平成 21 年度
実績報告

「実用化を目指した組込みシステム用
ディペンダブル・オペレーティングシステム」
平成18年度採択研究代表者

中島　達夫

早稲田大学　理工学術院・教授

高機能情報家電のためのディペンダブルオペレーティングシステム

§1. 研究実施の概要

本研究チームでは，以下に示す2つの面から仮想化を利用した情報家電機器向けのディペンダブル OS の構築に取り組む．

平成21年度は1つ目の仮想化技術に関しては，SH4a を利用したマルチコアプロセッサ上でハードウエア仮想化層の構築をおこなった．現状では，Linux，Toppers(ITRON 仕様を実装するオープンソース RTOS)，L4 マイクロカーネルが仮想化層の上で動作可能である．実際に，3.5 個の CPU を利用して SMP Linux を動作させ，0.5 個の CPU を利用して Toppers を動作させるデモを作成し，マルチコアプロセッサ上のハードウエア仮想化層を利用する有効性を検証した．また，ハードウエア仮想化層自体の信頼性向上がオープンディペンダビリティの実現に重要であることが明らかになったため，マルチコアプロセッサの冗長性を利用するシステムデザイン法を考案した．

2つ目の仮想化技術に関しては，SH4a プロセッサ用に実装されたハードウエア仮想化層上の高信頼 RTOS 上はモニタリングサービスを動作するようにした．現状では，システムコールディスパッチャの監視，スケジューラを実現するデータ構造の監視，カーネルローダブルモジュールの監視（特に P-Component の監視）を実現している．モニタリングサービスにより監視しているデータ構造の一貫性が失われた場合，モニタリングサービスは出来る限りデータ構造の一貫性を修復しようとする．しかし，修復に失敗した場合や，一貫性の崩壊によりシステムの実行の続行が危険な場合は，Linux カーネルを能動的に再起動するようにした．

§2. 研究実施体制

(1)「早稲田大学」グループ
①研究分担グループ長：中島　達夫（早稲田大学、教授）
②研究項目　仮想化を利用したディペンダブルOSの構築

(2)「追川」グループ

①研究分担グループ長:追川　修一(筑波大学、准教授)

②研究項目　組込みシステムに適したアイソレーション機能の研究

§3. 研究実施内容

(文中に番号がある場合は(4－1)に対応する)

本研究チームでは，以下に示す2つの面から仮想化を利用した情報家電機器向けのディペンダブル OS の構築に取り組む.

1つ目の仮想化技術は，プロセッサを仮想化するためのハードウエア抽象化層の提供である. 本研究では，プロセッサ仮想化は2つの役割を担当する. 1つ目は，モニタリングサービスを実行するための基盤環境の提供である. モニタリングサービスの整合性を保証するため，モニタリングサービスは Linux の外部で実行する. P-BUS が型非安全なエラーや攻撃による障害が発生しないことを保証するので，Linux カーネル外のメモリへのアクセスはソフトウエア的に禁止される. そのため，プロセッサ仮想化は原則として OS 間の隔離機能を提供する必要がないので，リアルタイム性をそこなうことなく性能が高い仮想化機能の実現が可能となる. 2つ目は，マルチコアプロセッサへの対応である. 本研究が提供するプロセッサの仮想化は，システム全体の負荷に応じて，適切な数のコアの利用を可能とする. 特に，情報家電機器は，使用するときと使用しない時の負荷が大きく変化する. そのため，システムの負荷が低い場合は，起動するプロセッサコアの数を最小限にすることにより，消費電力量を出来るだけ少なくすることが好ましい. 本研究により開発するプロセッサ仮想化は，ゲスト OS が使用するプロセッサを仮想プロセッサとして扱うことを可能とすることにより，物理的に利用するコア数を動的に変化することを可能としている.

現在，SH4a を利用したマルチコアプロセッサ上でハードウエア仮想化層は動作している. 現状では，Linux，Toppers(ITRON 仕様を実装するオープンソース RTOS)，L4 マイクロカーネルが仮想化層の上で動作可能である. 実際に，3.5 個の CPU を利用して SMP Linux を動作させ，0.5 個の CPU を利用して Toppers を動作させるデモを作成し，マルチコアプロセッサ上のハードウエア仮想化層を利用する有効性を検証した. また，ハードウエア仮想化層自体の信頼性向上がオープンディペンダビリティの実現に重要であることが明らかになったため，マルチコアプロセッサの冗長性を利用するシステムデザイン法を考案した.

2つ目の仮想化技術はモニタリングサービスを利用したLinuxカーネル内のエラーの仮想化である. Linux カーネルはソフトウエアのバグや外部からの悪意のある攻撃により，カーネル内のデータ構造の一貫性が失われる可能性がある. DEOS では，P-BUS が提供する検証機能により型非安全なエラーが発生しないことが保証されているが，型安全なエラーへの対応は P-BUS では不十分である. 本研究が提供するエラーの仮想化機能では，データ構造の一貫性制約ルールを定義し，その一貫性が失われたときに，エラーからの修復をおこなうことにより，データ構造の一貫性が保たれることを保証する. それにより，仮想的にエラーの発生が起きなかったように見せることを可能

とする．ここでは，データ構造の一貫性修復のことをエラーの仮想化と呼ぶ．データ構造の一貫性修復は，データ構造を完全にロールバックすることを前提としていない．前方回復と後方回復の両方を利用可能とすることにより，回復可能性を向上することを目指している．Linux にはガーベジコレクション機能がないため，前方回復を利用した場合は，一部のデータを解放できないケースも存在する．例えば，rootkit による攻撃の場合は，最終的に rootkit を消去するためには，システムを再起動する必要があるケースも存在する．そのため，本研究では，下に述べるハードウエア抽象化層と連動させることにより，システム全体の一貫性を回復するために定期的な再起動を利用する．

現在，SH4a プロセッサ用に実装されたハードウエア仮想化層上の高信頼 RTOS 上はモニタリングサービスは動作している．現状では，システムコールディスパッチャの監視，スケジューラを実現するデータ構造の監視，カーネルローダブルモジュールの監視（特に P-Component の監視）を実現している．モニタリングサービスにより監視しているデータ構造の一貫性が失われた場合，モニタリングサービスは出来る限りデータ構造の一貫性を修復しようとする．しかし，修復に失敗した場合や，一貫性の崩壊によりシステムの実行の続行が危険な場合は，Linux カーネルを能動的に再起動する．

我々が構築する仮想化技術は，大きく5つのコンポーネントから構成される．1つ目のコンポーネントはハードウエア抽象化，2 つ目は，モニタリングサービス，3 つ目は，再起動管理システムを実装した高信頼 RTOS，4 つ目はロギングサービス，5 つ目はアイソレーション機能である．前半の4つのコンポーネントに関して早稲田大学グループが研究をおこない，最後のアイソレーション機能は筑波大学チームが担当している．また，筑波大学チームは本チーム全体が開発した仮想化技術の有効性を示すためのアプリケーションとして遠隔監視・管理システムの構築もおこなっている．現状では，各コンポーネントのプロトタイプを作成し，実装上の問題点を明確にした．その結果を踏まえて，来年度以降はシステム全体の再実装をおこなうことにより，より実用化レベルに近いシステムの開発をおこなう．また，開発したシステムの可能性を検討し，情報家電機器だけではなく，より広いアプリケーションドメインへの適用を検討する．特に，システムがオープンシステムディペンダビリティを支援することにより可能となる新しいアプリケーションドメインを開拓し，情報技術の可能性を拡大することを検討する．

§4. 成果発表等

(4－1) 原著論文発表

- 論文詳細情報

1. Ki-duk Kwon, Midori Sugaya and Tatsuo Nakajima, Analysis of Embedded Kernel using Kernel Analysis System, The 6th International Conference on Embedded Software and Systems, pp. 417-422, 2009 年 5 月

DOI: http://doi.ieeecomputersociety.org/10.1109/ICESS.2009.74

2. Midori Sugaya, Yuki Ohno, and Tatsuo Nakajima, Lightweight Anomaly Detection System with

HMM Resource Modeling, International Journal of Security and Its Applications, pp35-54, Vol. 3, No. 3, 2009 年7月

3. Andrej van der Zee, Alexandre Courbot, Tatsuo Nakajima, ″mBrace: Action-based Performance Monitoring of Multi-Tier Web Applications″, In Proceedings of The 2009 International Conference On Embedded and Ubiquitous Computing(EUC 2009), pp.166-173, 2009 年 8 月

DOI: http://doi.ieeecomputersociety.org/10.1109/CSE.2009.219

4. Yuki Kinebuchi, Kazuo Makijima, Takushi Morita, Midori Sugaya, Tatsuo Nakajima, ″CONSTRUCTING MULTI-OS PLATFORM WITH MINIMAL ENGINEERING COST″, International Embedded Systems Symposium 2009, pp.195-206, 2009 年 9 月

DOI: 10.1007/978-3-642-04284-3_18

5. Lei Sun, Yuki Kinebuchi, Tomohiro Katori, Tatsuo Nakajima, Runtime Self-Diagnosis and Self-Recovery Infrastructure for Embedded Systems, International Conference on Self-Adaptive and Self-Organizing Systems, pp.284-285, 2009.年 9 月

DOI: http://doi.ieeecomputersociety.org/10.1109/SASO.2009.21

6. Lei Sun, Hiromasa Shimada, Tatsuo Nakajima, Reusable Integrity Management Services for Embedded Systems,The 2009 IEEE International Conference on Service-Oriented Computing and Applications (SOCA'09), pp.65-72, 2009 年 12 月