

ES 7000 によるミッション・クリティカル Linux の実現

Implementation of Mission Critical Linux on ES 7000

高橋 秀樹

要約 カーネル 2.6 のリリースにより、Linux はエンタープライズ・システム向けの OS として急速に普及しつつある。本稿では、Unisys のエンタープライズ・サーバ ES 7000 をメインフレーム IA サーバと位置付け、技術的な裏付けを元に ES 7000 Linux によるミッション・クリティカル・システムの実現性を検証している。まず RASIS の観点からカーネル 2.4.9 とメインフレームの性能比較を行い、更にカーネル 2.6 の強化機能による効果を予想することで現時点でのメインフレームとの性能差を比較している。次に、ES 7000 本体の実力をハードウェアとソフトウェアの両面から検証し、ES 7000 Linux におけるミッション・クリティカル要件を提示している。Linux によるミッション・クリティカル・システムは、すでに構築可能な範囲にあり、その実現に向けてソフトウェア・プロダクトの保守性の改善と Linux の高可用性をアピールしていく必要がある。

Abstract Linux has begun to be spread to users rapidly since Kernel 2.6 was developed and released for enterprise systems. In this paper, we position Unisys Enterprise Server ES 7000 as a mainframe IA server and the feasibility of the Mission Critical System with ES 7000 Linux is examined under technological grounding. First, functionalities of Kernel 2.4.9 and mainframe OS are compared from viewpoint of RASIS and then the capabilities differences between Kernel 2.6 and mainframe OS are examined where the newly enhanced functions on Kernel 2.6 is in our expectation based on our analysis of Kernel 2.6. Then actual ability and performance of ES 7000 hardware and software are examined for defining the mission critical requirements for ES 7000 Linux. The mission critical system on Linux could be implemented soon with the improved maintainability of software, and high availability of Linux should be appealed to customers for further progress.

1. はじめに

サーバ・ベンダ各社が Linux ビジネスへの参入を本格化する中、多くの企業は Linux ビジネスで十分な利益を確保できていない。無料で入手可能なオープンソース・ソフトウェア（以下、OSS という）の世界において、利益を生み出すためのビジネス・モデルは未だ確立されていないというのが現状である。

こうした中で、2004 年 8 月 2 日、米国 Unisys はエンタープライズ Linux 分野への本格参入を表明した。Unisys の 118 年間にわたる歴史がメインフレーム技術と Intel チップ技術をバランス良く組み合わせることで誕生した ES 7000^{*1} は、IA サーバがスケールアップ可能なプラットフォームであることを世界で初めて実証したメインフレーム・クラスの高可用性 IA サーバである。OSS の代表格である Linux を ES 7000 のオペレーティング・システム（以下、OS という）として採用し、ES 7000 Linux^{*2} によるミッション・クリティカル・システムを実現できれば、Unisys グループにおける Linux ビジネスの核として、確実に利益をもたらす一つのビジネス・モデルに育てることができる。更にそれは、OSS の普及・推進につながり、Linux

ビジネス全体の拡大と新しいビジネス・モデルへの展開が期待できる。

本稿では、ES 7000 をメインフレーム IA サーバと位置付け、技術的な裏付けを元に ES 7000 Linux の性能を明らかにし、ES 7000 Linux によるミッション・クリティカル・システム^{*3}の実現性を検証する。

2. メインフレームとの RASIS 比較

弊社では、2003 年度に「Enterprise Linux 実現プロジェクト」を立ち上げ、プラットフォーム側から見たエンタープライズ・システム^{*4}要件の調査を行った。この調査は、プラットフォームとして弊社がサポートしている Linux、メインフレーム、Unix および W 2 K を選択し、RASIS^{*5}の観点から各プラットフォーム間の詳細比較を行ったものである。

本章では Linux とメインフレームの比較に重点を置き、上記調査結果を参考にしながら Linux OS の実力を検証する。

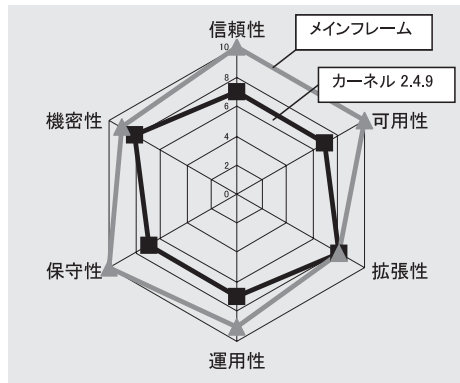


図1 カーネル 2.4.9 とメインフレームの RASIS 比較

2.1 カーネル 2.4.9 における RASIS 評価

図1は、2003年度に実施した RASIS 調査の比較結果からカーネル 2.4.9 と弊社メインフレームのデータを抜き出して作成したレーダチャートである。

カーネル 2.4.9 を使用した ES 7000 Linux と弊社の最新メインフレームを比較した場合、拡張性を除くすべての項目（信頼性、可用性、運用性、保守性、機密性）において Linux はメインフレームに劣っていることを示す調査結果になっている。表1は各項目の評価点とその理由をサマライズしたものであるが、カーネル 2.4.9 においては、ES 7000 Linux によるミッション・クリティカル・システムの構築が非現実的であったことがわかる。

2.2 カーネル 2.6 における RASIS の変化

2003年12月にリリースされたカーネル 2.6 の主な強化機能項目については雑誌等の特集記事で多数紹介されているため、本稿では各機能の詳細説明は割愛するが、各機能が貢献している RASIS エリアとミッション・クリティカル・システムへの貢献度という観点でカーネル 2.6 の主な機能を表2にまとめてみた。カーネル 2.6 の RASIS 改善エリアを明確にするために、性能ごとの機能数も示している。

図2は、表2の結果を踏まえ、各強化機能の効果を予想して導出したものである。カーネル

表1 カーネル 2.4.9 とメインフレームの RASIS 比較評点と主な理由

項目	評点	カーネル 2.4.9 (ES7000 Linux)	評点	メインフレーム(Unisys 最新機)
信頼性	7	①日本での稼働実績がない ②オンラインダイアグ機能等, 多くの点で機能的にメインフレームに見劣りする	10	①システムとして高い信頼性実績がある ②H/W リカバリが設計書上で明確に定義されており, データの完全性がシステム設計時に検討され, インプリメントされている
可用性	8	①動的再構成ができず, 稼働率もメインフレームに見劣りする ②対障害性に対する機能が弱い	10	①システムとして高い稼働率実績がある ②障害時のシステム再構成や動的再構成が可能
拡張性	8	①64bit 対応による拡張性がある	8	①現在も次期メインフレームを開発中であり, H/W 性能の向上に合わせた実績のある拡張性が維持されている
運用性	7	①電源管理ができていない ②システム構成の動的 UP/DOWN は周辺機器のみ可能 (CPU, メモリは不可)	9	①H/W リソースの動的な分割に優れる
保守性	7	①障害対応は回避策提示までは可能だが, 修正はディストリビュータと協議が必要 ②Dump 採取, 各種ログ採取機能が不十分	10	①システムとして高い保守実績がある ②全面的に自己完結型の対応が可能 ③プロダクトを完全に把握している
機密性	8	①機能的にはメインフレームに見劣りしないが, B1 レベルの認証*を受けていない ②メインフレームよりも攻撃対象になりやすい	9	①B1 レベルの認証を受けている ②ユーザインタフェース/管理機能は劣るものの, 客先管理体制は強固

(評点 : 各性能項目ごとに八つの詳細項目を設定し, 各詳細項目を 10 段階評価して求めた平均点)

* B1 レベルの認証 : 米国防総省が 1983 年に制定したセキュリティ評価基準 TCSEC のレベルの一つ . D, C1, C2, B1, B2, B3, A1 の順で高いセキュリティ要件が定義されている . A1 が最高要件 .

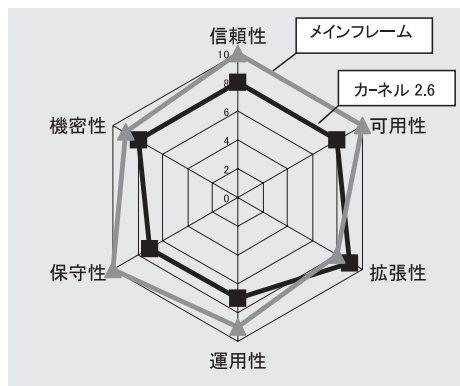


図2 カーネル 2.6 とメインフレームの RASIS 比較

2.6 は, エンタープライズ用途を強く意識し, 拡張性の改善を最大の開発目標としてきた . 品質向上のためにオーバヘッドを含む改善もあるが, 結果的に効率の改善に大きな効果を発揮すると考えられる . 一方, 保守性における強化機能は 0 件であり, 大きな改善が見られない点を特筆しておきたい . 各ディストリビュータは, LKCD, netdump 等のダンプ採取機能を独自に取り込み, 提供しているが, ミッション・クリティカル用途を考慮した保守性の改善が今後の重要な改善ポイントと言えるだろう .

RASIS の観点でカーネル 2.6 を評価した場合, メインフレームの信頼性, 可用性, 運用性, 保守性においてまだ大きなギャップが見られる . ただし, カーネル 2.4 からカーネル 2.6 への進化のスピードは予想を上回るものであり, 今後これらのギャップが急速に縮まっていく可能性は高い . この状況は, Windows® 2000 Datacenter Server が, 「ブルー・スクリーン停止」

表2 Linux カーネル 2.6 の主な機能強化項目

No	機能	機能内容	信頼性	可用性	拡張性	運用性	保守性	機密性	効率性	ミッション・クリティカル貢献度
1	Intel HW 対応	ハイパー・スレディング対応の強化							●	○
2	NUMA 対応強化	Non-Uniform Memory Access 対応			●				●	○
3	ロック処理の改善	ジャイアントロックの廃止			●				●	◎
4		RCU 排他制御/Sequence Counter 制御			●				●	○
5	割り込み処理改善	割り込み遅延処理の並列化			●				●	◎
6	スレッドの改善 (POSIX 対応強化)	NPTL (Native POSIX Thread Library) のサポート/プロセス間同期機構(Futex)の強化			●				●	○
7		PID の最大数が 32K から 10 億に増加			●					○
8	スケジューラの改善	O(1)スケジューラの導入	●		●				●	◎
9		CPU アフニティ制御	●						●	○
10		カーネル・プリエンブション対応	●						●	○
11	I/O 処理の改善	ブロック I/O 処理の性能向上	●		●				●	◎
12		ファイル I/O 処理の並列度/応答性向上	●		●				●	◎
13		デバイス I/O スケジューリングの強化			●				●	○
14		ダイレクト I/O のサポート							●	○
15		非同期 I/O のサポート							●	◎
16		ネットワーク負荷対策(割り込み+ポーリング)	●	●						○
17		マルチパス I/O	●	●					●	◎
18	メモリ管理処理の改善	メモリ開放処理の安定化 (リバース・マッピング, 開放ページの予約機能等)	●	●	●					◎
19		ラージ・ページ (Huge TLB 対応)	●	●	●					◎
20	ファイル・システムの改善	JFS (Journaled File System) のサポート			●					○
21		動的ファイル・システム拡張対応			●					○
22		NFSv4 のサポート (複合 RPC サポート, Lock 機構の改善, NFS over TCP サポート等)			●					○
23		ネットワーク負荷の削減 (Zero Copy NFS)			●				●	○
24		CIFS (Common Internet FileSystem) サポート								△
25	ネットワーク機能の強化	IPv6 の強化			●					○
26		SCTP/IPSec のサポート	●	●					●	○
27		IP ロードバランス機能 (IPVS) のサポート			●					○
28	電源管理の強化	ACPI の強化	●			●				○
29	セキュリティ機能の強化	LSM (Linux Security Module) の導入							●	◎
30		POSIX ACL のサポート							●	○
31		カーネル内暗号化処理の強化							●	◎
32	無停止 H/W 交換	PCI ホットプラグ対応 H/W のサポート	●	●		●				◎
33	デバイスサポートの拡張	サポートデバイスの増強 主な数値制限を 255 から 4095 に変更			●					◎
34	組込み向け強化	μCLinux, プリエンプ機能のサポート	-	-	-	-	-	-	-	△
RASIS への貢献機能数			12	11	16	3	0	4	17	-

(ミッション・クリティカル貢献度: ◎= 必須機能, ○= 必要機能, △= ディスクトップ/組込み向け機能)

に代表される Windows NT の信頼性を改善し、多数の大規模システム向け機能を開発した結果、Windows 環境でのエンタープライズ・システムが企業に広く浸透していった状況に良く似ている。カーネル 2.6 の登場は、Linux によるエンタープライズ・システムの構築を大きく加速すると考えられる。

3. ES 7000 Linux のミッション・クリティカル性能

本章では ES 7000 本体に着目し、ES 7000 のミッション・クリティカル業務に対する性能をハードウェア、ソフトウェアの両面から評価してみる。

3.1 ハードウェアのミッション・クリティカル性能

ES 7000 は、ES 7000/500 シリーズ*6 (IA 32) と ES 7000/400 シリーズ*7 (IPF (Itanium®

Processor Family 64 ビット) の二つのシリーズが提供されているが、本稿では現在の Intel アーキテクチャの主流である IA 32 マシンを中心に解説する。IPF マシンは、32 ビット命令をエミュレーションによって実行するため、64 ビットに未対応のアプリケーションに対しては IA 32 マシンよりも性能が劣る場合があり、まだ当初の予想ほど普及していない状況にある。

3.1.1 ES 7000/500 シリーズの RAS (信頼性/可用性/保守性) 機構

ES 7000 は、単一障害ではシステム停止が起こらないように設計されており、CMP (セルラー・マルチプロセッシング) アーキテクチャを採用したマシンである。「セルラー(セル方式)」とは、モジュール化設計を意味する。

図 3 に ES 7000/500 シリーズにおけるセルの主要部分を示す。セルは、最大で 8 個の Intel Xeon MP プロセッサ、64 MB の TLB(4 次キャッシュ)、16 GB の主メモリ(MSD : Memory Storage Director)、八つの PCI スロット、二つのパワー・サブライ、更にファン、クロック等のコンポーネントを含んでおり、完全に独立したサーバになれるように実装されている。各セルには、二重の AC コンセントと N+1 基の電源装置が用意されており、1 台の電源装置が障害を起こした場合でも、そのセルに関連したパーティションは通常の動作を継続できる。メイン・メモリと TLB には ECC メモリを実装し、全データ経路上のパリティ・チェックを行っている。Unisys がシステム・ヘルス・モニタ機能として開発したサーバ・ナビゲーション・ツールは、常時、システムの正常性を監視し、障害検知時には回復動作の起動、障害報告の記録、サポート・センタへのコール・ホーム要求の送信等を行う。MIP (管理インタフェース・プロセッサ) は、それ自身がコンピュータであり、ES 7000 本体の中に独立した電源装置を持つ。サーバ・ナビゲーション・ツールの制御の下で稼働しながら、主要なハードウェア・コンポーネントを監視し、障害発生時には必要に応じて自動的にパーティション・リブート等を起動する。MIP は、独立したコンピュータとしてパーティションを管理するため、単なるスケールアウト型の IA サーバとは異なり、Linux や Windows との共存環境においても、システム全体の統合管理が可能となっている。

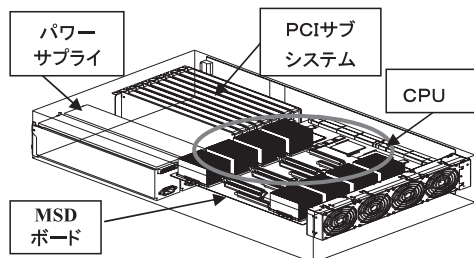


図 3 セルの主要部分

3.1.2 ES 7000/500 シリーズのハードウェア・アーキテクチャ

1) CMP (Cellular Multi Processing) アーキテクチャの特徴

ES 7000/500 シリーズは、CMP アーキテクチャを採用し、32 ビット・マシンとして高い優位性を保持しているマシンである。図 4 に ES 7000/500 シリーズのアーキテクチャ概略図を示す。CMP は、Unisys 独自のハードウェア・アーキテクチャであり、Intel チッ

ブを搭載しているが Intel のチップセットは使われていない。CMP は、カーネル 2.6 で機能強化が行われた NUMA (Non Uniform Memory Access) 技術とはまったく異なる技術であり、ここでは、NUMA 技術と CMP 技術の対比を行ってみたい。

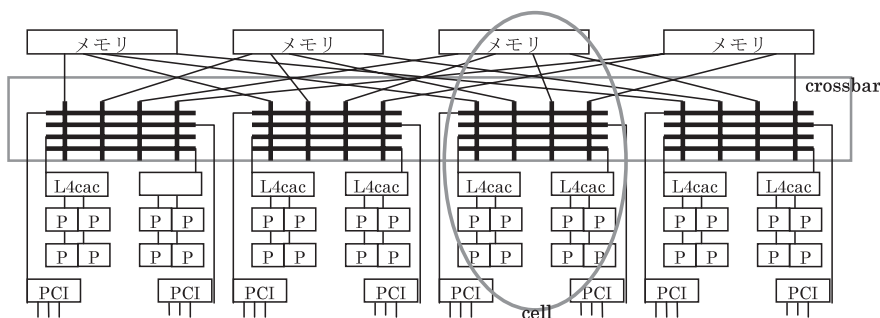


図 4 ES 7000/500 のアーキテクチャ概略図

CMP ハードウェアの最大の特徴は、最大 32 CPU までのマルチプロセッサ構成を SMP (Symmetric Multi Processing) 技術を採用して実現している点にある。ハードウェアの設計上、各 CPU からのメモリ・アクセスが均一であり、メモリ・アクセスにおいてローカル/リモートの区別がなく、文字通り UMA (Uniform Memory Access) で設計されている。UMA 概念図を図 5 に示す。

これに対し、IBM 社の xSeries® や HP 社の Superdome® といった最近の主流サーバは、NUMA ベースのハードウェア実装となっている。NUMA の概念図を図 6 に示す。NUMA ハードウェアは、CPU、メモリ、メモリ・コントローラ、I/O コントローラ等を 1 セットにして一つの箱型モジュールを形成する。メモリが各モジュール専用になり組み込まれるため、モジュール外の CPU からのメモリ・アクセスはリモート・アクセスとなり、ローカル・アクセスに比べ、アクセスタイムが大幅に増加する。(ES 7000 の設計当方で、7~10 倍のアクセスタイムが必要とされていた。)

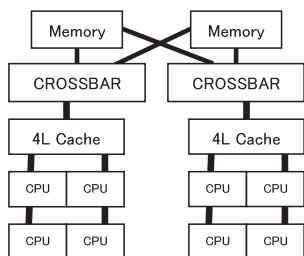


図 5 UMA 概念図

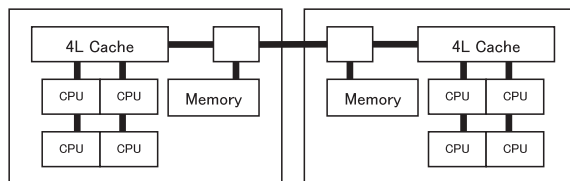


図 6 NUMA 概念図

Unisys が開発した CMP アーキテクチャも最近流行の NUMA アーキテクチャも従来の SMP マシンの課題であった CPU メモリ間のバス競合によるパフォーマンス低下の改善を目指したアーキテクチャである。NUMA のメリットは Intel 社のチップセットをそのまま採用でき、実装コストを大幅に引き下げられる点にあるが、NUMA は元来、高負荷トランザクション処理に向けたアーキテクチャとは言えない。

表3 CMP アーキテクチャと NUMA アーキテクチャの相違点

項目	CMP (Crossbar)	ccNUMA
サポート CPU 数	32 個	256 個
障害コンポーネントの切り分け	可能	可能
キャッシュ内容の一貫性	パラレル	ダイジェー・チェイン
メモリ間のインタリーブ	可能	不可能
CPU アフィニティ	単純	複雑
アプリケーション・チューニング	1 次元	多次元

表3に示しているように、CMPではキャッシュ内容の一貫性を保つためのインバリデーションをクロスバー経由でパラレルに実行できるのに対し、NUMAでは各キャッシュを順番にインバリデートしなければならない。CMPではメモリ・インタリーブにより同時に複数メモリからのデータ・フェッチが可能であるが、NUMAではメモリ・インタリーブをサポートできない。更に、CMPではCPUアフィニティ^{*8}処理が単純なのに対し、NUMAではリモートCPUからの最新キャッシュデータのリロード負荷が大きく、各プロセスをローカルCPU上で実行させるための複雑なNUMA対応ロジックが必要になる。

2) NUMA システムにおける最適化

CMPアーキテクチャは、メインフレームが採用してきたCPUとメモリ間のダイレクト接続の考え方をクロスバー技術を用いて発展させたものと考えてよい。CMPアーキテクチャでは、OSを垂直スケーリング用に最適化することで、CPU、メモリ、I/Oシステム間のバランスを確保し、高いシステム性能を発揮することができる。一方、NUMAアーキテクチャのシステムは最適化のための考慮点が多く、チューニングが難しいとされている。ここでは、NUMAシステムのチューニングについて検討してみたい。

大規模システムを要求するアプリケーションは、大きく分けて技術計算系アプリケーションとOLTP (OnLine Transaction Processing) 系アプリケーションの2種類に分類できる。技術計算系のアプリケーションは、タスクを分割し複数のCPUで並列処理させることが可能である。しかし、タスクをどのCPUに割り当てるかを決定する作業は、運用で対応するにせよソフトウェアに判断させるにせよ難しい作業となる。たとえば、あるCPUをメンテナンスのためにオフラインにしたり、逆にCPU数を増設したような場合、NUMAシステムではアプリケーションとCPUのバランスを再調整しなければならない。

OLTP系のアプリケーションは更にチューニングが難しい。トランザクションからの共有データに対するアクセスは、特定のノード (CPU) に集中することはなく、結果的に多くのメモリ参照がリモート参照となってしまう。また、時間帯や季節によるトランザクション量の変動に対して、最適なチューニングを自動的に管理することは極めて難しい。

NUMAアーキテクチャが多次元でのチューニングを必要とするのに対し、SMP (UMA) アーキテクチャを採用しているES 7000/500は、一次元でのチューニング (CPUのscale up/scale dn) で対応できる。これは、CMPアーキテクチャの優位性としてアピールできる点である。

参考として、MIRACLE LINUX SE V 2.1上で実施したベンチマークAIM 7^{*9}の測定結果を図7に示す。この図は、特別なチューニングを実施することなく、32 CPUまでのスケーラビリティがほぼニアに伸びていることを示している。

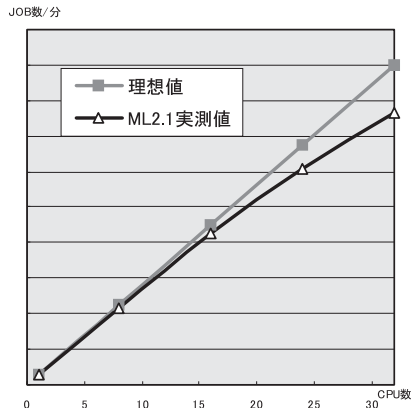


図7 ES 7000/540 での効率測定結果

3.1.3 ES 7000/400 シリーズのハードウェア・アーキテクチャ

ES 7000/400 シリーズは、Intel チップセットを採用した 64 ビット・マシンであり、基幹部分に Intel チップセットを搭載しているという点で CMP アーキテクチャ色が薄らいだマシンとなっている。

EM 64 T^{*10} の登場により、32 bit (または 36 bit) メモリ・アドレッシングの制約がなくなり、IA 32 系のマシンでも 64 ビット・データを扱えるようになったが、IPF は命令セット・レベルで並列性を強化しており、IA 32 に比べ性能向上の幅が大きいと予想できる。NUMA ハードウェアに対してはリモート・メモリ・アクセスを減らし、ローカル・メモリ・アクセスを優先するためのメモリ参照ロジックの組み込みが必要になるが、カーネル 2.6 ではすでに次のような対応が行なわれている。

- 1) ACPI^{*11} からノードの物理メモリ・アドレスの範囲、ノード間の距離等を入手し、ノードごとのメモリを物理アドレスが非連続なアーキテクチャの元で管理する。
- 2) できる限りプロセスが動いているノード上のメモリを割り当てる。
- 3) O(1) スケジューラによりプロセスの RUN キューを CPU ごとに分割し、CPU とプロセスの結びつきを強めることでプロセスの CPU 間移動を抑制する。

ES 7000/400 は、I/O 構成における CMP アーキテクチャの採用、カーネル 2.6 の NUMA 対応強化等により、たとえば大規模データベース業務のように、64 ビット・アーキテクチャ上で優位性を発揮する業務に対して、ミッション・クリティカル・システムの構築を可能にしている。

3.2 ソフトウェアのミッション・クリティカル性能

3.2.1 ES 7000 Linux の OS 性能

メインフレームにおけるメーカー独自の OS とは異なり、ES 7000 Linux の OS 性能は、オープン・スタンダードである Linux カーネルそのものの性能である。スケジューラやメモリ管理といった OS の基本機能分野にはまだ改善の余地を残しているが、逆にメインフレームでは取り込めていない新しい機能も多数含まれている。参考として、NEC 社が公開している Express[®] 5800/1320 Xd での Linux と Windows の TPC C 測定結果を表 4 に示す。

表 4 は、SUSE[®] LINUX ES 9 (カーネル 2.6) が Windows Server 2003 Datacenter (64 bit)

と比較して、パフォーマンス (TPM C) が 18% 優れ、トランザクション当たりの価格 (コスト/TPM C) が 23% 安い水準にあることを示している。両者の測定時期に差があり、この数字に着目して Linux が優れているという結論を導き出すことはできないが、Linux がすでに Windows 並みの処理性能を発揮できることを示す一つのデータと言える。

表 4 業界標準ベンチマーク TPC C によるコスト・パフォーマンス比較

プラットフォーム	OS	プロセッサ個数	プロセッサ速度/ キャッシュ容量	メモリ 容量	TPM・C	コスト/ TPM・C	有効 年月日
NEC Express 5800/1320Xd	SUSE LINUX ES9(カーネル 2.6)	32	Itanium2 6M 1.5GHz, 6MB L3	512GB	683,575	\$5.99	2004/10/5
NEC Express 5800/1320Xd	Windows Server 2003 Datacenter	32	Itanium2 6M 1.5GHz, 6MB L3	512GB	577,531	\$7.74	2003/12/1

TPC-C サイト http://www.tpc.org/tpcc/results/tpcc_results.asp?orderby=hardware から (2005/1/19 現在)

カーネル 2.6 は、2.2 節で述べたように機能面ではメインフレームにかなり近づいているが、ミッション・クリティカルという切り口からは保守性の低さが問題点として認められる。障害解析機能の充実に目が向けられていない点がメインフレームとの大きな違いであり、ミッション・クリティカル・システムへの適用にあたっては保守性を改善していくことが大きな課題となっている。

3.2.2 パーティションの粒度

ユーザから見た CMP アーキテクチャの最大の特徴は、一つの筐体を複数の独立したシステムとしてパーティショニングできる点にある。分散された複数のサーバを単一プラットフォームに統合し管理することで、パフォーマンスの向上だけでなく管理コストの低減が可能になる。しかし、追加キャパシティの要求に対して柔軟に対応でき、顧客の投資コストを最小限に押さえるシステムを提供するという点では、ES 7000 のパーティションごとの CPU 数、システムごとのパーティション数はまだ「粒度」が十分とは言えない。

性能に対する柔軟性をより高めるために、ES 7000 はソフトウェアとの連携によるダイナミック・パーティショニング機能を備えている。処理要件を変更すると、すぐにその場でハードウェア構成変更を行うダイナミック・パーティショニング機能は、この「粒度」を改善するための重要な機能であるが、本機能を実現するためにはカーネル内にハードウェア・インタフェースを定義しなければならない。Unisys は ES 7000 の設計当初から本機能の提供を計画していたが、結果的に 5 年を経過した 2005 年 1 月現在でも Microsoft 社から本機能は提供されていない。現在、Unisys は Linux 上でのダイナミック・パーティショニング機能を開発中であるが、自らの手で OS の変更が可能という点では Linux を採用することに優位性が認められる。

最近よく耳にするユーティリティ・コンピューティング^{*12} は、提供能力に対する粒度をより柔軟に動的に対応する考え方であり、これを ES 7000 をベースに実現しようとするれば、無数のパーティションが必要になる。Unisys が開発中のダイナミック・パーティショニング機能は、物理パーティションに対して動作する機能であり、障害コンポーネントを物理的に切り離すことが可能になるが、キャパシティ管理の粒度をより細かくしていくためには論理(仮想化)パーティションの機能を取り込み、ES 7000 提供能力の柔軟性を更に高めていく必要がある。

3.2.3 ソフトウェア製品の品揃え

各ハードウェアベンダがオープン・スタンダードなチップセット、オープン・スタンダードな Linux OS を採用することで競合他社とのサーバ・ハードウェアの性能差はある程度まで均一化されていくと想像できる。今後はハードウェア性能でマシンを決めるのではなく、ソリューションを提供するベンダや Sler からの推奨ハードウェアをマシン選定の決め手にするケースが増えていくと推測する。この場合、提供できるプロダクトが豊富であればあるほどプラットフォームの優位性を発揮できることになる。

表 5 に ES 7000 で評価済みの代表的なソフトウェアをまとめてみた。

アプリケーション系プロダクトの評価は、商用ソフトウェア、OSS とともにいくつかのプロダクトの作動確認を完了しているが正式サポートには至っていない。また、開発支援ツールとして、Windows における Visual Studio[®]や .NET のような高性能ツールが存在しない点等、Windows 環境のように体系化されたプロダクト・セットの提供には追いついていない。ただし、データベース系のミッション・クリティカル・システムをサポートするという観点に立てば、すでにソフトウェア・ラインアップは整っており、現在は各主要ソフトウェアの実績を積み上げている状況にある。

一方、Unisys ではメインフレーム向け主要ソフトウェアの Linux 対応を着々と進めており、メインフレームから Linux へ移行するシステムに対しても、ミッション・クリティカル・システムを強く意識したソフトウェアの品揃えが進んでいる。

表 5 ES 7000 Linux 上で評価済みの主要ソフトウェア

分野	ソフトウェア名	機能	販売元
アプリケーション	なし	—	—
クライアントサービス	なし	—	—
サーバ共用・D/B	・Oracle9i, Oracle10g	データベース・ソフトウェア	Oracle
分散基盤	・CLUSTERPRO SE2.1, SE3.0 ・HDLM	クラスタリング・ソフトウェア I/O パス二重化ソフトウェア	NEC 日立
開発支援	なし	—	—
運用管理	ARCserve V9, r11 Netbackup 5.1 サーバ・ナビゲーション・ツール	バックアップ・ソフトウェア バックアップ・ソフトウェア 運用管理・ソフトウェア	CA VERITAS Unisys
開発管理	なし	—	—

4. ミッション・クリティカル Linux のための要件

本章では、ES 7000 におけるミッション・クリティカル要件について考察する。

4.1 ミッション・クリティカル要件項目

今回の調査では、ソフトウェア品質を示す ISO/IEC 9126 (JIS X 0129) に対する検討も実施しているが、品質特性を切り口として要件をまとめた場合、ミッション・クリティカル全体としての要件を把握することが難しいと判断し、参考として参照するに留めている。表 6 にミッション・クリティカル・システムを実現するにあたり、初期段階で必須となる要件項目をまとめてみた。(ただし、現時点で既に対応済み、あるいは対応完了が見えている項目については記載していない) 筆者がミッション・クリティカル要件を検討する上で特に重要と考えた項目は、以下の三点である。

- 1) メインフレームと同等以上の信頼性，可用性，運用性，保守性，機密性か．
- 2) ユーザ要求を満たすプロダクトが存在するか．
- 3) 自社製品と同等レベルのサポートを提供できるか．

表6 ES 7000 Linux において未達成な重要ミッション・クリティカル要件

No	ミッション・クリティカルにおける重要項目	項目	内容
1	メインフレーム並みの RASIS 性能	可用性	99.999%の稼働率確保
2			障害リカバリ手順の完成度
3			障害時のシステム切り離し
4			コンポーネントの完全二重化
5		運用性	ジョブ管理機能
6		保守性	Linux カーネルの迅速な保守サポート
7			OSS, 商用 S/W の迅速な保守サポート
8			ダンプ情報/システム稼働情報/システム・ログ情報の完成度
9		機密性	B1 レベル認証以上のセキュリティ確保 (SE Linux の導入)
10	ユーザ要求を満たすプロダクト	プロダクト・セット	MIDMOST*
11			セキュリティ S/W
12	自社製品と同等レベルのサポート	人材確保/育成	Linux/他社 S/W に対するサポート・スキルの育成と確保
13		アライアンス強度	RedHat 社または Novell 社との強固なアライアンス
			Microsoft 社との HARQ (High Availability Resolution Queue) 契約に近いアライアンス
14			米国 Unisys のバックライン・サポート
15	検証環境	事前検証/緊急対応を可能にする ES7000 Linux 環境の確保	

* MIDMOST : 日本ユニシスが開発したオープン環境ミドルウェア．開発・保守生産性の向上とシステムの安定稼働，効率的な運用を実現するソフトウェア．

表6において，可用性，運用性，機密性は，オープンソース・コミュニティの中で急激な性能進化を遂げており時間とともに解決されていく可能性が高い．一方，保守性に関してはコミュニティ全体がミッション・クリティカル・システムをサポートするという意識になりきれておらず，今後も各社独自の強化策が必要になる領域と考えられる．

ユーザ要求を満たすプロダクトの提供については，メインフレームのようにすべてのプロダクトを自社開発で提供する必要がなく，最も高性能で高品質なプロダクトを OSS または他社商用ソフトウェアの中から厳選し調達することが可能であるが，システムの基幹部分に自社製ソフトウェアを適用できれば，ミッション・クリティカル・レベルの保守サポートを提供する上で大きな効果を期待できる．また，Linux はメインフレームよりもセキュリティの脆弱性を狙われやすいシステムであることを熟慮し，メインフレーム機能よりも優れたセキュリティ技術を実装していくことが重要である．

OSS や他社ソフトウェアに対して自社製品と同等レベルのサポートを提供するためには，障害の発生を未然に防ぐための検証環境の充実，新しい技術に対応できる人材の育成・教育活動が不可欠な要件となる．メインフレームのサポートでは，あらゆるプロダクトが自社製品であり，すべてのプロダクトに精通していたが，他社製品のサポートに対しては各製品の信頼性依存度に応じたアライアンスの強化が必須である．

5. ミッション・クリティカル・システムの実現に向けて

本章では，システム・プロバイダが取り組むべき事項について説明する．

5.1 高可用性データの提示

雑誌等では、高性能ブレード・サーバやグリッド・コンピューティングといった低価格でスケーラビリティを提供するソリューションが多く話題にされているが、IDCの市場調査等が示しているように、ミッション・クリティカル・システムを持つ顧客は、コンピュータの専門家が考えているよりもはるかに安全志向であり、重要なデータベースを扱うミッション・クリティカル業務は、現在もメインフレームを中心としたスケールアップ型システムで構築されている。

Linuxによるミッション・クリティカル・システムを実現していくためには、まず顧客の信頼を勝ち取る必要がある。そのためには、Linuxの高可用性計画を積極的に公開していく必要があるだろう。たとえば、表7に示すような高可用性目標を設定し、ユーザに対して数値データでLinuxの品質を示していくことが重要と考える。

一般的にクラスタリング・システムは高可用性を意味するシステム構成のことを指すが、単なるクラスタリングでは可用性の向上にはつながらない。なぜなら、ソフトウェア・バグの多くは、ハードウェア障害やリソース不足に起因して発生し、フェイルオーバー後も同一原因によるソフトウェア・バグが発生してしまう可能性が高いためである。ミッション・クリティカル・システムの実現には、ハードウェア障害の少ない高性能なサーバが必要になること等をデータとして示していく必要がある。

表7 可用性の達成目標例

	システム可用性	動作不能時間/年
シングルノード・システム	99.9%	8時間
クラスタリング・システム	99.99%	50分
完全二重化システム	99.999%	5分

5.2 保守サポート力の提示

高可用性を実現するためには、障害を発生させないシステムを構築することが最も重要であるが、障害発生時の保守サポート力こそが、Linux導入企業がベンダ、Sler各社に求めている最大の関心事ではないだろうか。顧客に対してサービス・レベルを明確に示せない限り、ミッション・クリティカルLinuxの実現はあり得ない。

5.3 企業コミュニティへの参加

ユニアデックスは、2004年6月から日本OSS推進フォーラム開発基盤WG^{*13}に参加している。この活動の中で、ユニアデックスはミッション・クリティカル・システムを視野に入れた高信頼化ツール(ダンプデータ解析ツール)の開発^{*14}を提案し、2004年10月からNTTデータ社、ミラクル・リナックス社との協同開発を行っている。本ツールは、Linuxの保守性改善を目指したものであり、ユニアデックスがメインフレーム技術のノウハウをベースに設計・開発を担当し、両社の技術支援を得ながら進めている開発作業である。

企業コミュニティの中で、積極的に情報、ノウハウを共有していくことが、オープンソース全盛時代を迎えようとしているベンダ各社にとって、生き残りをかけた重要な取り組みになっている。OSSの普及が進むにつれ、ソフトウェアに対する考え方は、開発・所有することから、組合せ・利用することへと大きく変化しつつある。これまでは想像し得なかった競合他社

との協業も各社の得意分野を活用し、より大きな成果をあげるための重要な手段となっている。ベンダ、Sler 各社は、企業コミュニティの形成がミッション・クリティカル Linux の実現を早め、新たなビジネス・チャンスの扉になることを忘れてはならないだろう。

6. お わ り に

ES 7000 Linux がミッション・クリティカル・システムの基盤として稼働できる日は、すでにすぐそばまで来ている。残された課題はけっして簡単に解決できるものばかりではないが、ES 7000 Linux によるミッション・クリティカル・システムは、大規模データベース・システムを基点に動き始めようとしている。今、我々が取り組んでいるミッション・クリティカル・システムへの挑戦は、Linux がミッション・クリティカル業務に耐え得ることを実証し、ハードウェアを含めたトータル・ソリューションを提供することへの挑戦である。ES 7000 Linux の成功は、Linux の普及促進、OSS 関連ビジネスの拡大に貢献し、Linux コミュニティ全体の成功につながるものと考える。

日本ユニシス・グループ、米国 Unisys、OSS コミュニティといった、あらゆるチャネルとの協力関係を強化し、たとえば、「ミッション・クリティカルなら ES 7000 Linux」と言わしめるような取り組みを今後も続けていく所存である。

-
- *1 ES 7000: Unisys 社が 2000 年に販売開始した Intel チップベースの大規模 IA サーバ・シリーズ名。
 - *2 ES 7000 Linux: オペレーティング・システムとして Linux を採用した ES 7000 システムの総称。
 - *3 ミッション・クリティカル・システム: 高度な信頼性や耐障害性が必要な業務を処理するシステム。このシステムは一時の中断や誤りがあってはならず、セキュリティの面でも万全の対策が必要である。
 - *4 エンタープライズ・システム: 全社規模の情報システム。特定の部門や部署だけが利用するシステムではなく、全社をターゲットにした大規模システムを指す。
 - *5 RASIS: 信頼性 (R), 可用性 (A), 保守性 (S), 保全性 (I), 機密性 (S) の略称。
 - *6 ES 7000/500 シリーズ: Intel Xeon プロセッサに基づいた ES 7000 ファミリー・システムの名称。
 - *7 ES 7000/400 シリーズ: Intel Itanium 2 プロセッサに基づいた ES 7000 ファミリー・システムの名称。
 - *8 CPU アフィニティ: CPU キャッシュの有効性を高めるために、プロセスを同一 CPU 上で実行させようとする CPU スケジューリング方法のこと。
 - *9 AIM 7: マルチユーザ環境向けのオープンソース・ベンチマーク。大量ディスク I/O も考慮された作りになっているが、CPU 負荷測定向きのベンチマーク。UNIX で 10 年以上の実績がある。
 - *10 EM 64 T: Intel 社が同社の 32 ビット・マイクロプロセッサに組み込んでいる 64 ビット拡張機能。
 - *11 ACPI (Advanced Configuration and Power Interface): パソコンの電力制御インターフェース。節電管理、バッテリー管理、プラグ・アンド・プレイなどと連動しており、OS が主導権を持つ仕組みになっている。
 - *12 コーティリティ・コンピューティング: コンピューティング資源をコーティリティ (電気・ガス・水道) のように、必要に応じて必要な分だけ利用するといったコンセプトを指す。
 - *13 日本 OSS 推進フォーラム: 経済産業省/総務省が後押しする OSS の活用推進を目的としたフォーラム。開発基盤 WG は具体的な課題解決を行う下部組織の一つ。
 - *14 高信頼化ツールの開発: ダンプデータ解析ツールは、「独立行政法人 情報処理推進機構オープンソースソフトウェア活用基盤整備事業」に係る委託業務の一環として開発している。

- 参考文献**
- [1] 高橋浩和 他, カーネル 2.6 のすべて, 日経 Linux, 10月号, 2003年10月, pp. 28~60
 - [2] 谷本雄一郎, DB サーバ統合における NUMA 対応 IA サーバの性能評価, IBM プロフェッショナル論文, 2003年8月, pp. 1~7
 - [3] CELLULAR MULTIPROCESSING AND UNIFORM MEMORY ACCESS, A White Paper, Unisys, 1999年12月, pp. 1~15
 - [4] ENTERPRISE CLASS LINUX, A White Paper, Unisys Server Technology, 2004年8月, pp. 1~7
 - [5] Intel E 8870 Chipset, Product Brief, Intel, 2004年, pp. 1~4
 - [6] Ian Brown, Unisys ES 7000, Gartner Group, 2000年10月, pp. 1~11
 - [7] Tom Manter, Unisys の Windows 2000 製品とサービス: メインフレームコンピューティングの再定義, Aberdeen Group, 2001年8月, pp. 1~50
 - [8] パーノン ターナー, SPARC システムが実現する垂直スケーリングアーキテクチャ, White Paper, IDC, 2004年3月, pp. 1~7
 - [9] ソフトウェア品質特性(ISO/IEC 9126)の解説, ADAMOS, <http://www.cam.hi-ho.ne.jp/adamosute/kyotu/iso9126.htm>

執筆者紹介 高橋 秀 樹 (Hideki Takahashi)

1981年群馬大学工学部情報工学科卒業。同年日本ユニシス(株)入社。入社以来、一貫してIX系メインフレームOSの開発・保守に従事。2003年からはメインフレーム業務に加え、ES 7000 Linuxの受入・開発に従事。現在、ユニアデックス(株)プロダクト事業グループ ソフトウェアプロダクト統括部サーバソフトウェア二部第一グループマネージャ。