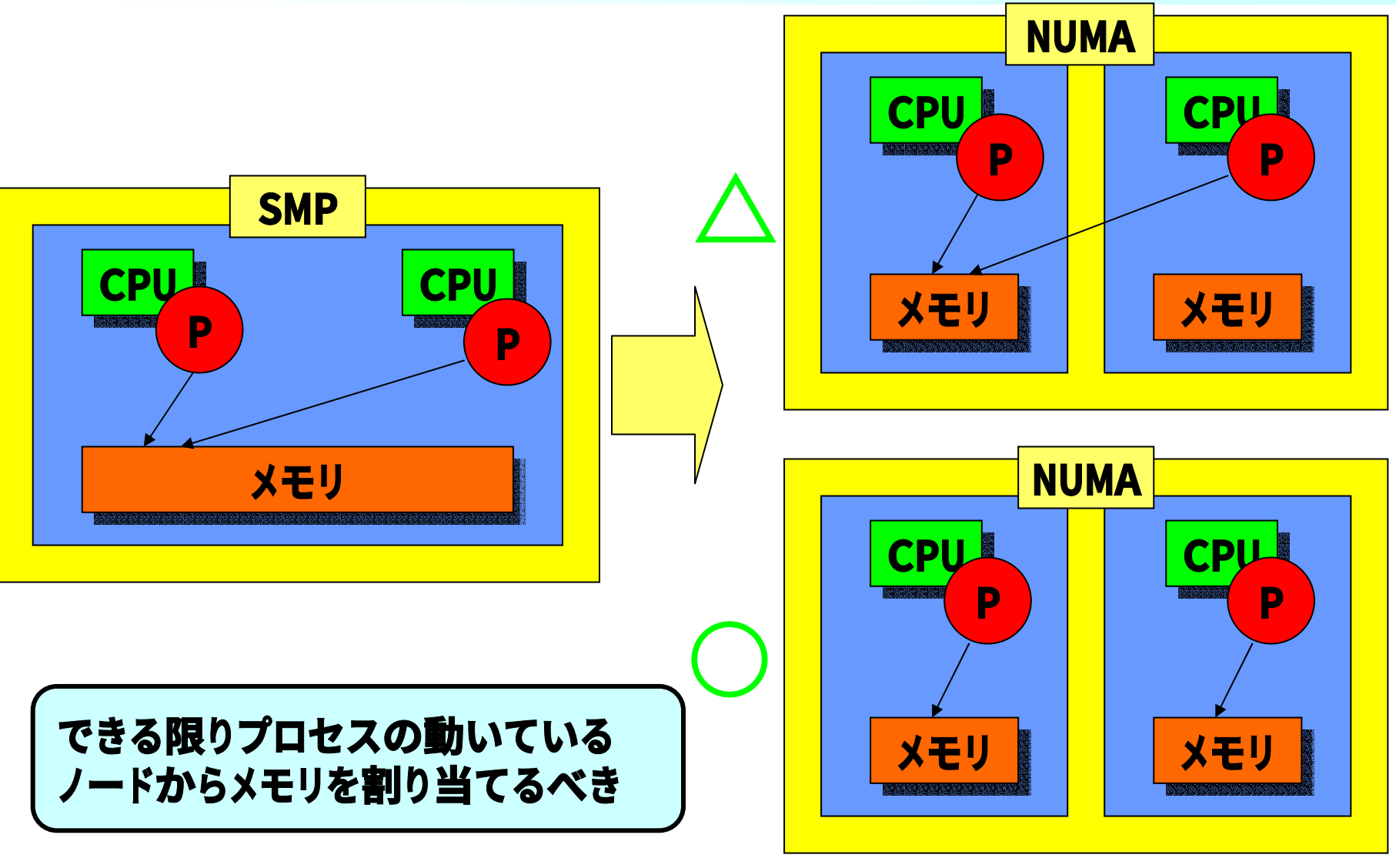
- DISCONTIGMEM = Discontiguous Memory (**非連続メモリ**)
- **物理アドレスが非連続なアーキテクチャをサポートする機構**
- NUMAでもこの機構を利用し、ノード毎のメモリを別々に管理



カーネル2.6でのNUMAサポート



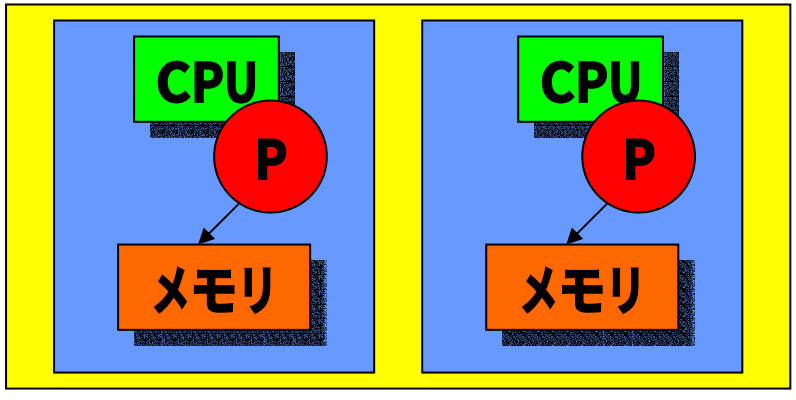
メモリ割り当て: SMPとNUMA



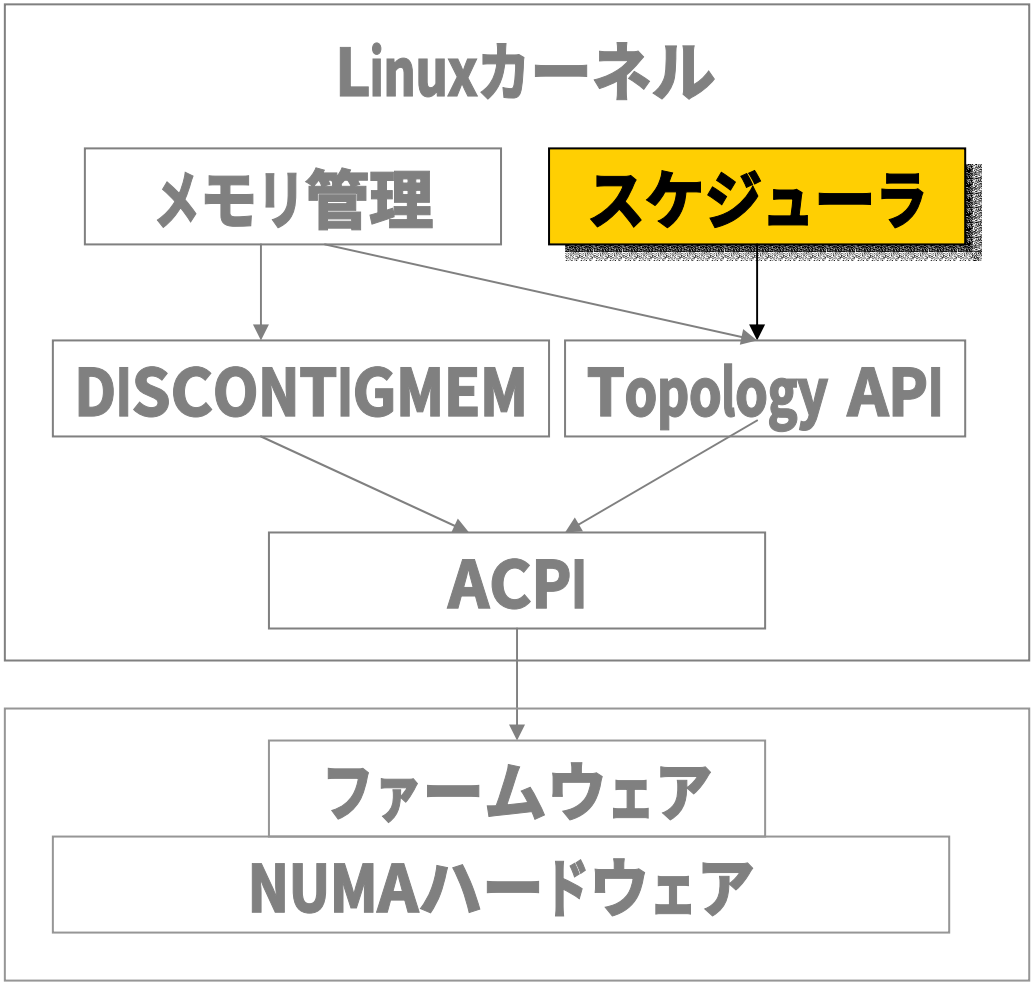
できる限りプロセスの動いている
ノードからメモリを割り当てるべき

ノードを意識したメモリの割り当て

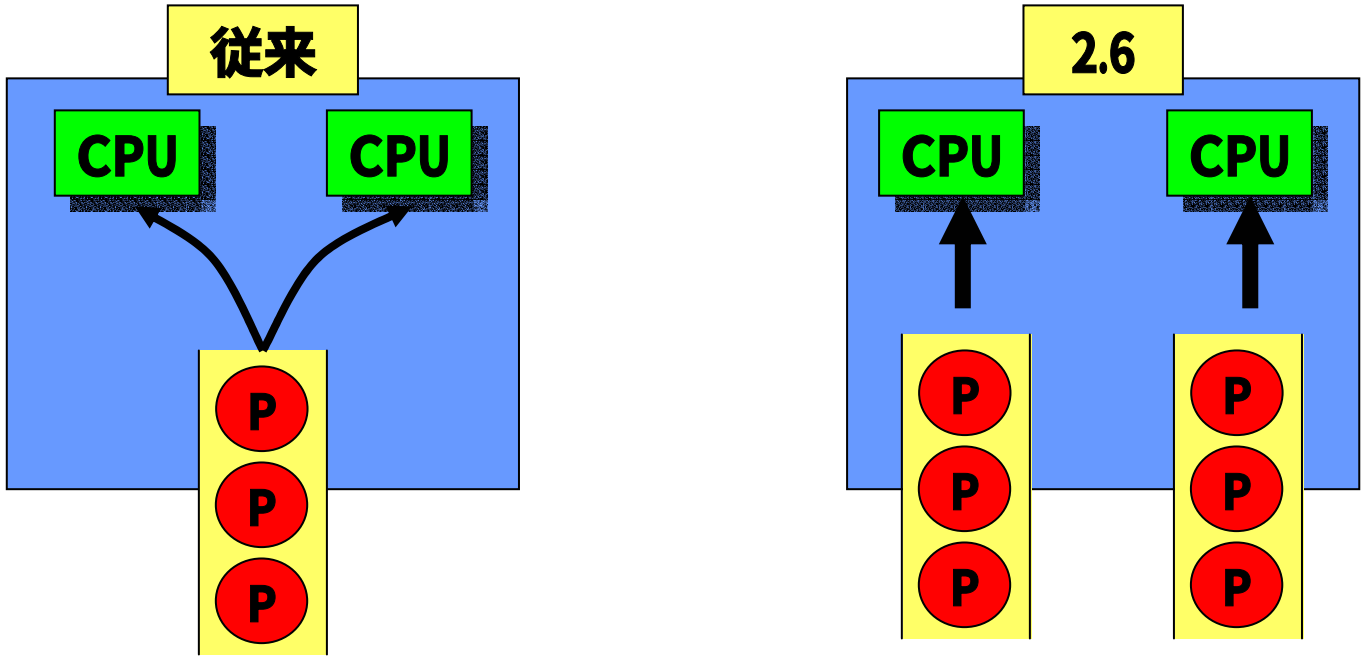
- DISCONTIGMEM**機構の利用**
- **ページ割り当て時に、プロセスが動いている ノードからメモリを確保** (first touch)
- **ノードのメモリが不足しているときには、 隣接するノードから順に空いている メモリを探す**



カーネル2.6でのNUMAサポート

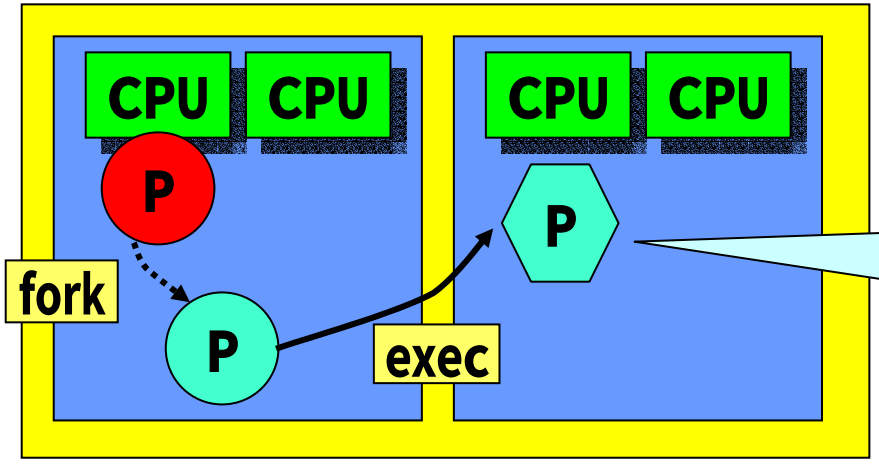


2.6でのスケジューラの改善

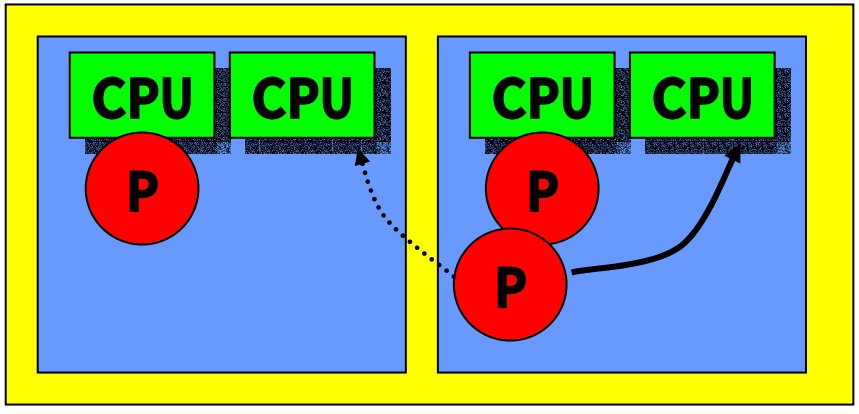


O(1)スケジューラによって、プロセスのランキューがCPU毎に分割された。この結果、CPUとプロセスの結びつきが強くなり、プロセスがCPU間を移動しにくくなっている

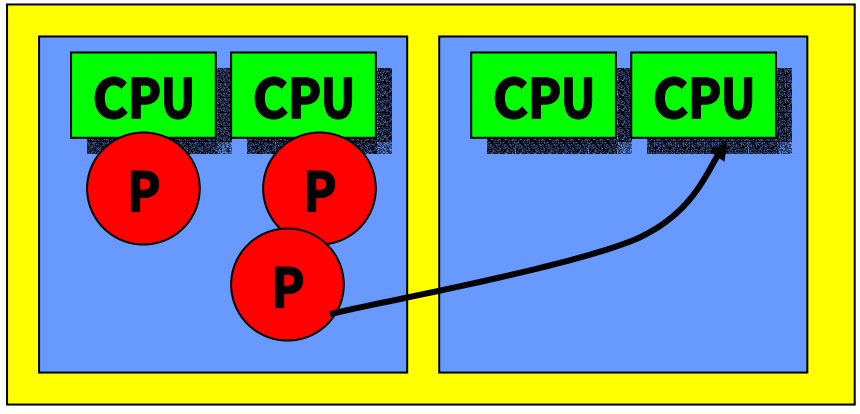
2.6のNUMAスケジューラ



exec時に、そのプロセスを
置くノードを決定する



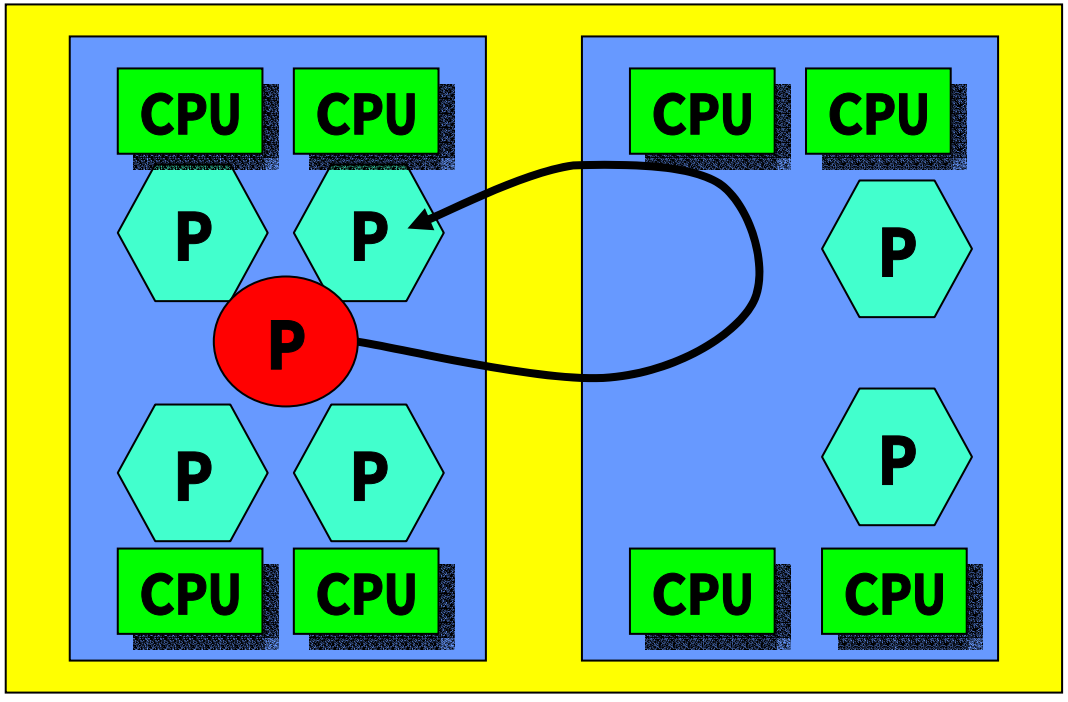
ノード内で負荷バランス

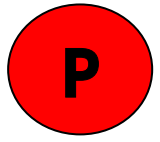


ノード間で負荷バランス

NUMAスケジューラの拡張 (Erich Focht)

- homenodeの概念の導入
  - 一時的なノード間の移動を、元に戻す





長寿命プロセス

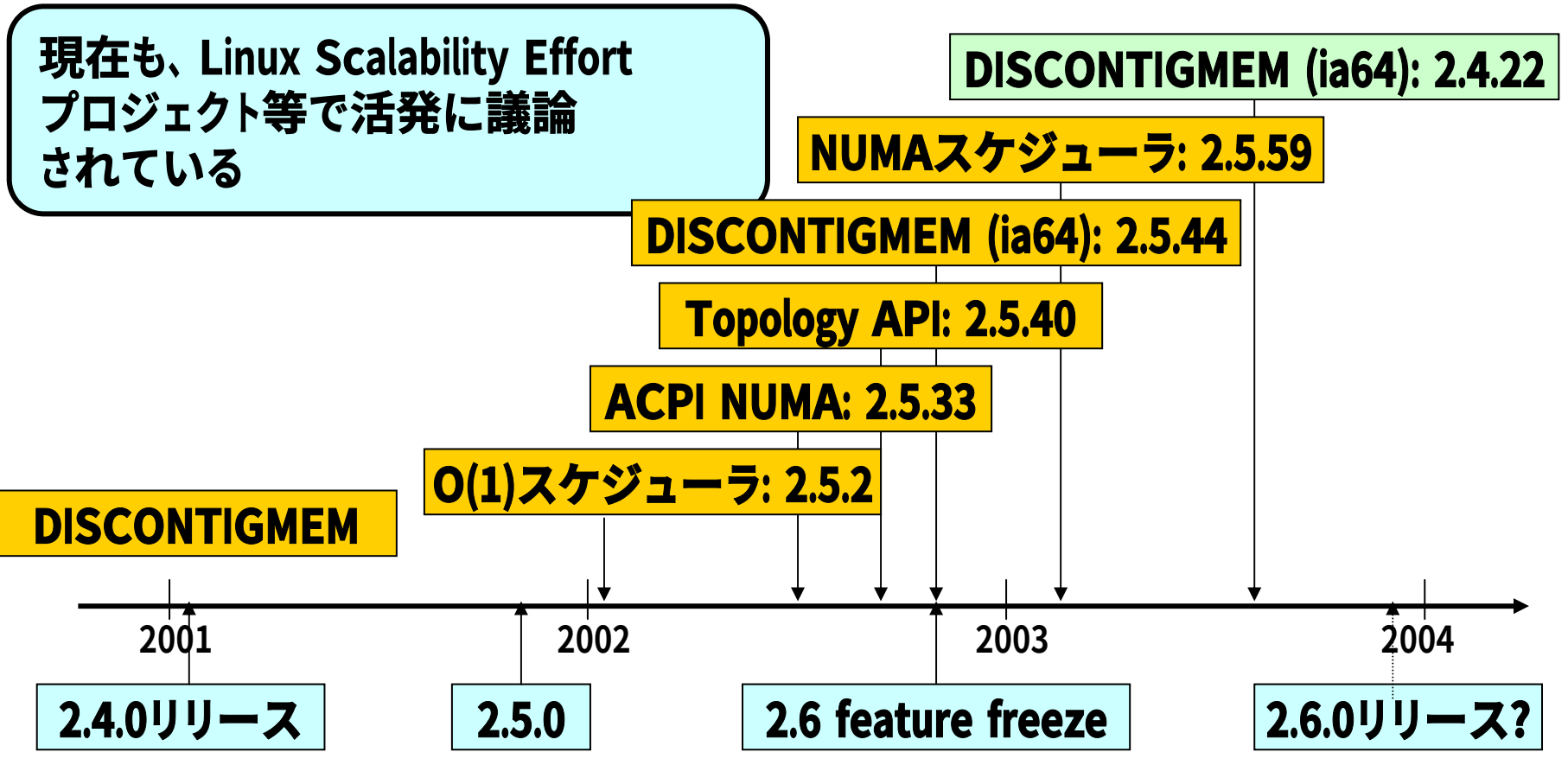
短寿命プロセス

現在の実装に存在する課題

- 十分な対処ができていないケース
  - ページキャッシュに乗ったページの扱い
  - スレッドのスケジューリング
  - リモートメモリ参照の多いプロセスの検出、移動
  - etc.
- 標準的なユーザーインターフェイス不在
  - NUMA APIへの期待

NUMAサポートの歴史

現在も、Linux Scalability Effort プロジェクト等で活発に議論されている



今後のNUMAサポート

NUMAアーキテクチャの動向

- さらなる大規模化
  - 64プロセッサを超えるプラットフォームの出現
- 非SMPアーキテクチャの増加
  - オンチップマルチスレッド
  - マルチコアCPU
- 小規模サーバやPCへも浸透

2.7以降の課題と展望

- NUMAを含む非SMPマルチプロセッサプラットフォームをより統一的な形でサポート
- ユーザレベルのNUMA API
- カーネルのさらなるNUMA最適化
  - 多プロセッサ時のスケーラビリティ改善
  - I/O, 割り込みのローカル化
  - NUMA-aware lock
  - カーネルテキスト/ユーザテキスト・ライブラリのレプリケーション
  - etc.

参考URL・文献

- Linux Scalability Effort (LSE)
  http://lse.sourceforge.net/
  Linux**カーネルのスケーラビリティに関する話題全般を扱う**
- Discontig Project
  http://discontig.sourceforge.net/
  NUMA**サポートに関する話題を扱う**
- Ottawa Linux Symposium 2003 Proceedings
  http://archive.linuxsymposium.org/ols2003/Proceedings/
- Node affine NUMA scheduler (Erich Focht)
  http://home.arcor.de/efocht/sched/

Thank You!