

Unisys e-@ction Enterprise Server ES 7000 を支える ソフトウェア新技術

Advanced Software Technology Supporting Unisys e-@ction Enterprise Server ES 7000

清水 孝 美

要 約 今日の急速なアプリケーション規模の成長により、サーバのスケラビリティが強く求められている。ユニシスはこの要求に応え、ユニシス独自の CMP (Cellular Multiprocessing) アーキテクチャによる Unisys e-@ction Enterprise Server ES 7000 (以降、ES 7000 と記述) を開発した。このアーキテクチャには二つのスケラビリティの要素がある。一つは最大 32 CPU までの大規模 SMP であり、もう一つは、システム全体を最大八つまで論理的に分割して個別に稼働させるパーティショニング機能である。また、ソフトウェアの新技術としては、分割した各パーティション間でメモリの一部を共有する「共有メモリ」機能がある。本稿では、ソフトウェアの新技術である共有メモリに焦点をあて、その仕組みや利用方法などを説明する。

Abstract Scalability is the requirements strongly asked for today's rapidly expanding the number of supported users of an application. In order to meet the requirements, Unisys has developed the Unisys e-@ction Enterprise Server ES 7000 based on a concept of Unisys proprietary Cellular Multiprocessing (CMP) architecture. In the CMP architecture, two scalability options are available; the large-scale SMP of up to 32 processors and the partitioning capability. The latter facilitates to subdivide whole system into up to eight logical partitions each with its own set of physical resources, and to support an environment capable of running multiple instances of different operating systems. Also, the CMP architecture provides an advanced software technology, such as the CMP Shared Memory capability, which facilitates to share a part of system memory among specific partitions.

This article describes the mechanism and utilization of the shared memory on a Windows operating system, focusing on the Unisys CMP architecture's shared memory.

1. はじめに

ユニシスは近年、WindowsNT におけるエンタープライズシステムおよびデータセンター市場の獲得に乗り出した。この市場における多くの顧客は、“メインフレーム”と“サーバ”に対する次のような認識を持っている。“メインフレーム”は高価で新しさはないが、綿密なサポートとサービスのもとに運用されている。また、稼働を開始すると、ノンストップで運用可能である。それに対して“サーバ”は、安価であり最新の技術を採用している反面、スケラビリティに欠け、サポート体制の問題があり、加えて、ミッションクリティカルな処理における信頼性にも問題がある。

ユニシスは市場に対する WindowsNT Server および Windows 2000 Datacenter Server の投入によって、顧客のこのような認識を覆すのに十分なシステムを開発した。ユニシス独自の Cellular Multiprocessing (CMP) アーキテクチャを採用した ES 7000 シリーズは、オープン市場における既存のコンポーネントを利用することによ

る優れたコストパフォーマンスを実現したシステムであるが、同時に、メインフレーム並みのキャパシティ、アベイラビリティ、スケーラビリティ、サービス、そしてサポートを提供することができる。

本稿では、この CMP アーキテクチャの主要な機能である「共有メモリ」および「パーティショニング」に関して記述し、ソフトウェアから見た特徴を紹介する。

2. CMP アーキテクチャ

ES 7000 シリーズを構成する CMP アーキテクチャの特徴としてのハードウェア構成、パーティショニング、および共有メモリの概要に関して以下に述べる。これらはすべて、従来のサーバ機には搭載されていない機能であり、ES 7000 シリーズの根幹となる機能である。

2.1 ハードウェア構成

ES 7000 シリーズサーバは CMP アーキテクチャを採用し、Intel 64 ビットプロセッサを基本とする IA 64 アーキテクチャに対応しており（初期のリリースでは 32 ビットプロセッサ）、最大 32 個の CPU を搭載可能である。その他、CPU 外部キャッシュ、メモリ、I/O なども、クロスバースイッチと呼ばれる高速バスにより相互に接続され、CPU パワーを十分に発揮できるよう設計されている。これらの各モジュールの概要を以下に示す。

- ・外部キャッシュ（サードレベルキャッシュ：TLC）
4 CPU 毎に配置され、モジュールあたり最大 16 MB（IA 64 プロセッサでは 32 MB）搭載可能
- ・メインメモリ（Memory Storage Unit：MSU）
最大 64 GB 搭載可能（初期リリースでは 32 GB）
- ・I/O（Direct I/O Bridge：DIB）
ハードディスクやオプションカードなどの I/O を最大 96 接続可能
- ・クロスバースイッチ
TLC、DIB および MSU を接続するバスであり、各接続毎に 1.6 GB/Sec の転送速度を持つ

CMP アーキテクチャにおける最大構成を図 1 に示す。

サードレベルキャッシュ（TLC）は、最大 4 CPU で構成されるサブポッドと呼ばれる構成単位毎に配置され、これにより、CPU キャッシュのヒット率向上を図っている。また、各サブポッドは、クロスバースイッチを経由して、メインメモリである MSU に接続されている。システムバスを構成するこのクロスバースイッチは、大規模 SMP システムで主に採用される NUMA（Non-Uniformed Memory Access）の欠点であるメモリアクセス速度のばらつきをなくし、どの CPU からでも均一のメモリアクセスを可能にする。クロスバースイッチからのメモリアクセスは、最大 1.6 GB/Sec の転送速度を持ち、システム全体では約 20 GB/Sec のデータ転送が可能である。

2.2 パーティショニング

ES 7000 は、CPU、メモリ、I/O を含むシステム全体をパーティションと呼ばれる論理的な領域にサブポッド単位で分割することが可能であり、このパーティション毎

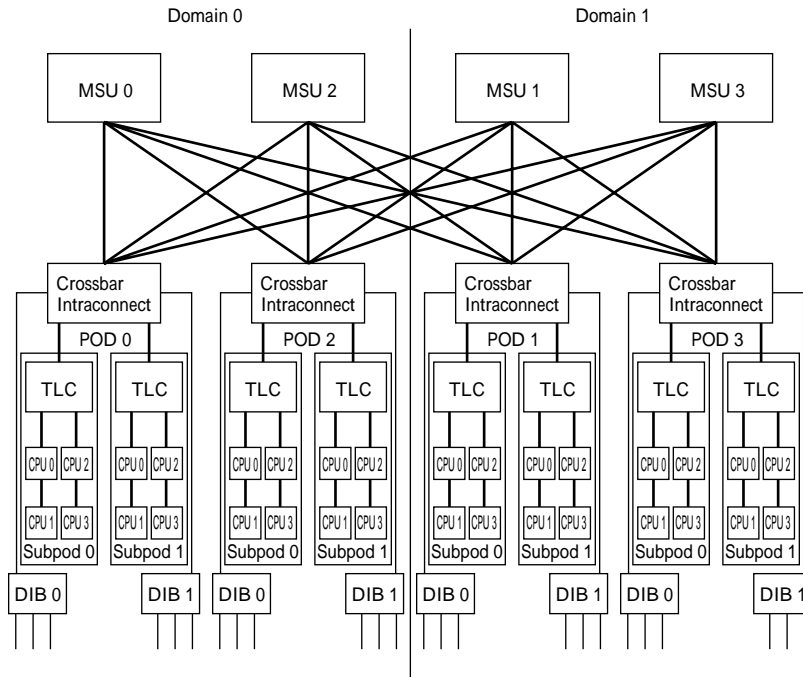


図 1 最大構成図

にオペレーティングシステムを稼働させることができる。したがって、最大構成においては八つのオペレーティングシステムを別々に稼働させることができる。これにより、複数のサーバで運用していた業務を、1台のES 7000ですべて実行させることが可能になる。また、分割されたそれぞれのパーティションはお互いに完全に独立しており、それぞれのパーティションにおいて発生したソフトウェアおよびハードウェア障害が、他のパーティションに影響を及ぼすこともない。

オペレーティングシステムは、WindowsNT Server 4.0 Enterprise Editionに加えWindows 2000 Datacenter Server を搭載可能である。

2.3 共有メモリ

複数のパーティションに分割した場合、それぞれ他とは独立した環境で動作するが、それだけでは複数のラックマウントモデルのサーバをひとつのラックに搭載することと、ソフトウェア的に大差ない環境である（もちろん、ハードウェア的には重要な意味がある）。ES 7000では、ソフトウェアの面からパーティションをさらに有効に活用するために、パーティション間でメモリを共有することを可能にしている。これは従来のサーバには搭載されていない機能であり、ES 7000の優位性を示す利点のひとつである。

オペレーティングシステムに対応した共有メモリドライバ、Windows 2000のHAL（Hardware Abstraction Layer）、およびハードウェア制御機構IMS（Integrated Maintenance System）の一部であるMIP（Management Interface Processor）に搭載される機能により、各パーティションで実行しているアプリケーション間でのデータの共有や通信を高速に行うことを可能にしている。以降の章では、共有メモリの仕組み

や活用方法に関して記述する。

3. 共有メモリおよびパーティショニングの活用

従来のサーバの運用形態は、Exchange Server や SQL Server など業務の中核を成すソフトウェアを別々のサーバ上で稼働させ、また、部門ごとにサーバ(部門サーバ)を設置するなどして、複数のサーバで構成された運用が中心である。さらに、業務の拡張にしたがってサーバの追加設置も必要になる。これにはコストがかかり、サーバ台数の増加によるメンテナンスコストの増大や、設置面積の拡大も無視できない。ES 7000 では、パーティショニング機能の搭載によりこの問題を解決する。

3.1 パーティショニングによるサーバ統合

CMP アーキテクチャの一つにパーティショニング機能がある。これは、サブポッドを一つの単位として、システム全体を最大八つの複数のパーティションに分割することができる。これらの各パーティション上で、オペレーティングシステムをそれぞれ独立して実行することが可能である。パーティションには CPU、メモリ、I/O などの各リソースも含まれ、それぞれ任意に割当てることが可能である。業務内容によっては多数の CPU やメモリを必要とするものもあるであろうし、CPU は少数でも大容量のメモリを必要とするものもあるであろう。このように、業務内容によって必要とされるリソースを柔軟に割当てることにより、業務内容に応じた最適なシステム環境を構築することができる。また、パーティション間でのリソースは完全に分割されているため、1 台の ES 7000 システムであっても、ハードウェアやソフトウェアの障害発生時にパーティションにまたがって影響を与えることはなく、一つのパーティションでシステムクラッシュやメモリ障害が発生しても、他のパーティションは何ら影響を受けずに、業務を継続することが可能である。

このため、ソフトウェアの面から見ると、複数のサーバを別々に運用している現状の業務を、リソースを柔軟に最適化させてそのまま移行することができる。さらに、ユニシスが Microsoft 社と共同で開発を行っている Windows 2000 Datacenter Server オペレーティングシステムの将来のバージョンでは、各パーティションが使用するリソースを、パーティションの負荷状況に応じて動的に増減することが可能になり(ダイナミックパーティショニング)、さらに柔軟な統合されたサーバ環境を構築することができる。

このようなパーティショニング機能の利点を最も生かせることができるもののひとつには、クラスタリングシステムがある。

図 2 に、パーティションを利用したクラスタリングシステムの例を示す。

この例では 8 CPU 構成の四つのパーティションに分割し、それぞれがデータベースファイルを共有している。また、パーティション間の通信には LAN を使用している。このように、従来は 4 台のサーバによって構築されていたクラスタリングシステムが、1 台の ES 7000 によって実現できる。

このほかにも、システムの開発時やシステム環境移行の際の検証にも、パーティショニング機能は有効に活用できる。例えば、最終的な実運用ではシングルパーティションとして稼働するシステムに対して、ソフトウェア開発やハードウェアのバージョ

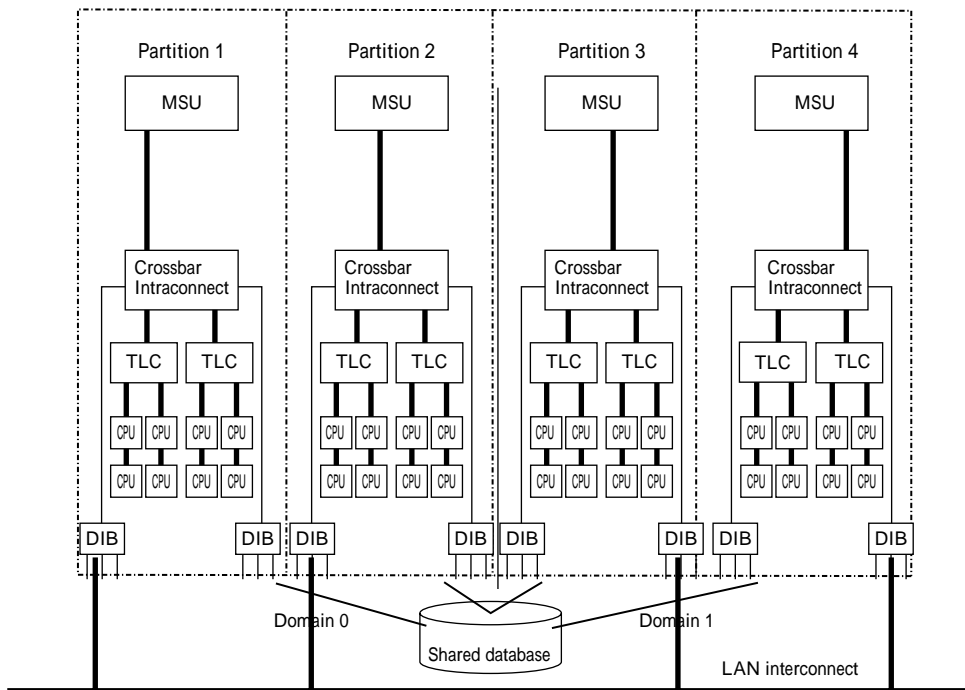


図 2 パーティショニングによるクラスタ構成

ンアップを行う場合、従来では別途サーバを設置して開発および検証作業を行う必要があった。ES 7000 においては、開発や検証用にパーティションを作成して、そこで作業を行うことが可能になる。このとき、検証用パーティションで問題が発生したとしても、稼働中の本番環境（パーティション）には何ら影響を与えることはない。その後、開発や検証で使用したパーティションを本番環境のパーティションに統合して、シングルパーティションとして本番運用することにより、システムのリリースにかかるコストを大幅に削減することができる。

ES 7000 においては、CMP アーキテクチャの一つである「共有メモリ」機能により、さらに独自性/機能性を高めている。

3.2 共有メモリを使用したパーティション間通信

ES 7000 システムでは、共有メモリ機能の実装により、複数のオペレーティングシステム環境におけるメモリ領域の共有が可能になった。ES 7000 に実装されている共有メモリ機能は、複数に分割されたパーティション間で有効であり、これらのパーティション間での通信のために使用することが可能である。この通信の用途には、大きく分けて二つのものがある。一つはサーバ間メッセージ通信であり、もう一つはパーティション間データ共有である。

3.2.1 サーバ間メッセージ通信（共有メモリ LAN）

複数のサーバによる従来の運用形態では、サーバ間の通信には Ethernet に代表される LAN を使用するのが通常である。しかし、もっとも普及しているタイプの Ethernet のデータ転送速度は最大でも 100 Mb/Sec の能力しか持たず、高速通信を行うためにはさらに高価なネットワークインフラを構築する必要がある。

CMP アーキテクチャで構成される ES 7000 では、パーティショニング機能を有効に活用するために、共有メモリ (Shared Memory) 機能を実装しており、従来型の LAN に代表される外部バスによる通信に代わって、内部バスによる高速通信を可能にした。この内部バスはすなわちクロスバースイッチであり、最大転送速度 1.6 GB/Sec (システム全体では約 20 GB/Sec) の高速通信を行う。

2 章で示したクラスタリングシステムを例に、実際の使用例を説明する。

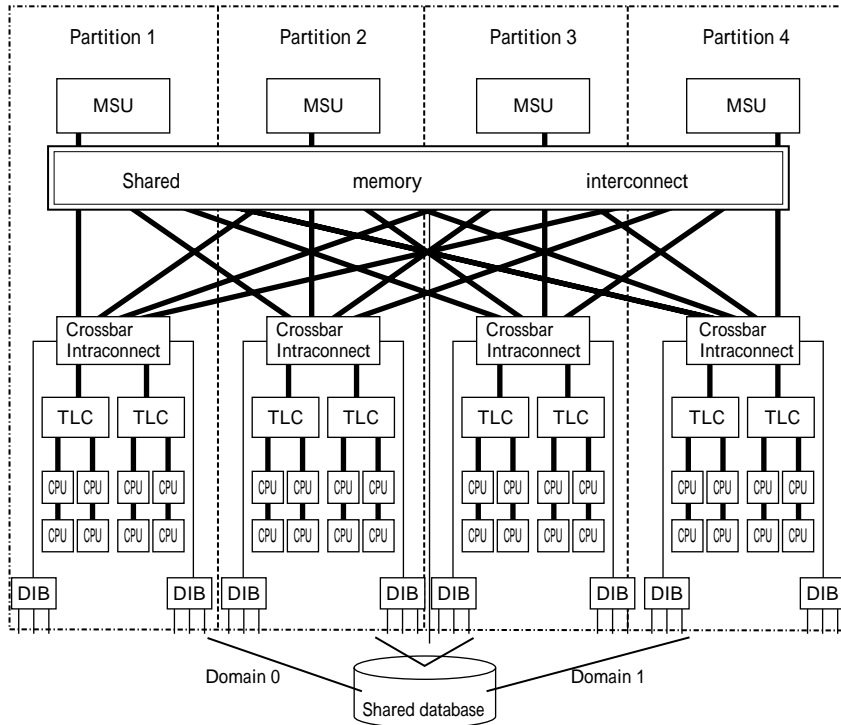


図 3 共有メモリを利用したクラスタリング構成

図 3 の例では、図 2 と異なる点が一つある。図 2 ではパーティション間は LAN 接続によって通信を行っていたが (LAN Interconnect)、この例ではすべてのパーティションは共有メモリで接続されている (Shared Memory Interconnect)。この場合の最大の利点は転送速度である。高速なメモリアクセス速度 (1.6 GB/Sec) で従来の LAN に代わるメッセージ転送が可能になり、さらに LAN のハードウェア接続による遅延を除外することができる。

ここで重要なことは、共有メモリ LAN によるメッセージ通信では、アプリケーションの変更は必要ないことである。従来のネットワーク接続のプログラミングのまま、共有メモリによる通信を実現することができる。これは、共有メモリ機能の中核を成す「共有メモリ Core Service」および「共有メモリ NDIS ドライバ」によって、ネットワークプロトコルをシミュレートすることにより通信ルートの切り替えを行っているためである。

LAN 接続から共有メモリ接続への切り替えにはいくつかの方法があるが、ここで

は、NDIS (Network Driver Interface Specification) による共有メモリ LAN と呼ばれる共有メモリ接続について、簡単に説明する。

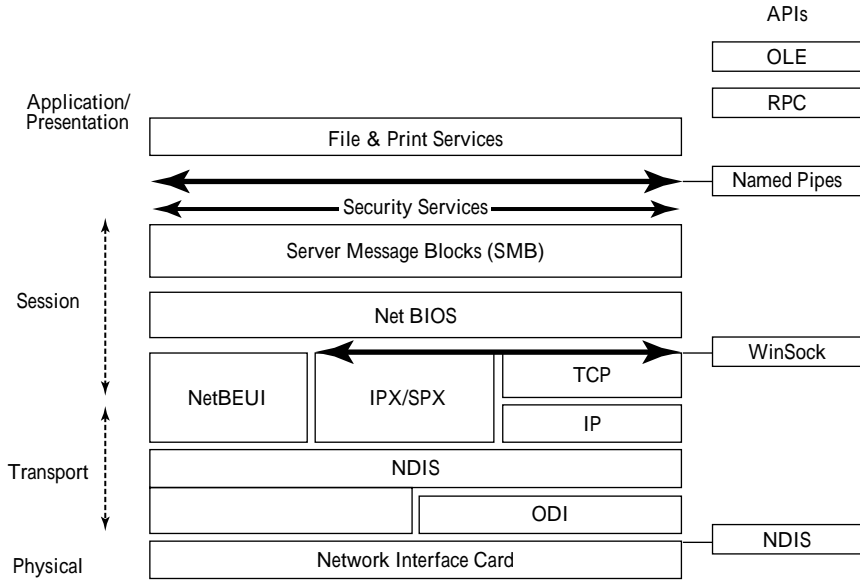


図 4 Windows ネットワーキング

図 4 は Windows ネットワーキングの構造を示したものである。NDIS による共有メモリ LAN は、NDIS 用の共有メモリドライバによって、ネットワークパケットを物理層ではなく共有メモリインタフェースに渡すことにより、ネットワークカードへのパケットの流れを共有メモリへ転送している (図 5 参照)。

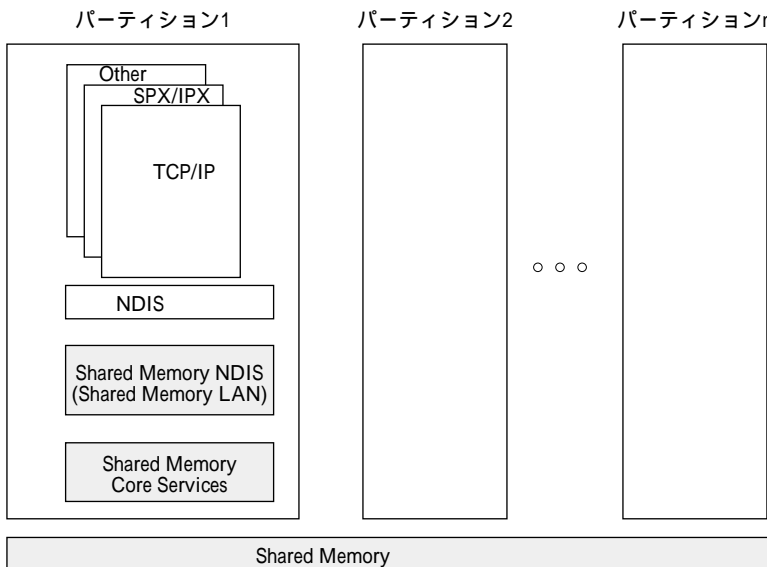


図 5 NDIS による共有メモリ LAN

共有メモリ NDIS ドライバがパケットデータを受信すると、そのデータを Core Service に渡して共有メモリに書きこむ。このとき、共有メモリへのアクセスは、Core Service、HAL、MIP の 3 者によって、それぞれが持つインタフェースを使用して管理/制御される。

NDIS と同様、Winsock 経由での共有メモリ LAN もサポートする。図 6 に Winsock を使用した場合の概念図を示す。

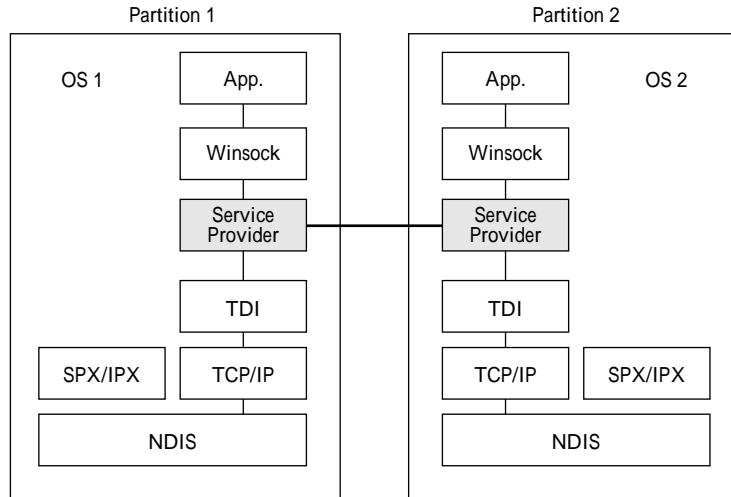


図 6 Winsock による共有メモリ LAN

Winsock の下位層に Service Provider と呼ばれるモジュールを挿入し、このモジュールが Core Service とインタフェースする。Winsock インタフェースで共有メモリ LAN を実現することにより、TCP/IP を経由せずにより上位の階層から Core Service とインタフェースすることが可能になる。この場合、プロトコルオーバーヘッドが削減され、パフォーマンスの面でもさらに有効である。

このほかに、VIA (Virtual Internet Architecture) での実装も予定されている。このように複数のネットワークインタフェースやプロトコルをサポートすることにより、将来的には Windows 2000 同士の通信のみならず、Windows 2000 MCP、Windows 2000 OS 2200 のような異種 OS 間の共有メモリ LAN が可能になる。

3.2.2 パーティション間データ共有

共有メモリを使用したサーバ間メッセージ通信に加えて、将来的には、パーティション間におけるデータ共有をサポートする予定である。サーバ間メッセージ通信と大きく異なるところは、アプリケーション側で明示的に共有メモリを使用することである。従来のアプリケーション間における共有メモリは、プラットフォーム（単一のオペレーティングシステム）内でのみ使用可能であったが、ES 7000 では、パーティション間すなわち複数のオペレーティングシステムの間でのメモリデータの共有が可能となる。共有メモリによるデータ共有は、MIP、Core Service ドライバ、および HAL などの CMP アーキテクチャ固有のモジュールによって管理され、Core Service ドラ

イバによって提供される共有メモリ API を使用することにより、アプリケーションが共有メモリを自由に使用することができる。

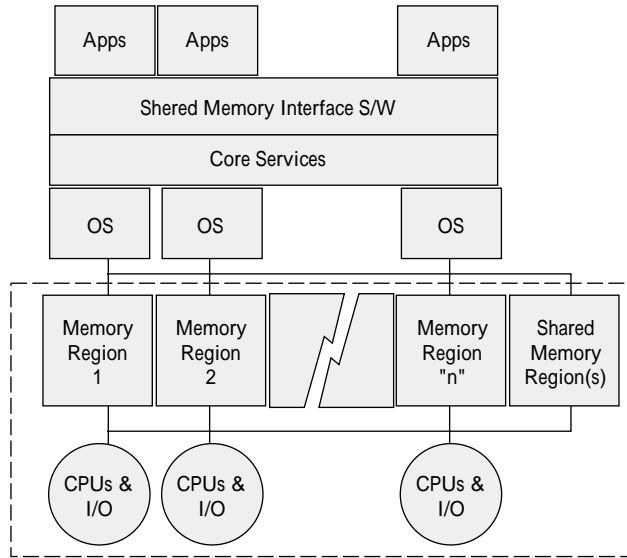


図 7 ソフトウェア階層

図 7 では、共有メモリをサポートするシステムのソフトウェアの階層を示している。破線で囲んでいる部分がプラットフォームを示し、複数のパーティションに分かれていることを表している（Memory Region は各パーティションのローカルメモリを示す）。

ソフトウェアの階層としては、まずパーティション上で実行されるオペレーティングシステムがあり、その上位に共有メモリを管理する Core Service が位置する。共有メモリへのアクセスは、最上位に位置するユーザアプリケーションではなく、このアプリケーションと Core Service の橋渡しをするソフトウェアによって行われることが理想である。ユーザアプリケーションは、このソフトウェアを利用することにより複雑な共有メモリ管理から解放され、アプリケーション本来の実行に専念することができる。

4. 共有メモリアーキテクチャ

4.1 メモリマッピング

メモリマッピングの方法としていくつかの方法があるが、ES 7000 では「プライベートメモリ」として共有メモリをカーネル（システム）メモリ空間またはユーザ仮想メモリ空間にマッピングする。このメモリ領域は、オペレーティングシステムのメモリ保護機能により他のパーティションから保護される。このようにマッピングされたメモリ領域は、共有メモリインタフェースを介して各パーティション間で共有される。共有メモリは、ひとつの領域を ES 7000 システム全体で共有するのではなく、複数の領域を共有領域としてアサインすることにより、それぞれの領域をアクセスする複数

のパーティショングループによる個別利用が可能である．図 8 に，複数のパーティショングループによるメモリ共有の例を示す．

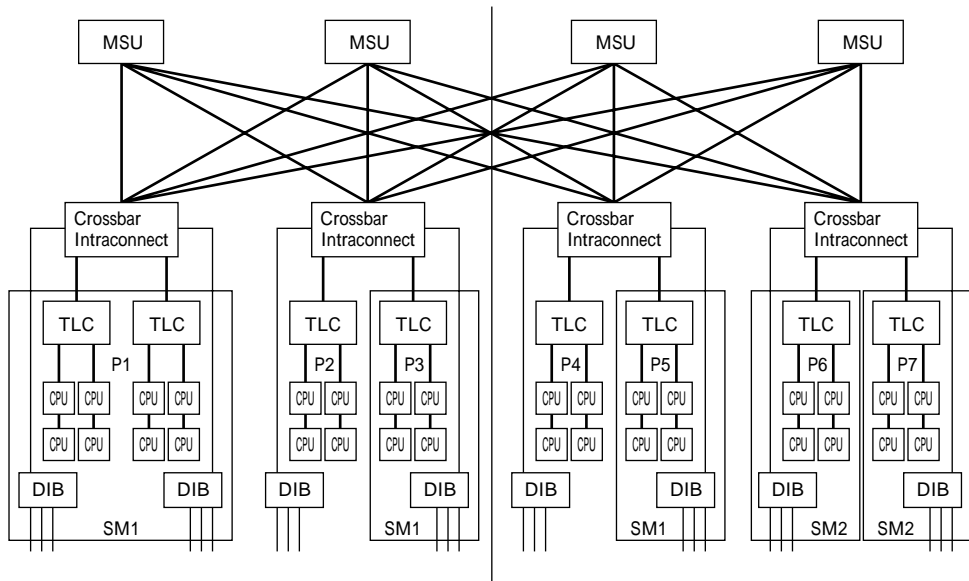


図 8 複数のパーティショングループによるメモリ共有

図 8 では，以下のような構成のパーティショングループで共有メモリを使用している．

- ・パーティション P1，P3，P5 による共有メモリ SM1 の使用
- ・パーティション P6，P7 による共有メモリ SM2 の使用
- ・パーティション P2，P4 は共有メモリを使用せずに，従来のローカルなメモリ空間のみを使用

ES 7000 システム全体の総メモリ容量は，各パーティションにおける OS のプライベート空間とすべての共有メモリ領域（図 8 の例では SM1 および SM2）を合わせた大きさになり，最大 64 GB まで使用可能である．

4.2 Core Service

共有メモリを使用したパーティション間データ共有を行う場合，3.2.2 節で記述したように，共有メモリをアクセスするソフトウェアは Core Service が提供する API を使用してメモリアクセスを行う．このソフトウェアのことを「共有メモリクライアント」と呼ぶ．共有メモリ LAN の場合は，共有メモリ NDIS ドライバが共有メモリクライアントとなり，パーティション間データ共有の場合は，Core Service API を使用して Core Service を呼び出すソフトウェアが共有メモリクライアントとなる．

図 9 は，図 7 で示したソフトウェア階層において，Core Service API を使用する共有メモリクライアントを示している．図中の Shared Memory Interface Software に位置するものが共有メモリクライアントである．前述のように，共有メモリの使い方によって，このソフトウェアは NDIS ドライバであったり何らかのサービスであっ

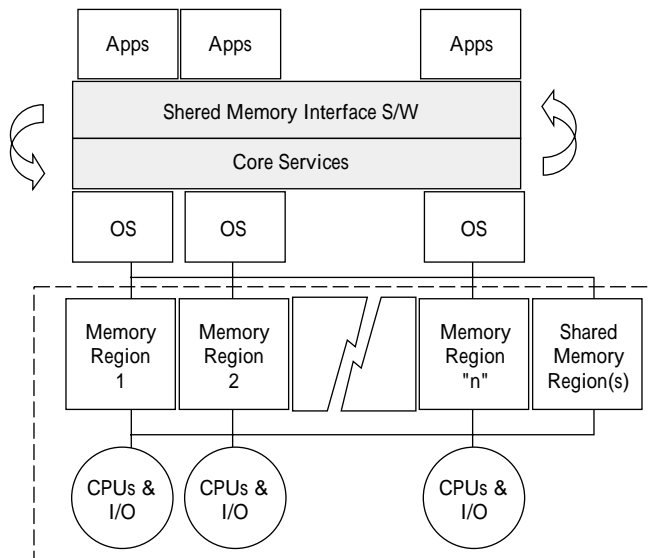


図 9 Core Service 共有メモリクライアント

たりさまざまなケースがある。また、アプリケーションが直接 Core Service を呼び出す場合もある。

Core Service はこのような共有メモリクライアントに対して、以下のような機能を提供する。

- ・共有メモリ領域の作成（初期化）
IMS によって設定された共有メモリ領域を割当て、管理構造体の作成や共有メモリクライアントのパーティショングループにおけるメンバーシップ設定などの初期化作業を行う。共有メモリ領域は、オペレーティングシステムが持つメモリ管理機能によるコントロールは受けずに、Core Service によってすべて管理される。
- ・機能の提供
クラスタリングソフトウェアやネットワークソフトウェアなどの、共有メモリクライアントに対する機能やインタフェースを提供する。主な機能としては、「パーティショングループへのクライアントの登録/削除」、「共有メモリの割当/解放」、および「クライアント間でのシグナル送信」などがある。
- ・プロセッサ間通信
プロセッサ間インタラプト (IPI) によって、プロセッサ間通信 (IPC) を行う (4.2.3 節を参照)。
- ・メンバーシップ登録
パーティショングループにおけるメンバーシップ登録を管理する。これは、パーティションや共有メモリクライアントによるパーティショングループへの参加/撤退や、何らかの障害によるパーティションの消滅に対してもサポートする。
- ・共有メモリの障害管理

共有メモリにバッドスポットなどの障害が発生した場合、その領域を共有メモリ領域から除外し、代替え領域を割当てるなどの管理を行う。

これらの機能を中心に共有メモリ全体を管理し、クライアントからのアクセス要求があった場合は MIP や HAL とインタフェースを取りながら、クライアントの要求を処理していく。

4.2.1 共有メモリ領域の割当て

共有メモリ領域は 128 MB から 2 GB までの大きさで、128 MB 単位で任意の大きさを定義できる。この設定は IMS 側のユーザインタフェースにより行い、システムのブート時に IMS が BIOS に情報を提供する。BIOS はこの情報をもとに、オペレーティングシステムとは別の領域に共有メモリを割当てる。

オペレーティングシステムの起動後は、Core Service デバイスドライバが共有メモリクライアントの共有メモリ管理を行う。

図 10 は従来のシステムにおけるメモリ管理を示しているが、すべてのメモリ領域はオペレーティングシステムによって管理されている。

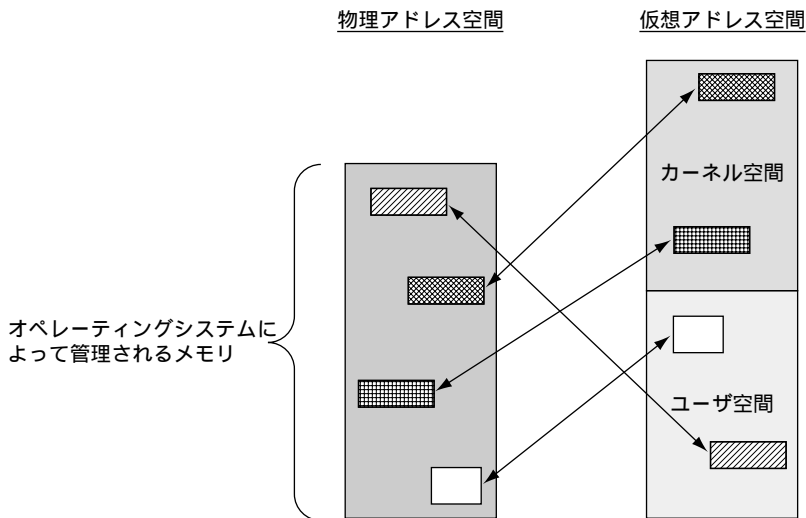


図 10 Windows 2000 によるメモリ領域の割り当て

共有メモリシステムの場合は、共有メモリとして割当てられた領域は図 11 のようにマッピングされ Core Service によって管理される。

Windows 2000 では、使用するメモリ量を MAXMEM パラメタによって指定することができるが、共有メモリシステムでは、概念的には同様なことを行っている。

オペレーティングシステムが提供する Win 32 もしくはカーネル API を使用してメモリを割当てる場合、そのメモリ領域はオペレーティングシステムの管理領域に割当てられ、オペレーティングシステムによって管理される。また、共有メモリとして割当てる場合は Core Service デバイスドライバを呼び出す。Core Service は、要求のあった共有メモリ領域を Core Service が管理する物理メモリ上に割当て、さらにカーネル API を呼び出して、その領域を仮想アドレス空間にマッピングする。

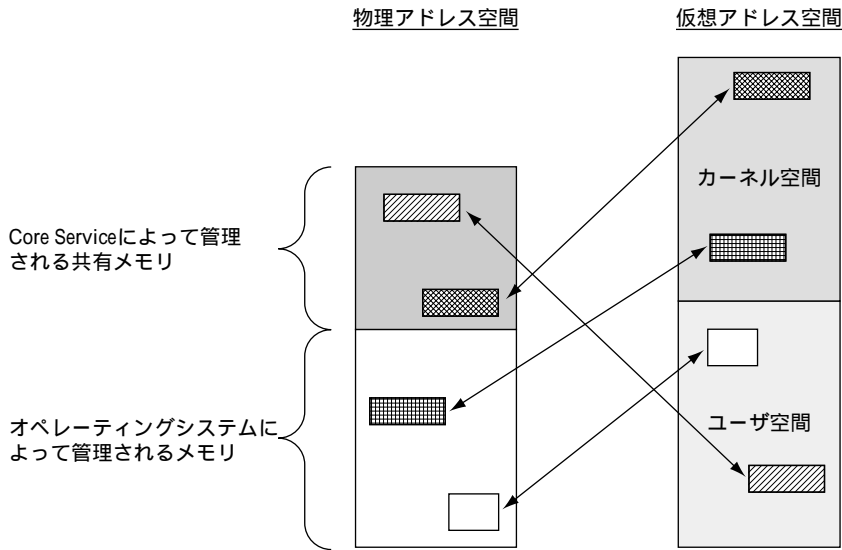


図 11 共有メモリ領域の割り当て

4.2.2 共有メモリ API

Core Service とインタフェースし、様々な機能を提供するソフトウェアのことを、「共有メモリクライアント」と呼ぶ。前述のように、共有メモリクライアントは Windows 2000 が提供する Win 32 API やカーネル API ではなく、Core Service が提供する API を使用する。例えば、Core Service API である ShMemAllocMem および ShMemDeallocateMem は、それぞれ Windows 2000 が提供する VirtualAlloc および VirtualFree と同様の機能を提供する。違いは、前者は Windows 2000 によって管理されるメモリ領域ではなく、共有メモリに対するアクセスに使用するという点である。

このような従来の API と同様の機能を持つもの他に、メモリ管理やパーティション管理などの API や、次に述べるようなクライアント間通信の API など、共有メモリ機能特有の API 群が提供される。

4.2.3 クライアント間通信

共有メモリは通常、複数のパーティションにまたがって使用されるため、パーティション間におけるクライアント間通信が必要になる。このクライアント間通信は、図 12 に示すように Core Service を経由したシグナル送信によって行われる。

この図はクライアント間通信の概略を示したもので、おおまかには以下の手順に従って通信が行われる。

- ① 送信側の共有メモリクライアントは、受信側の共有メモリクライアントにシグナルを送信するために Core Service API を呼び出す。
- ② 送信側の Core Service は、送信すべきシグナルを Partition Input Queue と呼ばれるバッファ（共有メモリ上に存在する）に入力する。
- ③ Partition Input Queue が空の場合、送信側の Core Service は受信側の Core Service にシグナル送信を通知するために、IPI (Inter-Processor Interrupt)

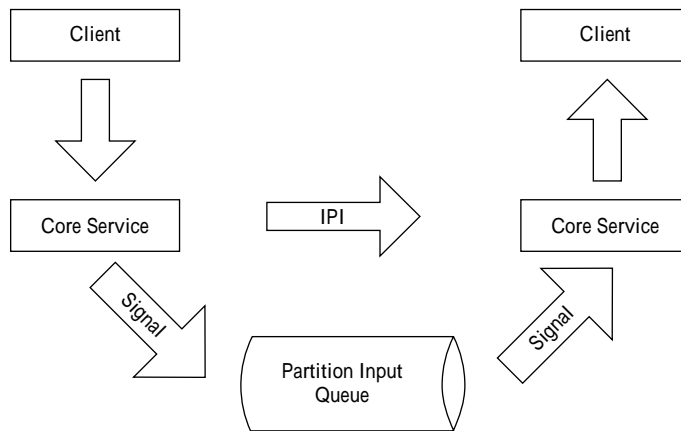


図 12 シグナルによるクライアント間通信

を発行する。

- ④ 受信側の Core Service は Partition Input Queue からシグナルを取り出す。

前述のように、共有メモリクライアントはこのシグナル送信や IPI を考慮する必要はなく、Core Service が提供する API を呼び出すことにより、以降の処理はすべて Core Service によって行われる。

この Partition Input Queue は共有メモリ上に存在し、その共有メモリ領域を使用するパーティション毎に設定される。また、このようなクライアント間通信では、Partition Input Queue 以外にも各パーティション上のローカルメモリ上に Send Queue、Tank Queue、Get Queue などの複数のキューが存在し、データの送信や受信のメカニズムの中で様々に使用される。

5. 将来の機能拡張

以上に述べた機能以外に、以下のような機能が今後の ES 7000 に実装予定である。これらの機能の実装により、ES 7000 の機能がさらに向上することが期待される。

- IA 64 アーキテクチャのプロセッサおよび Windows 2000 の 64 ビット対応 IA 64 プロセッサの使用および、共有メモリソフトウェアの 64 ビット対応によるアドレス範囲の拡大
- 異種 OS における共有メモリの使用
Windows 2000 だけではなく、MCP や 2200 OS における共有メモリの利用
- 共有メモリクライアントのミドルウェア化
ミドルウェア化した共有メモリクライアントの利用による、アプリケーションのスケラビリティ向上

6. おわりに

CMP アーキテクチャの特徴の一つである「共有メモリ」機能を中心に述べてきたが、実際の内部構造は、CMP アーキテクチャを構成するさまざまなコンポーネントが複雑に絡み合って実現されるため、本稿では概略的な内容にとどめて記述した。

冒頭でも述べているが、CMP アーキテクチャを実装した ES 7000 は高度なスケーラビリティを実現したシステムであり、ハードウェア的な特徴である「パーティショニング」機能とソフトウェア的な特徴である「共有メモリ」機能を融合することによって、先進的なシステム環境を提供することができる。このシステムは、Windows 2000 Datacenter Server によって実現される高スケーラビリティ環境に充分対応しているシステムである。

今後は、パーティショニング機能をベースとして、共有メモリの活用に関するさまざまな実証活動が行われるであろう。パーティショニング機能の活用例として、まずは「サーバ統合」があげられる。具体的な例はここでは述べないが、多数のサーバで運用されていた業務をその規模に応じて 1 台の ES 7000 上のいくつかのパーティションで実行することにより、設置や運用にかかるコストの削減や共有メモリ LAN によるパフォーマンスの向上も期待できる。また、ダイナミックパーティショニングにより、過去にない柔軟な運用が可能になり、ES 7000 の特徴をさらに発揮できるであろう。

客先業務を分析し、その実証を通じて、パーティショニングや共有メモリをさらに有効に利用する環境を見出していくことが重要である。

-
- 参考文献** [1] Cellular Multiprocessing Shared Memory White Paper, Unisys Corporation, 2000.
[2] Jim Hunter and Rich Grzegorek, NT Shared Memory Platform Interfaces, Design Review Document #98502/3, Unisys Corporation, 2000.

執筆者紹介 清水 孝 美 (Takayoshi Shimizu)

1983 年東海大学卒業。同年日本ユニシス(株)入社。金融向けワークステーションの回路設計およびドライバ開発に従事。その後、同システムの OS 日本語化に従事。現在、W 2 K テクノロジーセンタ ES ソフトウェア室に所属。