

次世代メニーコア型スパコンのための システムソフトウェア

堀 敦史

システムソフトウェア技術部会 部会長（理研 AICS）

清水 正明

システムソフトウェア技術部会 副部会長（日立）

2020/12/15



- IHK/McKernel
 - 最新情報
- Process-in-Process (PiP)
 - 新しい並列実行モデル

- マルチ OS カーネル

- 軽量カーネルと（重量）Linux カーネルのコンビネーション
- （OSサービスをあまり必要としない）並列アプリは McKernel 上で動作
- 頻度の低いシステムコールは Linux カーネルに委託

- 特長

- ノイズが非常に少ない
- Linux に比べ機能追加が容易



Process-in-Process (PiP)

- マルチプロセスモデル（MPI）とマルチスレッドモデル（OpenMP）の
いいとこ取り
- **特長**
 - 容易なプログラミング（共有メモリとは似て非）
 - 高性能なノード内通信
 - In-situ などへの応用の可能性

McKernel と PiP の開発状況

- 富岳への導入を目指し
 - Aarch64 へのポーティング（ほぼ終了）
 - 品質向上（x86_64、aarch64）
- 富岳上で安定動作を目指し鋭意作業中
 - McKernel
 - 現在バージョン 1.7.0
 - 3月末 1.8.0
 - PiP
 - これまで PiP-V1（最初の公開版）
 - PiP-V2 および PiP-V3 がリリース

PiP の可能性

- **PiP は新しいノード内並列実行モデル**
 - ノード内通信の高速化「以外」への応用
- **新しいスレッドモデル**
 - 従来
 - システムレベルスレッド（カーネルスレッド）
 - 利点 割り込み可（プリンプティブなスケジューリング）、システムコール対応
 - 欠点 遅いスレッド切り替え
 - ユーザレベルスレッド
 - 利点 高速なスレッド切り替え
 - 欠点 割り込み不可、システムコールに対応できない
 - 新しいスレッド実行モデル

新しいスレッドモデル

- **バイレベルスレッド**
 - スレッドはカーネルスレッドにもユーザレベルスレッドに「も」なれる
 - 例えば
 - システムコールを呼ぶ時はカーネルスレッド、そうでない時はユーザスレッド
- **さらに、ユーザレベルプロセス**
 - プロセス間のコンテキスト切り替えをユーザレベルで実現
 - アドレス空間を共有する PiP でのみ実現可能
- **バイレベルスレッド、ユーザレベルスレッドとも実装済み**
 - PiP-V3

PiP のインストールツール

- **PiP のフルインストールには**
 - PiP のパッチが当たった Glibc および GDB が必要
 - これらをインストールするのは煩雑
- **PiP-pip (PiP Package Installing Program)**
 - <https://github.com/RIKEN-SysSoft/PiP-pip>
 - Yum、docker、spack、github のインストールが可能
 - RedHat/CentOS 7 および 8
 - x86_64 および Aarch64 に対応