
TOPPERSプロジェクト プレス発表
2009年4月23日

TOPPERS/FMPカーネル TraceLogVisualizer(TLV)

本田 晋也

名古屋大学 大学院情報科学研究科
附属組込みシステム研究センター(NCES) 助教

honda@ertl.jp

TOPPERS/FMPカーネル

組み込みシステムにおけるマルチプロセッサの利用

大きく二つの理由により利用が進んでいる

性能向上と消費電力削減の両立

- 低性能CPU複数個の消費電力 < 高性能CPU1個の性能
- 消極的なマルチプロセッサの利用
 - ソフトウェアエンジニアとしては高性能で低消費電力なプロセッサが使いたい

要件の異なるサブシステムの組み合わせ

- リアルタイム性と高機能性の両立
 - 携帯電話, カーナビ, NC工作機
- サブシステムの設計の容易化
- 積極的なマルチプロセッサの利用

マルチプロセッサ対応RTOSの必要性(1/2)

シングルプロセッサ用RTOSの利用

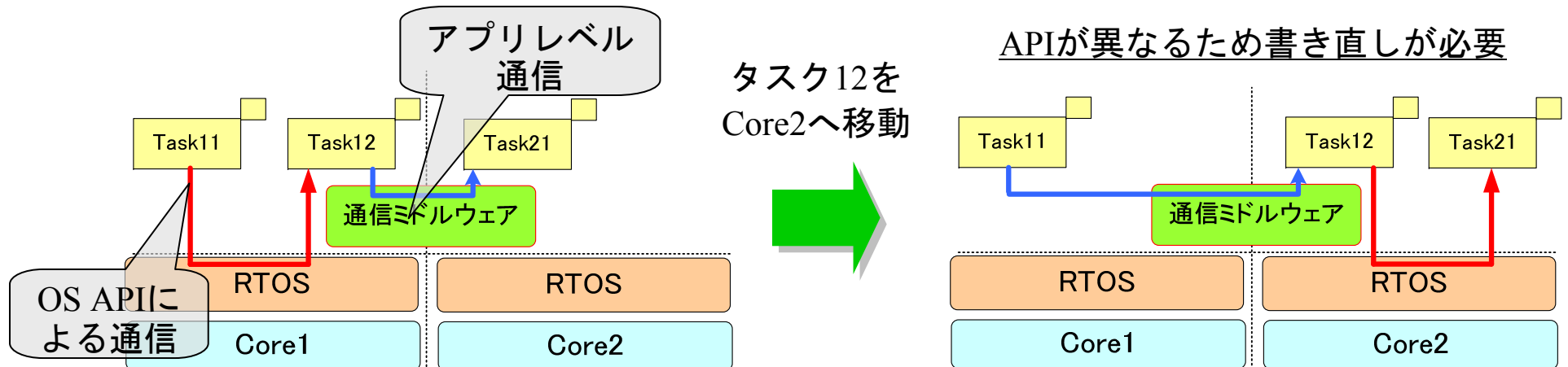
- 各プロセッサに独立にシングルプロセッサ用のリアルタイムOSを載せる(プロセッサによってはOSレスも)

プロセッサ間通信

- プロセッサ間の同期・通信は, (OSではなく)アプリケーションソフトウェアレベルで実現する必要がある

➡アプリケーション毎に開発する必要がある

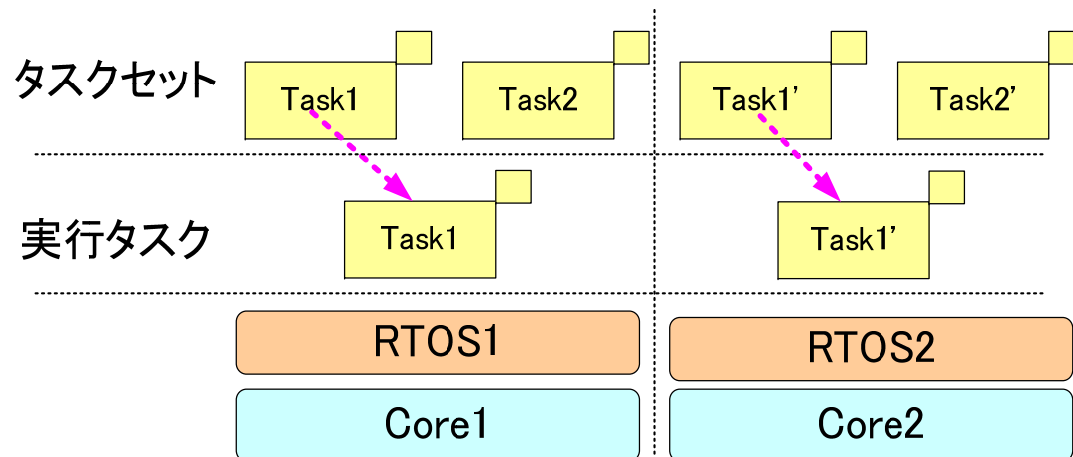
➡ある処理を別のプロセッサに移そうとすると, 同期・通信部分のプログラムの作り直しが必要



マルチプロセッサ対応RTOSの必要性(2/2)

タスクマイグレーション

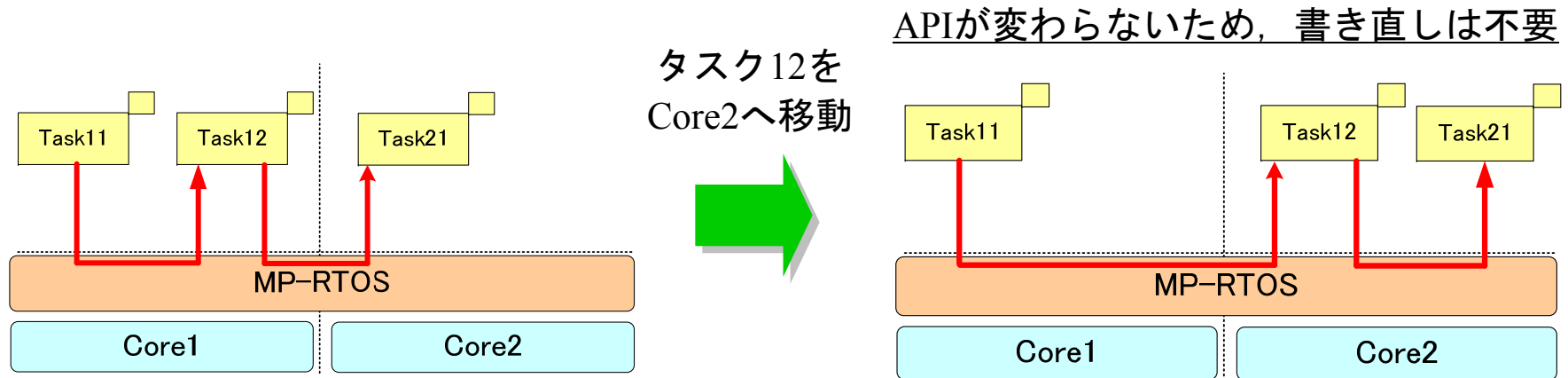
- 動的な負荷分散が必要な場合は、アプリケーションレベルで実現する必要がある
- ➡ アプリケーション毎の作り込みが必要
 - それぞれのコアに同じ処理を実行するタスクを用意して、必要に応じて起動するコアを変更する
 - ➡ 実行途中状態での処理の切り替えは困難



マルチプロセッサ向けOSの機能(1/2)

プロセッサ間通信のサポート

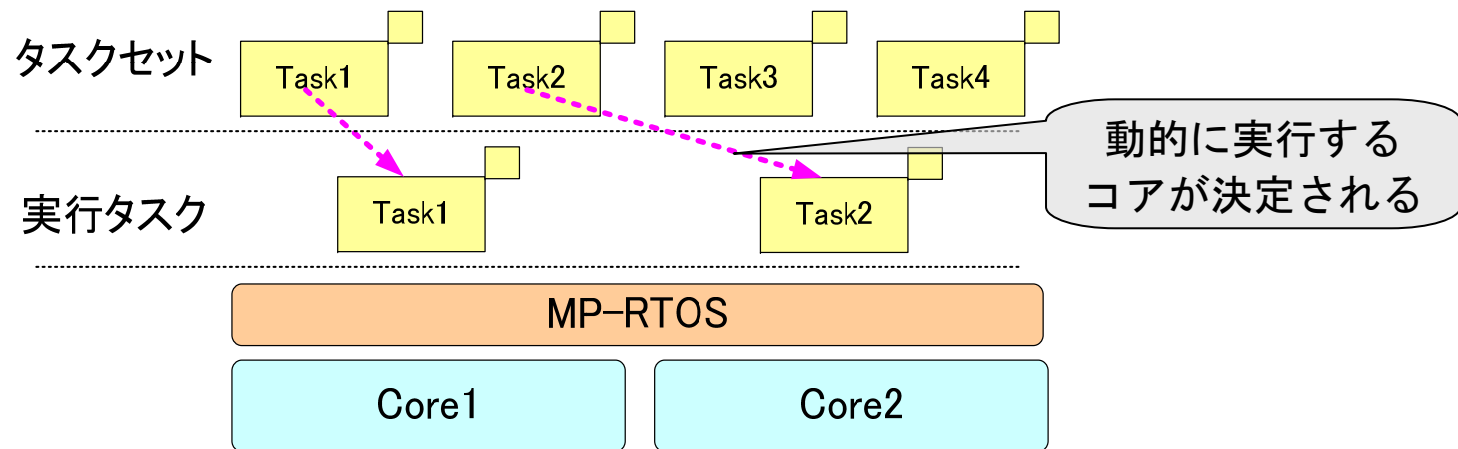
- アプリケーション毎にプロセッサ間の同期・通信を実現する必要がないため、開発工数が減少する
- プロセッサ内通信と互換の同期・通信機能
 - ある処理を別のプロセッサに(動的にも静的にも)容易に移動できる



マルチプロセッサ向けOSの機能(2/2)

動的な処理の割り当て(場合によっては)

- 負荷に応じて, スループットが最大になるよう, 処理(タスク)を動的にプロセッサに割り当てる
- タスクの途中状態での移動が可能
 - タスク切り替えと同じタイミングで可能
- 組込みシステムにおいては, リアルタイム性の観点からデメリットとなる場合があるので, 理解した上で使用することが重要(後述)



OSサポートの視点からのマルチプロセッサの分類

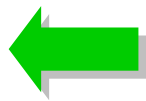
対称型マルチプロセッサOS (SMP-OS)

- プロセッサ内通信と互換のプロセッサ間通信
- 動的な処理(タスク)の割り当て

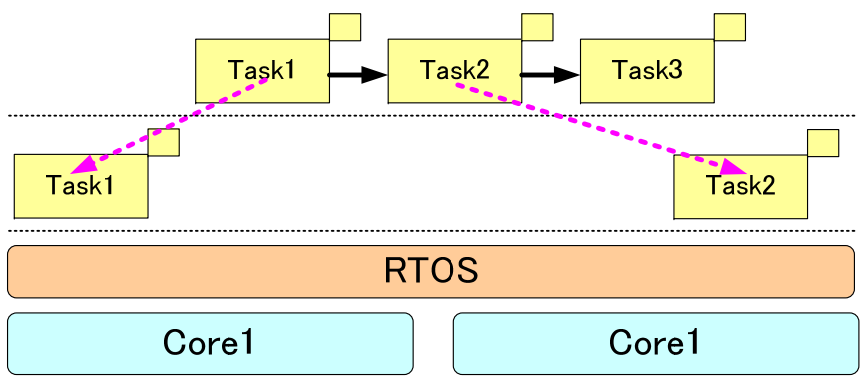
非対称型マルチプロセッサOS (AMP-OS)

- プロセッサ内通信と互換のプロセッサ間通信
- 静的な処理(タスク)の割り当て

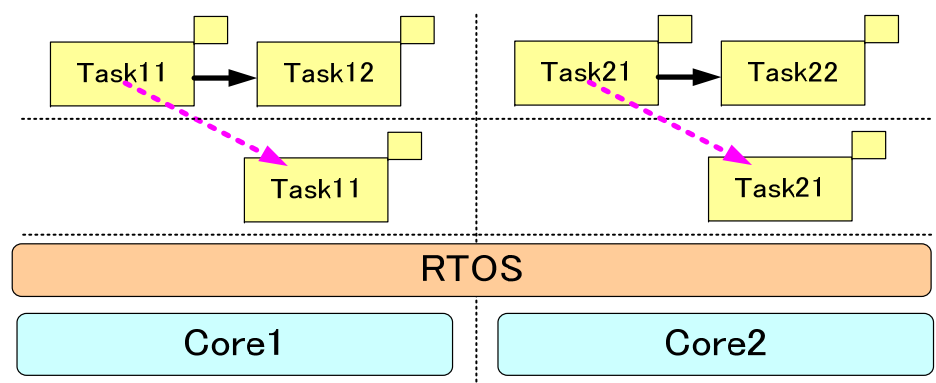
OSの内部構造の違いによる性質から、AMP-OSの方が組みみに適している場合が多い



SMP-OS



AMP-OS



ITRON仕様をマルチプロセッサ拡張した仕様と実装を開発

非対称型マルチプロセッサ用リアルタイムOS

- TOPPERS/FDMPカーネル

- FDMP (Function Distributed Multiprocessor)
- 2005年にカーネル仕様を公開
- 2006年4月にカーネル実装をオープンソースとして公開(1.1)

対称型マルチプロセッサ用リアルタイムOS

- TOPPERS/SMPカーネル

- SMP (Symmetric MultiProcessor)
- EPSONと名古屋大学との共同研究
- カーネル仕様を会員向けに公開

FDMP/SMPカーネルに対する要望や改良点

動的なタスク移動のサポート(FDMP)

- サポート範囲はリアルタイム性とのトレードオフ

リアルタイム性(SMP)

- SMPカーネルではリアルタイム性の保障は困難

アーキテクチャ最適化(FDMP/SMP)

- 特にメモリアーキテクチャに対する最適化を可能に

カーネルコード共有(FDMP)

- メモリのバンク数はプロセッサ数より少く、キャッシュの効率化のためには、カーネルのプログラム(バイナリコード)は、すべてのコアで共有

TOPPERS/FMPカーネル：概要

FMP = Flexible MultiProcessor

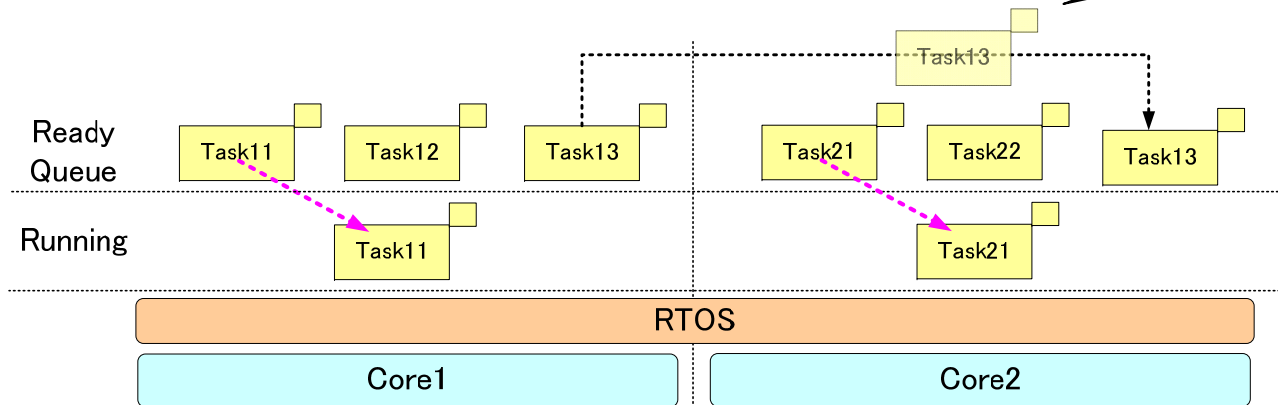
リアルタイム性と動的なタスク移動との両立を目指す

コア毎のタスクスケジューリング

- リアルタイム性の確保が(比較的)容易
- タスクを移動させるAPIを追加
- 負荷変動への対応が可能
 - マイグレート可能なタイミングは要検討

API(ユーザー)による移動

ポリシーとメカニズムの分離



TOPPERS/FMPカーネル：開発状況

開発状況ステータス

- ASPカーネルをベースに開発
- 2009年5月にオープンソースとして一般公開

サポートプロセッサ

- ARM社 MPCore, ALTERA社 NiosII
- ARMプロセッサの命令セット シミュレータ SkyEye
 - リリースには含めないがサポート済み
 - 東芝 MeP, ルネサス SH-X3, SH2A-DUAL

TLVとの連携

- 実行トレースをTLVで可視化可能

TraceLogVisualizer (TLV)

開発の背景

マルチコア環境でのデバッグ

- マルチコア環境では各コアが独立に並列動作
 - × ブレークポイントやステップ実行を用いたデバッグが困難
 - 実行後のトレースログの解析によるデバッグが有効

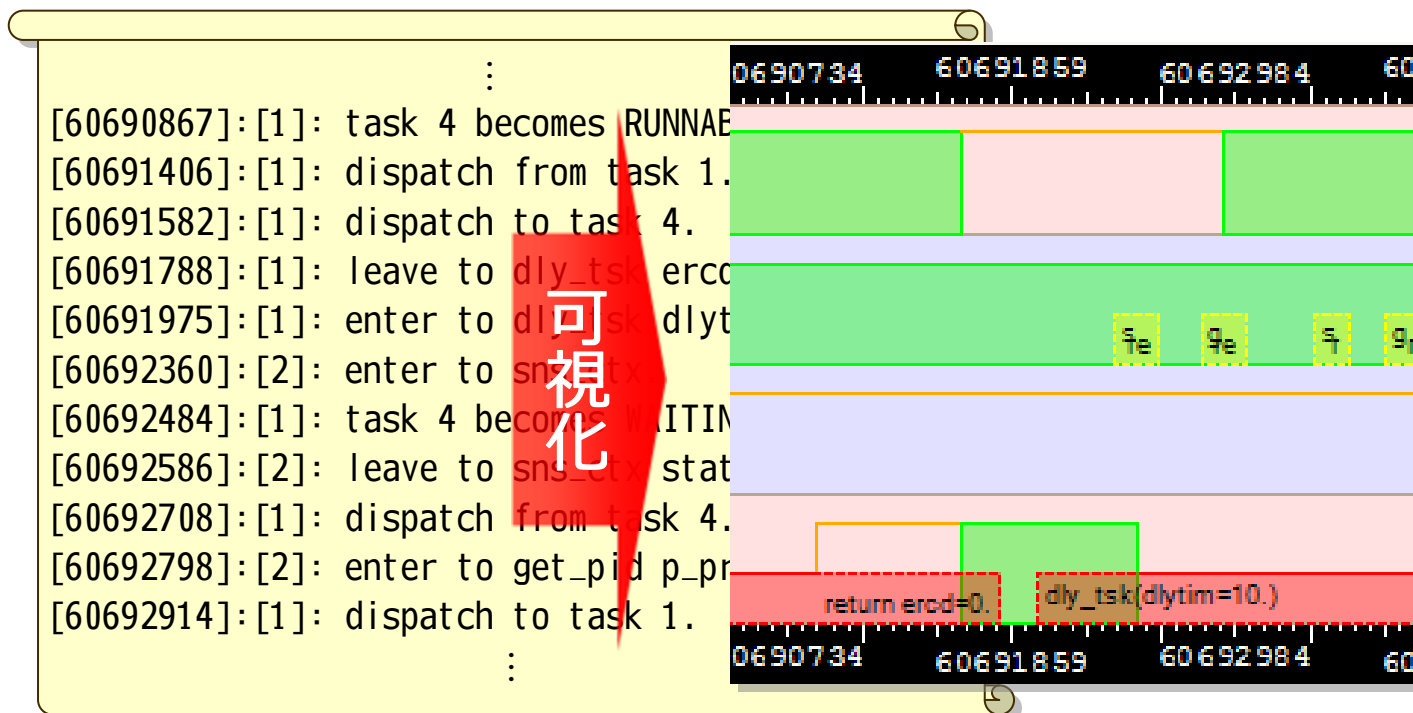
トレースログの解析によるデバッグ

- RTOSやシミュレータ、エミュレータなどが出力するトレースログを解析することによって動作を確認する
- 開発者がトレースログを直接扱うのには限界がある
 - × トレースログのサイズや処理の複雑さによっては解析不可能
 - トレースログの解析を支援するツールの要求

→ トレースログを可視化表示するツールの開発へ

2コア上で動くRTOSのトレースログの例

- ×時系列に各コアの動作が分散
- ×膨大な量
 - ↳約1msの間に11個のログ



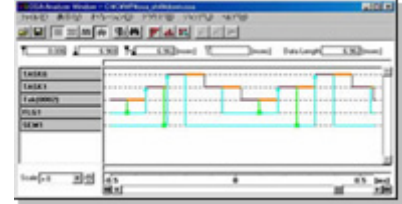
→トレースログを可視化表示し解析を支援

既存の可視化ツール

組み込みシステム向けデバッグソフトウェア

- PARTNER-Jet イベントトラッカー
- WatchPoint OSアナライザ

WatchPoint OSアナライザ
<https://www.sophia-systems.co.jp/ice/products/watchpoint>



組み込みシステム向け統合開発環境

- EvenTrek
- QNX System Profiler

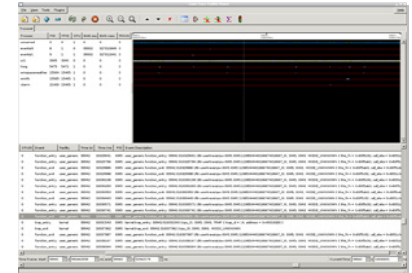


EvenTrek
http://www.esol.co.jp/embedded/eb_multicore2.html

Unix系OSのトレースログプロファイラ

- Dtrace-Chime (Solaris)
- LTTV(Linux)

LTTV
<http://ltt.polymtl.ca/screenshots/>



✗ ログの形式が標準化されていない

➔ 汎用性に乏しい

✗ 可視化表示項目が提供されているものに限られる

➔ 拡張性に乏しい

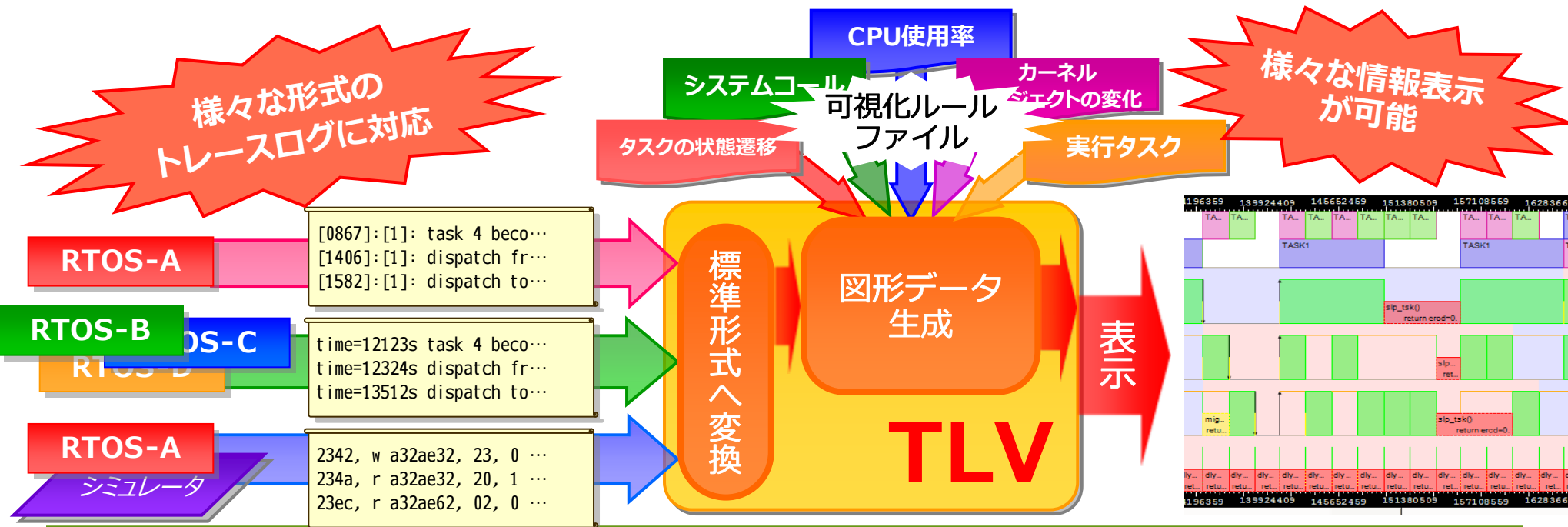
TraceLog Visualizer (TLV)

トレースログを可視化表示するツール

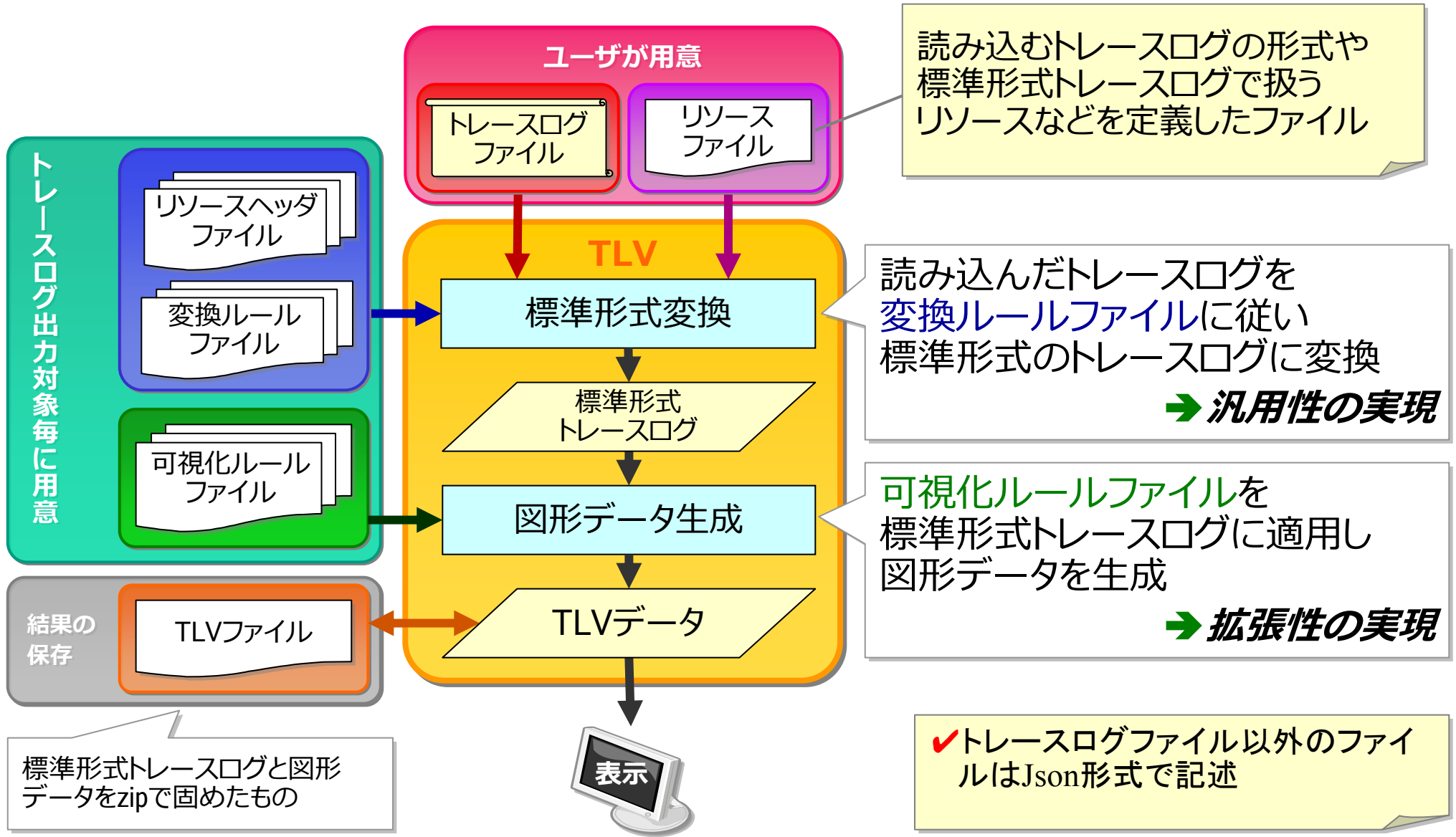
開発目標

- 汎用性 ... ログの形式に非依存化
- 拡張性 ... 可視化表示項目をプラグイン化

✓ 2009年5月TOPPERSプロジェクトからリリース



TLV全体像



今後の取り組み

TraceLogVisualizer (TLV)

- TOPPERSプロジェクトのメンバーからのフィードバックを反映し、一般公開に向けて機能や品質を向上を目指す

TOPPERS/FMPカーネル

- 組込システム向けのマイグレーション機構の研究・開発
- 実アプリケーションを用いた評価
- マルチプロセッサRTOSの検証技術
 - *コンソーシアム型研究共同研究で実施(後述)*
- 保護機能との統合