

## パネル討論

「マルチコア、ネットワーク・オン・チップのもたらすもの-  
複雑化による落とし穴と  
性能・機能・ディペンダビリティ向上のチャンス」

佐藤三久

CREST「ディペンダブルOS」領域

「省電力でディペンダブルな組込み並列システム向け計算プラットフォーム」

(筑波大・ルネサステクノロジ)研究代表者

／筑波大学 計算科学研究センター センター長

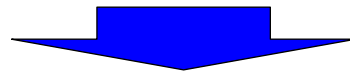
# CREST「ディペンダブルOS」領域の概要

## ■ オープンシステムディペンダビリティとは

組込みシステムは不完全さと不確実さに起因する未来に障害となりうる要因（開放系障害要因）を宿命的に抱えており、それが顕在化した時のシステムへの影響をマネージして（顕在化前もしくは顕在化後に適切に対応し、影響を最小とするように）、利用者が期待する便益を出来る限り、安全にかつ継続的に提供すること

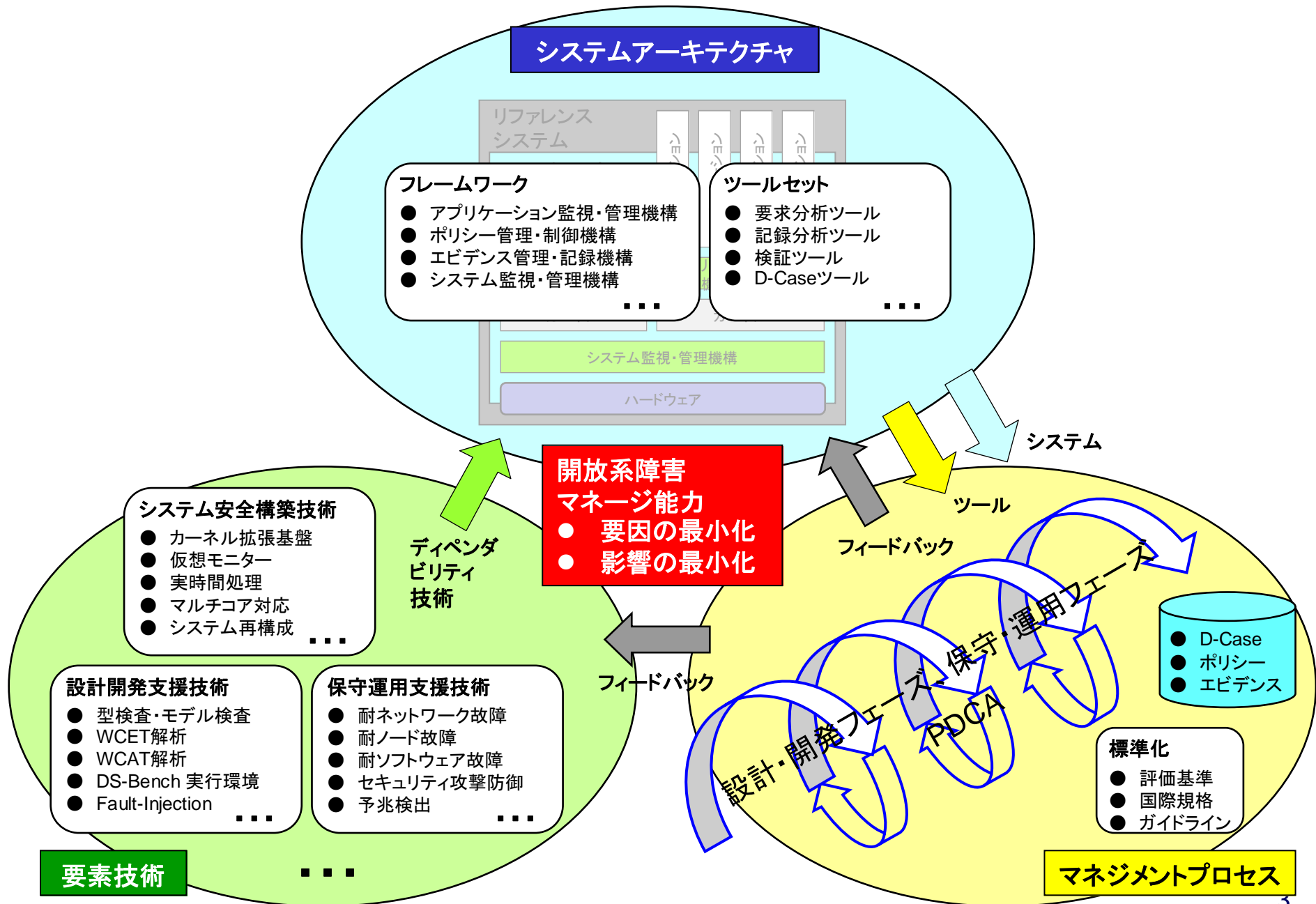
## オープンシステムの問題点

- 仕様実装の不完全さ、システムの完全理解の困難さ
- 使用状況の不確実さ、システムの挙動予測の困難さ



- ◆ 不完全さと不確実さを合わせ持っている（開放系の特質）

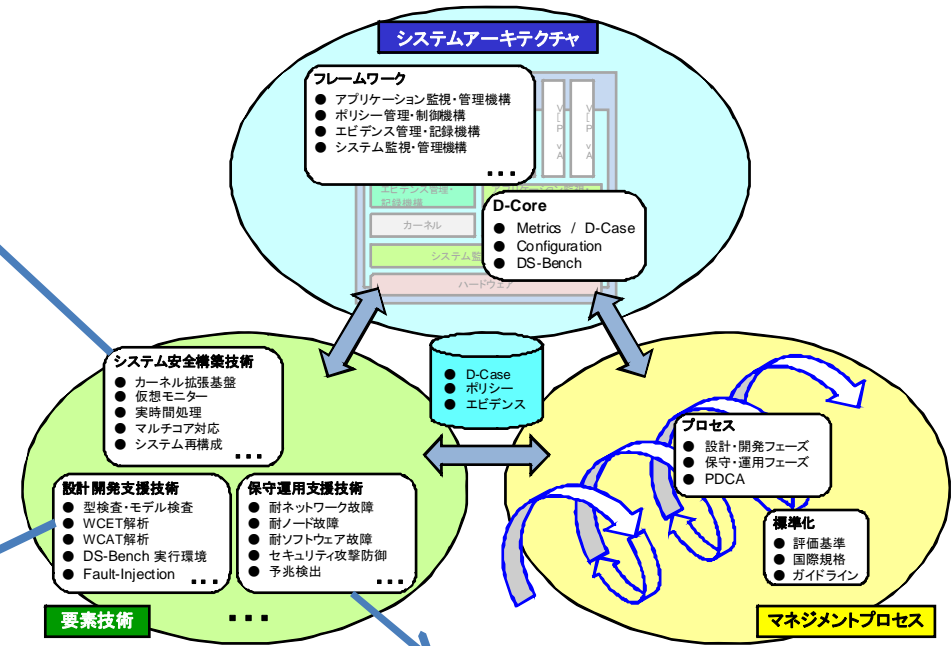
# オープンシステムディペンダビリティ実現の仕組み



# DEOSの全体構想

## システム安全構築技術

開発モジュール	チーム名
仮想モニタ モニタリング(VMO) マルコア制御(VMC)	中島
P-Bus Core 論理分割(LPAR)	石川



## 設計開発支援技術

開発ツール	チーム名
型検査・モデル検査(TCHK/MCHK)	前田
最悪実行時間予測(RETAS)	石川
電力使用量予測(GREEN)	徳田
Fault Injection (D-Cloud)	佐藤
ハイベンダブルシステムベンチマーク実行環境(DS-Bench)	石川

## 保守運用支援技術

開発モジュール	チーム名
動作時間予約機構(TR) 耐故障ネットワーク機構(SCTP+FHO)	徳田
耐故障ネットワーク機構 (RI2N/PEACH)	佐藤
アカウントティング機構(ACT)	中島/センター
シングルIPアドレス機構(SIAC)	石川

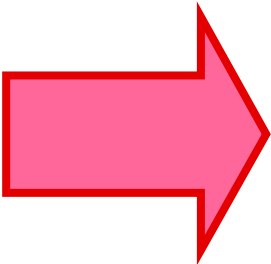


## 「ディペンダブル並列システム」に要請されるネットワーク技術

- 並列システムによる高性能化が可能であること
  - 並列処理による性能向上を達成するための十分な性能
- 並列システムの信頼性の向上をサポートすること
  - 冗長性(プロセッサ・機器)による信頼性確保
  - 一部が壊れていても、動作できる柔軟性
  - プロセッサだけでなく、入出力機器についても信頼性を確保できること

### ■ 省電力性

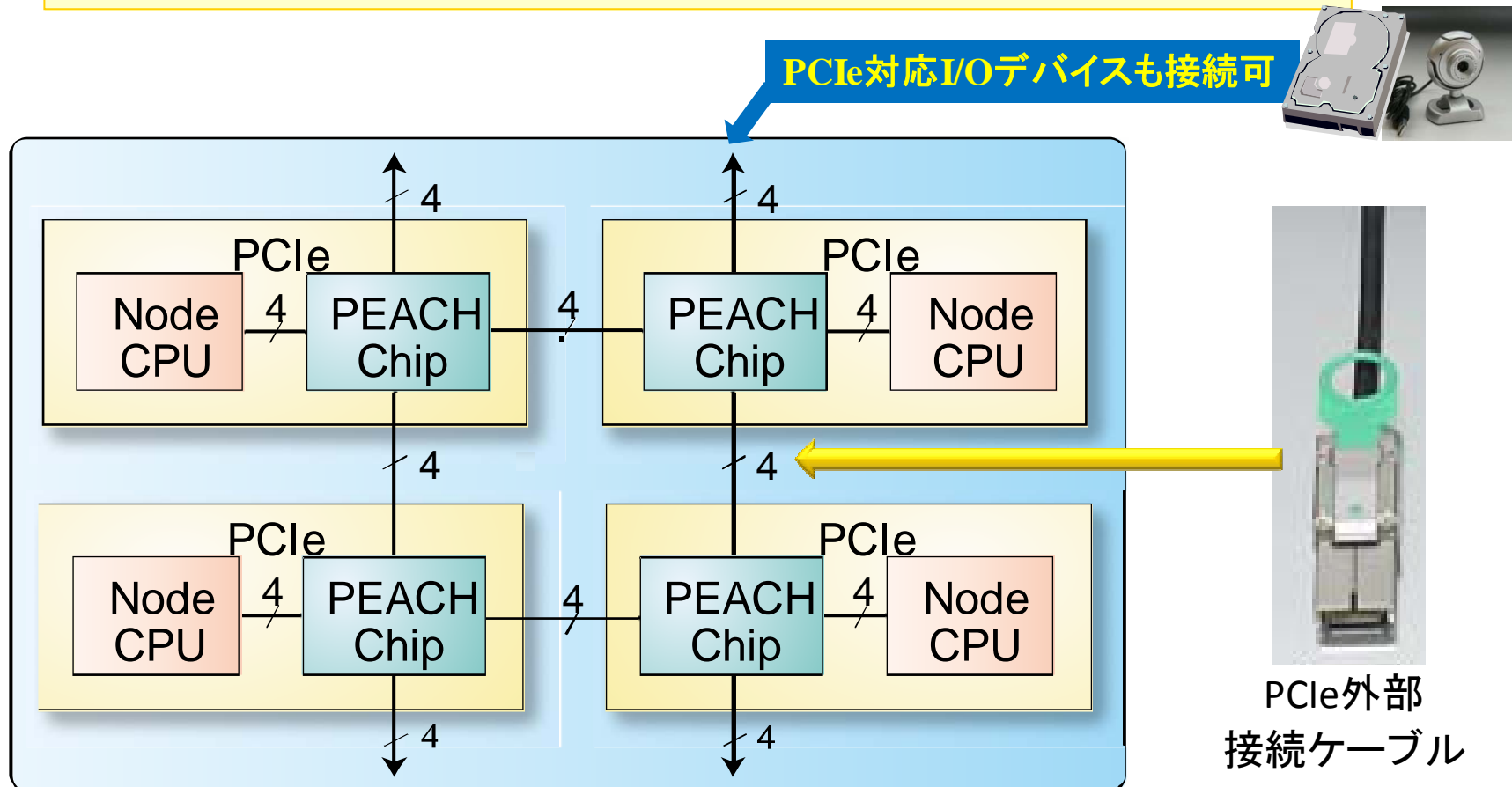
オープンシステムディペンダビリティを  
サポートする場合にも、システム内の  
ハードウェア障害に対処する運用時の  
信頼性を確保するのは重要

- 
- RI2N (Redundant Interconnection with Inexpensive Network)
    - Ethernetによるマルチリンク技術(統合デモ)

- PEARL (PCI-Express Adaptive and Reliable Link)
  - PCI-Express Gen2を近距離ノード接続に利用、PEACHチップを開発中

# PEARL (PCI Express Adaptive & Reliable Link)

- ノード間をPCI Expressで直接接続
  - PEACH (PCI Express Adaptive Communication Hub) チップ
  - PCI外部接続ケーブルを用いてノード間を接続(数mまで)
  - 入出力デバイスも接続可能





## 「マルチコア、ネットワーク・オン・チップのもたらすもの-複雑化による 落とし穴と性能・機能・ディペンダビリティ向上のチャンス」

- マルチコアって、組み込みでは本当に使われているのか？
  - こちらのCRESTの「推進委員」の方々からはあまり、聞こえてこない。
  - コアごとに、機能分担をするということでは使われている(か?)
    - 別々のチップになっていたものを、一つのチップに集約
    - 特定機能の強化(動画処理、...)
  - これが、期待される使われ方か？
    - 本来は性能向上(低電力/省電力)でも使いたい。
- 何が問題か？もちろん、機能は重要だが、...
  - APIが標準化されていない。
    - 共有メモリであっても、完全な共有メモリではない(共有メモリハードウェアは高価)
    - チップ内のリンクに(まだ)標準がない。
  - 標準的なAPIがないと、ソフトウェアの資産が蓄積しない
    - ソフトウェアとの協調もしにくい



# 「マルチコア、ネットワーク・オン・チップのもたらすもの-複雑化による 落とし穴と性能・機能・ディペンダビリティ向上のチャンス」

## ■ 提案：ソフトウェアと協調するためには、標準的なインタ フェースの共同して策定をするべき

- 「標準化」がソフトウェア開発、プログラミングモデルの確立を促進する
  - 例：ハイエンドのHPCにおけるMPI(1995-), OpenMP(1997-)
  - 例：GPGPUのCUDA, OpenCL
  - ... 教育も。
- ソフトウェアによるディペンダビリティ機能との協調を可能にする
  - checkpoint/Fail-over
  - アプリケーションレベルのrecovery
- ハードウェア(アーキテクチャ)へのソフトウェアからのフィードバックも
  - 「このような機能がほしい」とか
- 現在、我々のリンクのハードPEARLについては、MCAPIを提供
  - MCAPI: Multicore Communications API
    - <http://www.multicore-association.org/workgroup/mcapi.php>
    - 本来はチップ内のリンクにも適用？
  - 実際の実装、XMCAPIを進めている