2-6-応. クラスタシステム構築に関する知識

1. 科目の概要

HPC (High Performance Computing)クラスタにおけるプログラミングに利用できる、HPF、OpenMP、MPI といった技術について解説する。さらに PC クラスタの構築方法や PC クラスタを管理するための SCore、その他のユーテリティも紹介する。

2. 習得ポイント

本科目の学習により習得することが期待されるポイントは以下の通り。

習得ポイント	説　明	シラバスの対応コマ
2-6-応-1. クラスタシステムの種類(HA クラスタとHPC クラスタ)	HA クラスタとHPC クラスタについて、それぞれの概要と特徴、背景と歴史を説明する。また、各クラスタシステムの目的と位置づけを比較し、各システムに固有の技術を概説する。	1,2
2-6-応-2. Active-Standby型HAクラスタの構築	Heartbeatを利用してHAクラスタを構築する手法について説明する。その構成要素や、仮想IPアドレスを共有したActive-Standby型クラスタの構築方法について説明する。	3
2-6-応-3. HDDのミラーリングによる冗長化	DRBD（Distributed Replicated Block Device）を利用してネットワークを介してHDDをミラーリングを行う方法について説明する。	4
2-6-応-4. スーパーコンピュータの歴史とクラスタによる実現	科学技術計算基盤としてのスーパーコンピュータについて、その概要と歴史、最近の動向を紹介する。さらにクラスタによるスーパーコンピュータの実現方法について触れ、PC クラスタに関するいくつかのプロジェクトを紹介する。	1,5
2-6-応-5. 並列計算機の種類と構成	並列プログラミングの概念を理解するために、まず計算の高速化と並列処理の必要性を示す。また並列計算機の種類について触れ、プロセッサやメモリの結合方式、バスのトポロジーなど、構成方式の違いについて解説する。	1,5
2-6-応-6. PCクラスタの構築方法	一般的なPC クラスタとして代表的なシステムであるBeowulf 型のPC クラスタを解説する。同クラスタを構成する方法を示し、各種の設定手順と並列プログラムのコンパイルおよび実行例を示す。またベンチマークの方法も紹介する。	12
2-6-応-7. HPCクラスタ向けユーティリティの利用	バッチ処理システム、システム監視ツール、並列処理ライブラリ、並列プロファイラ、並列プログラム向けデバッガ、並列プログラム開発環境、並列プログラムのベンチマークソフトウェアなど、HPC クラスタで効果的に活用できる各種のユーティリティソフトウェアの機能と特徴を説明する。	13
2-6-応-8. グリッドコンピューティング	グリッドコンピューティングの概念と各種のグリッドコンピューティングについて説明する。また米国、英国、欧州、アジア、そして日本におけるグリッドコンピューティングに関連する様々なプロジェクトを紹介する。	14

【学習ガイダンスの使い方】

1. 「習得ポイント」により、当該科目で習得することが期待される概念･知識の全体像を把握する。
2. 「シラバス」、「IT 知識体系との対応関係」、「OSS モデルカリキュラム固有知識」をもとに、必要に応じて、従来の IT 教育プログラム等との相違を把握した上で、具体的な講義計画を考案する。
3. 習得ポイント毎の「学習の要点」と「解説」を参考にして、講義で使用する教材等を準備する。

3. IT 知識体系との対応関係

「2-6-応. クラスタシステム構築に関する知識」と IT 知識体系との対応関係は以下の通り。

科目名	1	2	3	4	5	6	7	8	9	10
2-6-応. クラスタシステム構築に関する知識	クラスタシステム概論	HAクラスタ	Active-Standby型クラスタ	信頼性の確保	HAクラスタの周辺技術	HPCクラスタ	コンピュータシミュレーション	並列プログラミング概論	並列プログラミング:マルチスレッドプログラミング	並列プログラミング:OpenMP

11	12	13	14
並列プログラミング:PVMとMPI	Beowulf型PCクラスタ	HPCクラスタの周辺技術	グリッド・コンピューティング

<IT 知識体系上の関連部分>

分野		科目名	1	2	3	4	5	6	7	8	9	10	11	12	13
組織間連携事項と情報システム	1	IT-IAS 情報保証と情報セキュリティ	IT-IAS1. 基礎的な問題	IT-IAS2. 情報セキュリティの仕組み(対策)	IT-IAS3. 運用上の問題	IT-IAS4. ポリシー	IT-IAS5. 攻撃	IT-IAS6. 情報セキュリティ分野	IT-IAS7. フォレンジック(情報証拠)	IT-IAS8. 情報の状態	IT-IAS9. 情報セキュリティサービス	IT-IAS10. 脅威分析モデル	IT-IAS11. 脆弱性		
	2	IT-SP 社会的な観点とプロフェッショナルとしてのコミュニケーション	IT-SP1. プロフェッショナルとしてのコミュニケーション	IT-SP2. コンピュータの歴史	IT-SP3. コンピュータを取り巻く社会環境	IT-SP4. チームワーク	IT-SP5. 知的財産権	IT-SP6. コンピュータの法的問題	IT-SP7. 組織の中のIT	IT-SP8. プロフェッショナルとしての倫理的な問題と責任	IT-SP9. プライバシーと個人の自由				
応用技術	3	IT-IM 情報管理	IT-IM1. 情報管理の概念と基礎	IT-IM2. データベース問合わせ言語	IT-IM3. データアーキテクチャ	IT-IM4. データモデリングとデータベース設計	IT-IM5. データと情報の管理	IT-IM6. データベースの応用分野							
	4	IT-WS Webシステムとその技術	IT-WS1. Web技術	IT-WS2. 情報アーキテクチャ	IT-WS3. デジタルメディア	IT-WS4. Web開発	IT-WS5. 脆弱性	IT-WS6. ソーシャルソフトウェア							
ソフトウェアの方法と技術	5	IT-PF プログラミング基礎	IT-PF1. 基本データ構造	IT-PF2. プログラミングの基本的構成要素	IT-PF3. オブジェクト指向プログラミング	IT-PF4. アルゴリズムと問題解決	IT-PF5. イベント駆動プログラミング	IT-PF6. 再帰							
	6	IT-IPT 技術を統合するためのプログラミング	IT-IPT1. システム間通信	IT-IPT2. データ割り当てと交換	IT-IPT3. 統合的コーディング	IT-IPT4. スクリプティング手法	IT-IPT5. ソフトウェアセキュリティの実現	IT-IPT6. 種々の問題	IT-IPT7. プログラミング言語の概要						
	7	CE-SWE ソフトウェア工学	CE-SWE0. 歴史と概要	CE-SWE1. ソフトウェアプロセス	CE-SWE2. ソフトウェアの要求と仕様	CE-SWE3. ソフトウェアの設計	CE-SWE4. ソフトウェアのテストと検証	CE-SWE5. ソフトウェアの保守	CE-SWE6. ソフトウェア開発・保守ツールと環境	CE-SWE7. ソフトウェアプロジェクト管理	CE-SWE8. 言語翻訳	CE-SWE9. ソフトウェアのフォールトトレランス	CE-SWE10. ソフトウェアの構成管理	CE-SWE11. ソフトエアの標準化	
	8	IT-SIA システムインテグレーションとアーキテクチャ	IT-SIA1. 要求仕様	IT-SIA2. 調達/手配	IT-SIA3. インテグレーション	IT-SIA4. プロジェクト管理	IT-SIA5. テストと品質保証	IT-SIA6. 組織の特性	IT-SIA7. アーキテクチャ						
システム基盤	9	IT-NET ネットワーク	IT-NET1. ネットワークの基礎	IT-NET2. ルーティングとスイッチング	IT-NET3. 物理層	IT-NET4. セキュリティ	IT-NET5. アプリケーション分野	IT-NET6. ネットワーク管理							
	10	CE-NWK テレコミュニケーション	CE-NWK0. 歴史と概要	CE-NWK1. 通信ネットワークのアーキテクチャ	CE-NWK2. 通信ネットワークのプロトコル	CE-NWK3. LANとWAN	CE-NWK4. クライアントサーバコンピューティング	CE-NWK5. データのセキュリティと整合性	CE-NWK6. ワイヤレスコンピューティングとモバイルコンピューティング	CE-NWK7. データ通信	CE-NWK8. 組込み機器向けネットワーク	CE-NWK9. 通信技術とネットワーク概要	CE-NWK10. 性能評価	CE-NWK11. ネットワーク管理	CE-NWK12. 圧縮と伸張
			CE-NWK13. クラスタシステム	CE-NWK14. インターネットアプリケーション	CE-NWK15. 次世代インターネット	CE-NWK16. 放送									
	11	IT-PT プラットフォーム技術	IT-PT1. オペレーティングシステム	IT-PT2. アーキテクチャと機構	IT-PT3. コンピュータインフラストラクチャ	IT-PT4. デプロイメントソフトウェア	IT-PT5. ファームウェア	IT-PT6. ハードウェア							
	12	CE-OPS オペレーティングシステム	CE-OPS0. 歴史と概要	CE-OPS1. 並行性	CE-OPS2. スケジューリングとディスパッチ	CE-OPS3. メモリ管理	CE-OPS4. セキュリティと保護	CE-OPS5. ファイル管理	CE-OPS6. リアルタイムOS	CE-OPS7. OSの概要	CE-OPS8. 設計の原則	CE-OPS9. デバイス管理	CE-OPS10. システム性能評価		
コンピュータアーキテクチャ	13	CE-CAO コンピュータのアーキテクチャと構成	CE-CAO0. 歴史と概要	CE-CAO1. コンピュータアーキテクチャの基礎	CE-CAO2. メモリシステムの構成とアーキテクチャ	CE-CAO3. インタフェースと通信	CE-CAO4. デバイスサブシステム	CE-CAO5. CPUアーキテクチャ	CE-CAO6. 性能・コスト評価	CE-CAO7. 分散・並列処理	CE-CAO8. コンピュータによる計算	CE-CAO9. 性能向上			
複数領域にまたがるもの	14	IT-ITF IT基礎	IT-ITF1. ITの一般的なテーマ	IT-ITF2. 組織の問題	IT-ITF3. ITの歴史	IT-ITF4. IT分野(学科)とそれに関連のある分野(学科)	IT-ITF5. 応用領域	IT-ITF6. IT分野における数学と統計学の活用							
	15	CE-ESY 組込みシステム	CE-ESY0. 歴史と概要	CE-ESY1. 低電力コンピューティング	CE-ESY2. 高信頼性システムの設計	CE-ESY3. 組込み用アーキテクチャ	CE-ESY4. 開発環境	CE-ESY5. ライフサイクル	CE-ESY6. 要件分析	CE-ESY7. 仕様定義	CE-ESY8. 構造設計	CE-ESY9. テスト	CE-ESY10. プロジェクト管理	CE-ESY11. 並行設計(ハードウェア、ソフトウェア	CE-ESY12. 実装
			CE-ESY13. リアルタイムシステム設計	CE-ESY14. 組込みマイクロコントローラ	CE-ESY15. 組込みプログラム	CE-ESY16. 設計手法	CE-ESY17. ツールによるサポート	CE-ESY18. ネットワーク型組込みシステム	CE-ESY19. インタフェースシステムと混合信号システム	CE-ESY20. センサ技術	CE-ESY21. デバイスドライバ	CE-ESY22. メンテナンス	CE-ESY23. 専門システム	CE-ESY24. 信頼性とフォールトトレランス	

4. OSS モデルカリキュラム固有の知識

OSS モデルカリキュラム固有の知識として、Linux 上での並列プログラミングと Linux 上で動作するクラスタシステムがある。並列プログラミング言語とグリッドコンピューティングに関する知識を Linux 上の OSS 実装を通して習得する。

科目名	第1回	第2回	第3回	第4回	第5回	第6回	第7回
2-6-応 クラスタシステム構築に関する知識	(1) HA (High Availability)クラスタとHPC (High Performance Computing)クラスタ (2) HAクラスタ (3) HPCクラスタ	(1) HAクラスタ概論 (2) LVSで実現するロードバランサー	(1) HAクラスタにおける2つの考え方 (2) アクティブ・スタンバイクラスタの構 (3) 障害発生と復帰 (4) アクティブ・スタンバイクラスタを実現	(1) 可用性の評価指標 (2) 障害の状況 (3) 運用管理の留意点	(1) スパコンとHPCクラスタ (2) PCクラスタの要素技術 (3) HPCクラスタの適用領域	(1) 科学技術研究基盤としてのコンピュータシミュレーション (2) スーパーコンピュータ動向	(1) 計算の高速化と並列処理の必要性 (2) 並列計算機の分類 (3) 並列計算機の構成方式 (4) プロセッサ間のネットワーク (5) 並列処理のプログラミング (6) 並列処理の効率 (7) 数値計算の並列化 (8) 並列プログラミング環境とツール

第8回	第9回	第10回	第11回	第12回	第13回	第14回	第15回
(1) マルチスレッドプログラミング入門 (2) マルチスレッドプログラミングの基礎	(1) プログラミング並列化の方法 (2) 並列化の応用例	(1) OpenMPの概要 (2) APIの紹介 (3) OpenMPプログラムサンプルの紹介とデモ	(1) PVM (Parallel Virtual Mahcine) (2) PVMの構成 (3) MPI (Message Passing Interface) (4) MPIによるプログラミング	(1) Beowulf PC クラスタのコンポーネント (2) システム構築 (3) mpich2 のサンプルプログラムのコンパイル・実行 (4) ベンチマーク測定	(1) バッチ処理システムの紹介 (2) システム監視の紹介 (3) 並列ライブラリの紹介 (4) 並列デバッカ、プロファイラの紹介 (5) 開発環境の紹介 (6) ベンチマークソフトウェアの紹介 (7) PC クラスタで使用可能な商用ソフト	(1) グリッド・コンピューティングの概要 (2) グリッド・コンピューティングの分類 (3) グリッド・コンピューティング関連プロジェクトの紹介 (4) The Globus Alliance と Globus Toolkit	(1) クラウドコンピューティングと大規模分散クラスタ (2) 大規模分散クラスタで活用される代表的な技術

（網掛け部分は IT 知識体系で学習できる知識を示し、それ以外は OSS モデルカリキュラム固有の知識を示している）

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	応用
習得ポイント	2-6-応-1. クラスタシステムの種類(HA クラスタと HPC クラスタ)	
対応する コースウェア	第 1 回 クラスタシステム概論 第 2 回 HA クラスタ	

2-6-応-1. クラスタシステムの種類(HA クラスタと HPC クラスタ)

HA クラスタと HPC クラスタについて、それぞれの概要と特徴、背景と歴史を説明する。また、各クラスタシステムの目的と位置づけを比較し、各システムに固有の技術を概説する。

【学習の要点】

* クラスタは、複数のコンピュータを結合して協調動作させ、ひとまとまりのシステムとして扱えるようにしたものである。
* 高可用性(HA)クラスタとは、複数のコンピュータを結合させることで、一台のコンピュータでは得られない高い信頼性を確保するものである。
* 高性能(HPC)クラスタは、複数のコンピュータで演算処理を分担し、一台のコンピュータでは得られない高い処理速度を確保するものである。

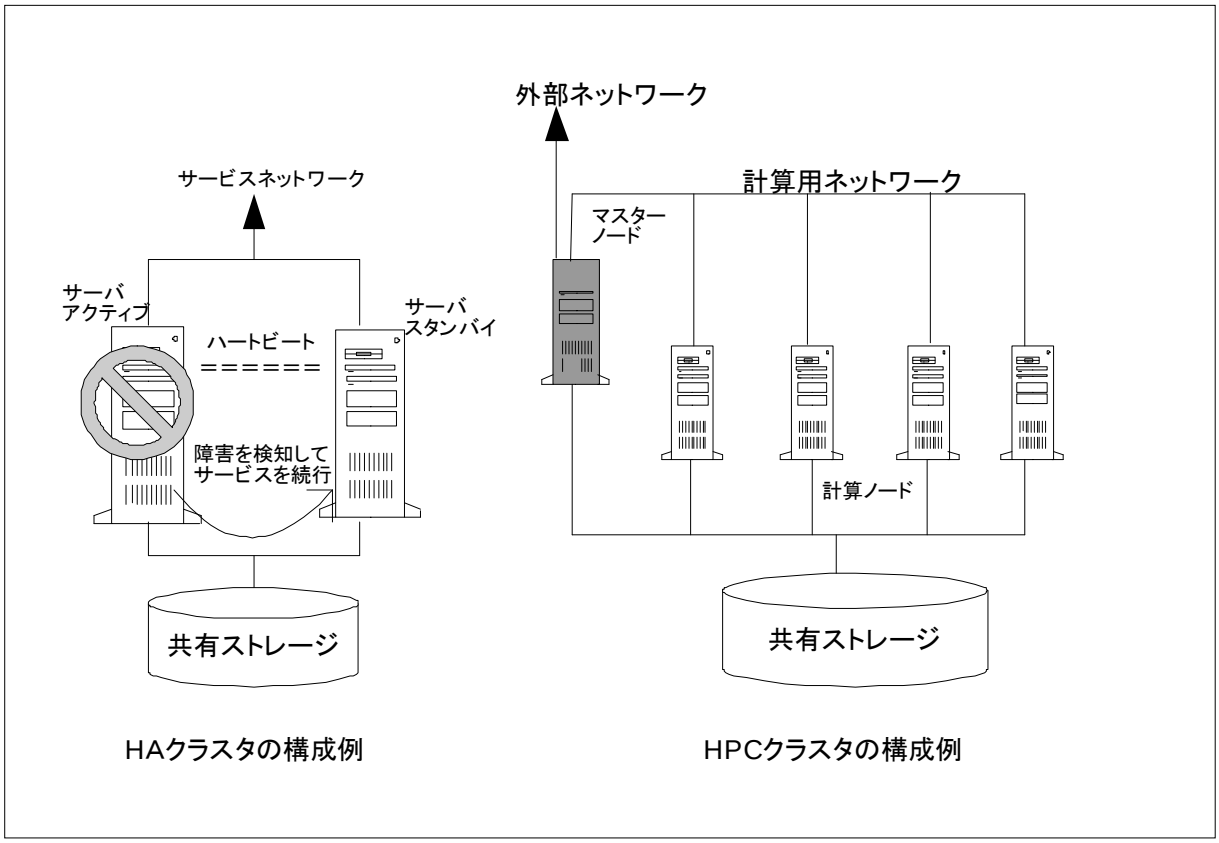

図 2-6-応-1.　HA クラスタと HPC クラスタ

【解説】

1) クラスタとは

クラスタ(コンピュータクラスタ、クラスタシステム)は、複数のコンピュータを結合して協調動作させ、ひとまとまりのシステムとして扱えるようにしたものである。クラスタを構成するコンピュータの一台一台のことをノードと呼ぶ。
クラスタには、目的別に HA(High Availability)クラスタと HPC(High Performance Computing)クラスタの2種類がある。

2) HA クラスタ(High Availability Cluster)

HA クラスタとは、複数のコンピュータを結合させることで、一台のコンピュータでは得られない高い信頼性を確保するものである。HA クラスタは、その実現方式により、フェイルオーバークラスタと負荷分散クラスタの2種類に分けられる。

＊ フェイルオーバークラスタ

フェイルオーバークラスタは、同一の役割を持たせた複数のコンピュータを、実際にサービスを提供する稼働系(現用系)ノードと、それをバックアップするための待機系ノードの2種類に分け、稼働系に障害が発生した場合に、待機系ノードを稼働系ノードに切り替えてサービスを継続することで、冗長性を高める。Active-Standby 型クラスタ、あるいは高可用密結合クラスタとも呼ばれる。伝統的には、単に HA クラスタと言った場合、このフェイルオーバークラスタを指す。

＊ 負荷分散クラスタ

負荷分散クラスタは、同一の役割を持たせた複数のコンピュータを、実際にサービスを提供する稼働系ノードとして並列に動作させ、一台が停止しても残りのノードでサービスを継続することにより、冗長性を確保すると同時に、複数ノードで処理を分担することにより、一台のサーバでは得られなかった処理性能を確保する。Active-Active 型、あるいは密結合並列クラスタとも呼ばれる。

3) HPC クラスタ(High Performance Computing Cluster)

HPC クラスタは、複数のコンピュータを結合させて演算処理を分担し、一台のコンピュータでは得られない高い処理速度を確保するものである。スーパーコンピュータの一形態であり、主に有限要素法や境界要素法等に基づく構造解析、金融工学、気象予測、シミュレーション天文学、分子動力学のような大規模数値解析に基づくシミュレーションなどに利用される。日本の文部科学省の科学技術・学術審議会では、2005年時点において、1.5TFLOPS 以上の演算性能を持つコンピュータを、政府調達におけるスーパーコンピュータとしている。ここで、FLOPS とは、Floating point number Operations Per Second の略であり、1秒間に浮動小数点数演算が何回できるかを表す、コンピュータの性能指標である。コンピュータの演算処理性能を議論・比較する際に一般的に用いられる。世界のスーパーコンピュータの演算性能ランキングとして、LINPACK ベンチマークによりランク付けした TOP500(http://www.top500.org/)が最も有名である。

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	応用
習得ポイント	2-6-応-2. Active-Standby 型 HA クラスタの構築	
対応する コースウェア	第 3 回　Active-Standby 型クラスタ	

2-6-応-2. Active-Standby 型 HA クラスタの構築

Heartbeat を利用して HA クラスタを構築する手法について説明する。その構成要素や、仮想 IP アドレスを共有した Active-Standby 型クラスタの構築方法について説明する。

【学習の要点】

* Heartbeat は Active-Standby 型の HA クラスタを構築するのに利用される。2 台のサーバが相互に状態を監視し、Active ノードに問題が見つかった場合に Standby ノードへサービスを引き継ぐ。
* Heartbeat でクラスタを構築するには、実サービス用の仮想 IP アドレスの他に、Active ノードと Standby ノード間でハートビート通信を行うための IP アドレスがそれぞれ 2 つ必要となる。
* Heartbeat の基本的なパラメータは ha.cf で設定する。監視パケットの送信間隔や、相手ホストがダウンしたことを判定する時間などを定義する。

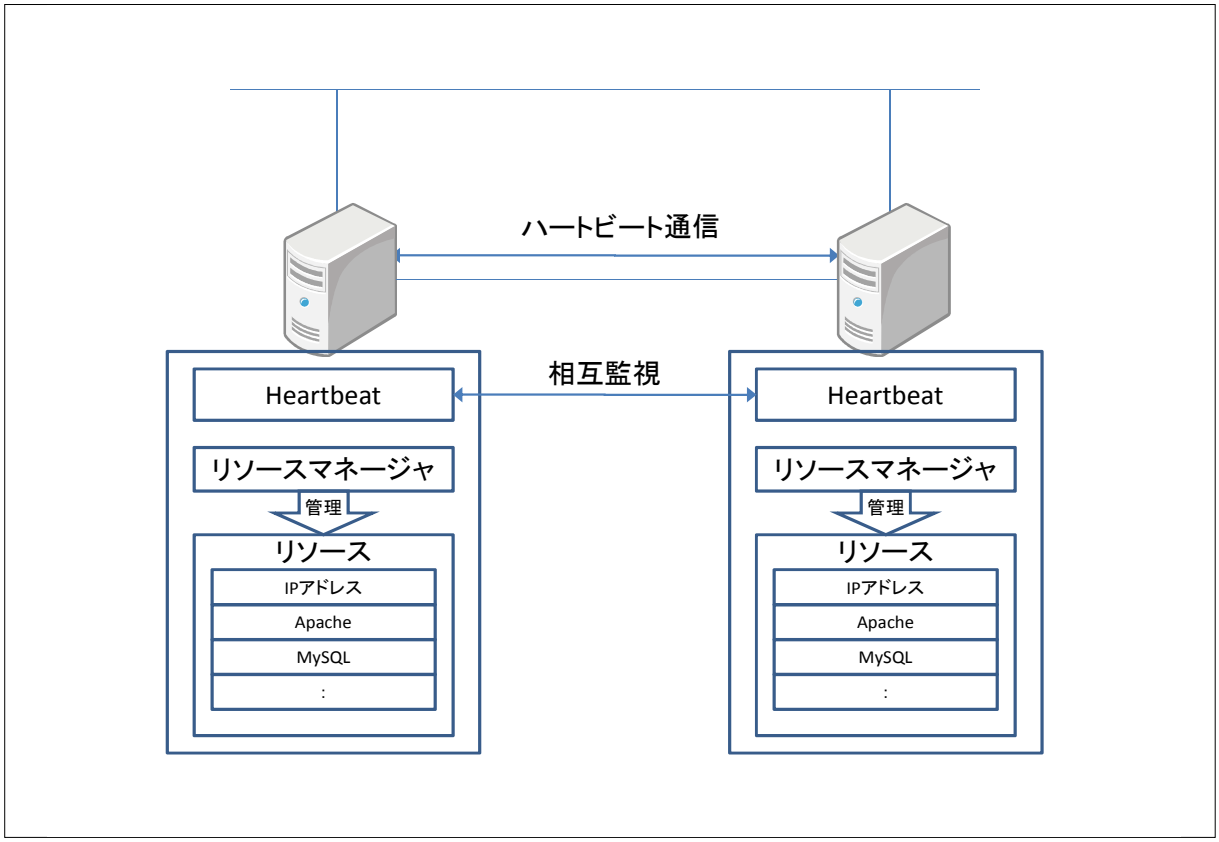

図 2-6-応-2.　Heartbeat の構成要素

【解説】

1) Heartbeat の概要

Linux で HA クラスタを実現するためのパッケージ群であり、Linux-HA プロジェクトにて開発されている。通称が Heartbeat で定着してしまっているが、プロジェクトでは Linux-HA という名称を推奨している。

Heartbeat は、クラスタに参加しているノードの死活監視を相互に行い、Active ノードが停止した際には Standby ノードを立ち上げる。通常時は、サービス提供に必要なプロセス群は Standby ノードでは立ち上がっておらず、切り替えのタイミングで起動される。

構成要素は以下の通りである。

* Heartbeat　参加ノードの相互監視を行うモジュール。
* リソース　サービス提供に関する要素。Web サーバや DB サーバ等が該当。
* リソースマネージャ　リソースのモニタリング、停止指示・起動指示を行う。
  途中のバージョンまでは CRM(Cluster Resource Manger)として
  実装されていたが、Pacemaker として独立プロジェクトとなった。
* リソースエージェント　各リソースとリソースマネージャの仲介役。
  実際にリソースの停止・起動を行うプログラム。

2) ハートビート通信の構成

サービス提供用通信との影響を避けるために、ハートビート通信用に物理的に別のネットワークを構成することが推奨される。Heartbeat は、以下の方法でハートビート通信を行うことができる。TCP/IP ベースの通信の場合は、UDP ポート 694 で通信する。

* serial　参加ノード間で、シリアルポートを相互に接続する。
* unicast　通信先 IP アドレスを指定し、ユニキャストで通信する。
* broadcast　接続セグメントのノード同士、ブロードキャストで通信する。
* multicast　接続セグメントのノード同士、マルチキャストで通信する。

3) 設定ファイル

Version2 ではリソースマネージャが導入されたこともあり、Version1 と Version2 では設定ファイルが大幅に違う。Version2 は柔軟に設定できるがその分複雑になるため、必要に応じ選択する。なお、haresources2cib.py スクリプトで、Version1 用の設定を Version2 の設定に変換できる。

Heartbeat は以下の設定ファイルを持ち、クラスタのメンバは同一の内容を設定する必要がある。

* /etc/ha.d/ha.cf　Heartbeat 全般の設定を行う。
* /etc/ha.d/authkeys　クラスタのメンバ間で認証を行うための情報を記述する。
* /etc/ha.d/haresouces　Version1 ベースのリソースの設定を行う。
* /var/lib/heartbeat/crm/*　Version2 ベースのリソースの設定を行う。
* /etc/ha.d/resource.d/*　各リソースごとのリソースエージェントが保存されている。

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	応用
習得ポイント	2-6-応-3. HDD のミラーリングによる冗長化	
対応する コースウェア	第 4 回　信頼性の確保	

2-6-応-3. HDD のミラーリングによる冗長化

DRBD（Distributed Replicated Block Device）を利用してネットワークを介して HDD をミラーリングを行う方法について説明する。

【学習の要点】

* DRBD（Distributed Replicated Block Device）とはネットワークを介して HDD のミラーリングする OSS である。Heartbeat と組み合わせることで共有ディスクを使わずに HA クラスタを構築するのに役立つ。
* DRBD を利用するにはあらかじめ fdisk でメタデータ領域と、データ領域のパーティションを切る必要がある。
* DRBD のパラメータは drbd.conf に設定する。メタデータ領域とデータ領域の指定や、ミラーリングのパラメータを定義する。

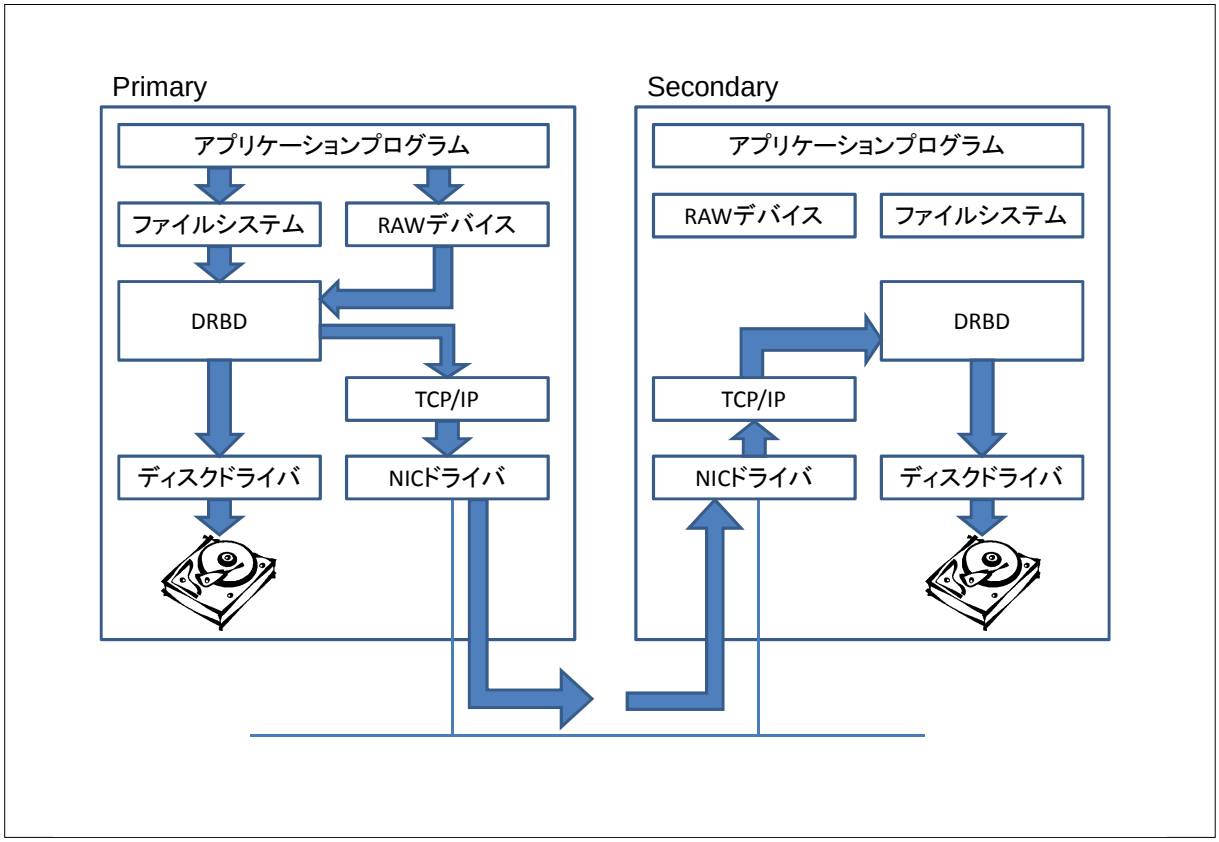

図 2-6-応-3.　DRBD によるデータ書き込み

【解説】

1) DRBD の概要

ネットワークを経由して、ディスクへの書き込み内容を逐次ミラーリングするプログラムである。ファイルシステムとディスクドライバの中間に介在し、カーネル内の機能として実装される。
ディスクには、通常書き込まれる実データと、DRBD の管理上必要なメタデータが書き込まれる。メタデータは、実データと同一のパーティションに書き込むだけでなく、別パーティションに書き込むこともできる。
Secondary 側ディスクは DRBD によりデータ書きこまれるため、通常はリードオンリーでマウントされる。Primary/Secondary ともに書き込み可能にすると、データの整合性がとれずにファイルシステムを破壊してしまう。

2) DRBD のインストール

DRBD はカーネルの機能として動作するため、バイナリパッケージにするのであればカーネルバージョンごとに作成しなければならない。それぞれの環境にあったバイナリパッケージが提供されているとは限らないため、その場合はソースからコンパイルする必要がある。

3) DRBD 関連ファイル

DRBD は以下のファイルで設定する。リソースと呼ばれる単位でボリュームを定義し管理する。なお、ここでいう「リソース」とは、Heartbeat でいうリソースと別物なので注意する。

* /etc/drbd.conf

DRBD は以下のコマンドを用いて、ディスクの制御等を行う。なお、/proc/drbd にて、ステータスを確認することもできる。

* drbdadm　　DRBD 全般の管理ツール。リソース単位で操作を行う。
* drbdsetup　　同期のバッファサイズ設定など、DRBD デバイスのセットアップを行う。
* drbdmeta　　メタデータの管理を行う。
* drbd-overview　　/proc/drbd とほぼ同様の内容を確認できる。

4) DRBD の構築

DRBD を構築するためには、空いているパーティションが必要である。構築の流れは以下の通り。

* /etc/drbd.conf を編集し、drbd リソースを定義する。
* fdisk で drbd 用のパーティションを作成する。
* drbdsetup で、パーティションの準備をする（メタデータ等の書き込み）
* プライマリとなるサーバで、drbdadm を用いてディスクを Primary に設定する。その後、自動で同期が開始される。
* プライマリとなるサーバで、DRBD デバイスにファイルシステムを作成する。

5) Heartbeat との連携

Heartbeat のリソース設定を定義する。Heartbeat Version2 系の設定を行いたい場合は、/etc/ha.d/haresoucesで DRBD を含めた定義を行った後、haresources2cib.py で定義を変換する方法が簡単である。

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	基本
習得ポイント	2-6-応-4 スーパーコンピュータの歴史とクラスタによる実現	
対応する コースウェア	第 1 回 クラスタシステム概論 第 5 回 HPC クラスタ	

2-6-応-4 スーパーコンピュータの歴史とクラスタによる実現

科学技術計算基盤としてのスーパーコンピュータについて、その概要と歴史、最近の動向を紹介する。さらにクラスタによるスーパーコンピュータの実現方法について触れ、PC クラスタに関するいくつかのプロジェクトを紹介する。

【学習の要点】

* スーパーコンピュータとは、その演算処理速度が、同世代の一般的なコンピュータに比べて非常に高速な計算機を指して言う。
* ベクトルプロセッサなどの高性能計算に特化した専用プロセッサではなく、ワークステーションやサーバに利用されるような汎用プロセッサを大量に並べることにより構成されたスーパーコンピュータは、HPC クラスタとも呼ばれる。

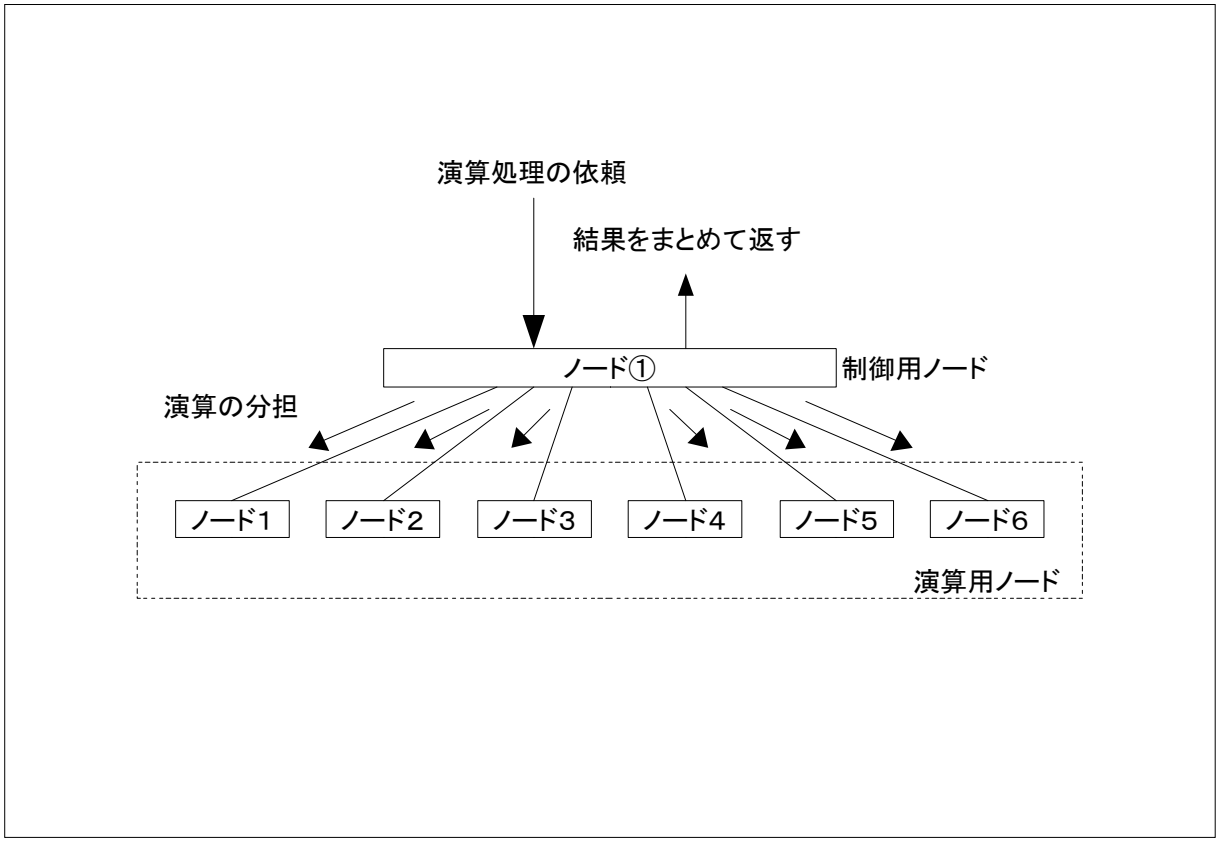

図 2-6-応-4.　HPC クラスタ概念図

【解説】

1) スーパーコンピュータの概要と歴史

スーパーコンピュータとは、その演算処理速度が、同世代の一般的なコンピュータに比べて非常に高速な計算機を指して言う。スーパーコンピュータは、主に有限要素法や境界要素法等に基づく構造解析、金融工学、気象予測、シミュレーション天文学、分子動力学、などのような大規模数値解析に基づくシミュレーションに利用される。日本の文部科学省の科学技術・学術審議会では、2005年時点において、1.5TFLOPS 以上の演算性能を持つコンピュータを、政府調達におけるスーパーコンピュータとしている。ここで、FLOPS とは、Floating point number Operations Per Second の略であり、1秒間に浮動小数点数演算が何回できるかを表す、コンピュータの性能指標である。コンピュータの演算処理性能を議論・比較する際に一般的に用いられる。世界のスーパーコンピュータの演算性能ランキングとして、LINPACK ベンチマークによりランク付けしたTOP500(http://www.top500.org/)が最も有名である。

一般にスーパーコンピュータの歴史は、学術系では 1971 年米イリノイ大学による ILLIAC の開発、産業系では 1976 年 Cray 社による商用機 Cray-1 の出荷がその始まりとされる。その後、1980 年代初頭に米国にてスーパーコンピュータが急速に発展した。それを追って日本も主なベンダとして NEC、日立、富士通がスーパーコンピュータ市場へ参入し、その価格性能比で世界を席巻した。その後、最先端のスーパーコンピュータ技術においては日米両国が突出した状態であるが、日本はバブル崩壊などの影響により、スーパーコンピュータへの投資が停滞し、1990 年代から 2000 年代にかけて、米国に大幅に遅れをとるまでになった。このような中、1996 年から計画され、2002 年に運用が開始された地球シミュレータは、科学技術庁により NEC をベンダとして開発され、TOP500 において、2002 年 6 月に第二位に5倍の性能差をつけて世界最速となり、2004 年 11 月に IBM が開発した Blue Gene に首位を奪われるまで、5 期に渡り第一位であった。

現在では、日本では汎用京速計算機と呼ばれる国家プロジェクトが、文部科学省主導の下産官学によって進められており、理論演算性能 10PFLOPS を目標とし、2012 年頃の完成および運用開始を目指している。米国でも、政府主導の下、各省単位で、数十 PFLOPS のスーパーコンピュータを 2010 年までに構築する計画が複数進められている。

2) クラスタによるスーパーコンピュータの実現

ベクトルプロセッサなどの高性能計算に特化した専用プロセッサではなく、ワークステーションやサーバに利用されるような汎用プロセッサを大量に並べることにより構成されたスーパーコンピュータは、HPC クラスタとも呼ばれる。ベクトルプロセッサは非常に高価である上に、高速化できる演算の種類が限られており、また汎用プロセッサの性能が劇的に向上してきていることから、近年では TOP500 にランクインするほとんどのスーパーコンピュータが HPC クラスタである。

3) 主要な HPC クラスタに関するプロジェクト

ここでは、HPC クラスタに関する主要なプロジェクトをいくつか挙げる。

* Beowulf
* Score
* TSUBAME
* PACS-CS

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	応用
習得ポイント	2-6-応-5. 並列計算機の種類と構成	
対応する コースウェア	第 1 回 クラスタシステム概論 第 5 回 HPC クラスタ	

2-6-応-5. 並列計算機の種類と構成

並列プログラミングの概念を理解するために、まず計算の高速化と並列処理の必要性を示す。また並列計算機の種類について触れ、プロセッサやメモリの結合方式、バスのトポロジーなど、構成方式の違いについて解説する。

【学習の要点】

* 近年のスーパーコンピュータは価格性能比の観点から、汎用プロセッサによるノードを多数結合して並列に動作させることにより、全体として演算処理性能のスループットを得る「HPC クラスタ」によるスーパーコンピュータが主流である。
* HPC クラスタの性能を引き出すためには、その上で動作するプログラムが並列処理に対応していなければならない。
* 並列計算機のメモリ結合方式には、共有メモリモデルである UMA と NUMA、メッセージ交換モデルである NORA がある。
* インターコネクトのトポロジーには、共有メモリ型ではバスベースおよびスイッチベース、分散メモリ型ではメッシュ型や動的ネットワーク型、完全結合ネットワーク型などがある。

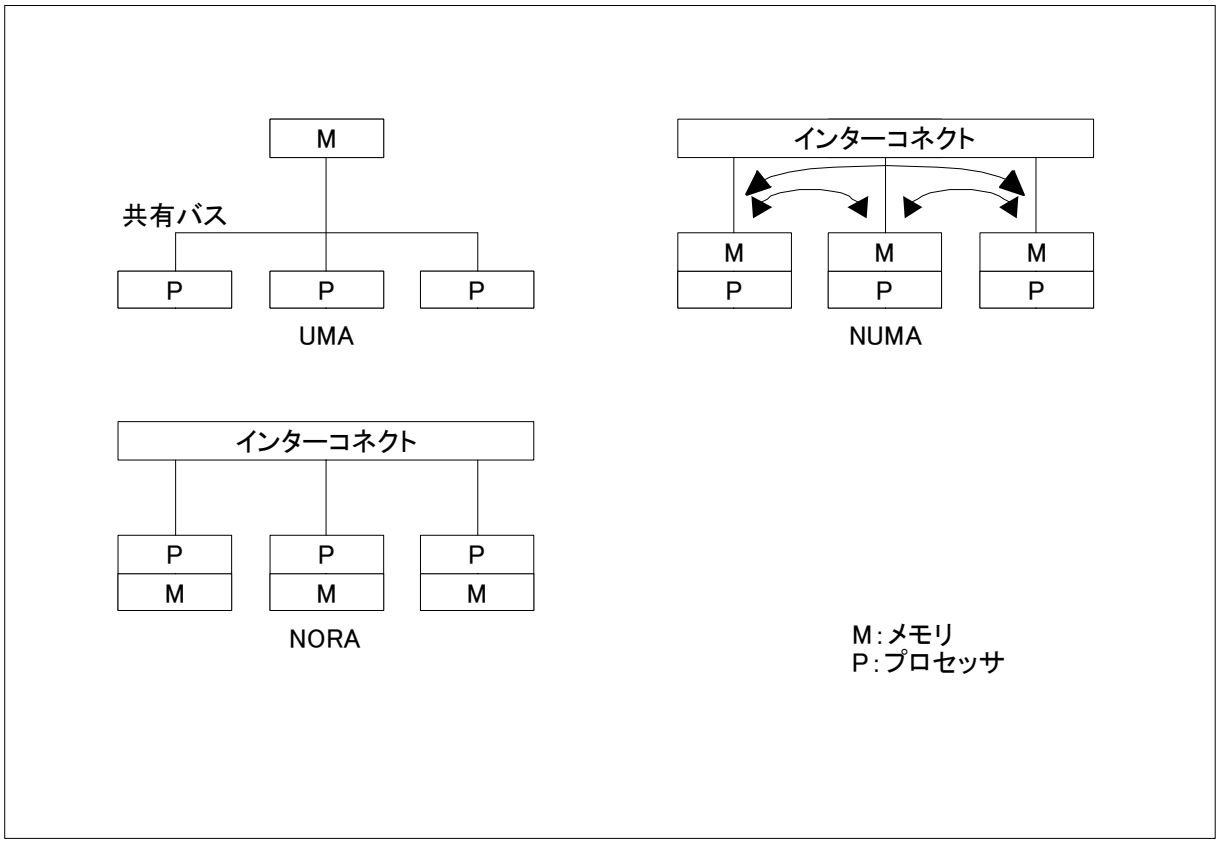

図 2-6-応-5.　HPC クラスタにおけるメモリの結合方式

【解説】

1) 計算の高速化と並列処理の必要性

軍事用途における核シミュレーションや、金融用途における株式市場シミュレーション、人体シミュレーションや気象予報計算、飛来する宇宙線の解析など、大規模計算を現実的な時間内に完了したい需要は大変に多く、計算機の高速化は国家レベルのプロジェクトが編成されるほどに重要である。このような中で、単一のコンピュータにより実現される演算性能には限界があり、近年のスーパーコンピュータは価格性能比の観点から、汎用プロセッサによるノードを多数結合して並列に動作させることにより、全体として演算処理性能のスループットを得る「HPC クラスタ」によるスーパーコンピュータが主流である。

HPC クラスタの性能を引き出すためには、その上で動作するプログラムが並列処理に対応していなければならない。例え 1024 ノードの HPC クラスタシステム上でプログラムを実行したとしても、そのプログラムが逐次処理にて演算を行うよう構築されている場合、実際に使用されるノードはそのうちの 1 ノードだけであり、他の 1023 ノードについては演算に使用されないことになる。

2) スーパーコンピュータの採用プロセッサ

1980 年代から 1990 年代までは、高性能計算に特化した専用のベクトルプロセッサを、各スーパーコンピュータベンダが独自に開発し、採用していた。ベクトルプロセッサは、行列演算等のベクトル計算を一度に行うことができ、科学技術計算分野の演算で高効率を発揮する。しかし、専用のベクトルプロセッサの開発には大変な費用がかかることと、ワークステーションやサーバ向けの汎用プロセッサが、パイプラインや SIMD 命令などベクトルプロセッサの機能を一部取り込むなどして、その性能を劇的に向上させてきたことから、近年では汎用プロセッサを多数並べた HPC クラスタが主流である。特に最近では、パーソナルコンピュータに広く採用されている、Intel x86 系 CPU の採用例も多くなっている。

3) メモリの結合方式

* 共有メモリモデル
  - UMA(Uniform Memory Access)
  - NUMA(Non-Uniform Memory Access)
* メッセージ交換モデル NORA(NO Remote-memory Access)

4) インターコネクトのトポロジー

* 共有メモリ型
  - バスベースアーキテクチャ
  - スイッチベースアーキテクチャ
* 分散メモリ型
  - 静的ネットワーク(メッシュなど)
  - 動的ネットワーク(クロスバー)
  - 完全結合型ネットワーク
  - オメガネットワーク
  - バスベースネットワーク

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	応用
習得ポイント	2-6-応-6.　PC クラスタの構築方法	
対応する コースウェア	第 12 回 Beowulf PC クラスタ	

2-6-応-6.　PC クラスタの構築方法

一般的な PC クラスタとして代表的なシステムである Beowulf 型の PC クラスタを解説する。同クラスタを構成する方法を示し、各種の設定手順と並列プログラムのコンパイルおよび実行例を示す。またベンチマークの方法も紹介する。

【学習の要点】

* Beowulf とは Linux マシンをノードとする分散メモリ型の PC クラスタシステムである。
* クラスタを構成する各 PC は Ethernet を介して接続され、NFS 経由で/home を共有し、rsh で通信を行う。このようにオープンソースプログラムと、安価な PC を利用できるために、非常に安くスーパーコンピュータと同等の計算環境を手に入れることができる。
* Beowulf における並列プログラミングは MPI を利用することが多い。
* Beowulf は、ファイルサーバ、マスタ演算ノード、スレーブ演算ノードから構成される。

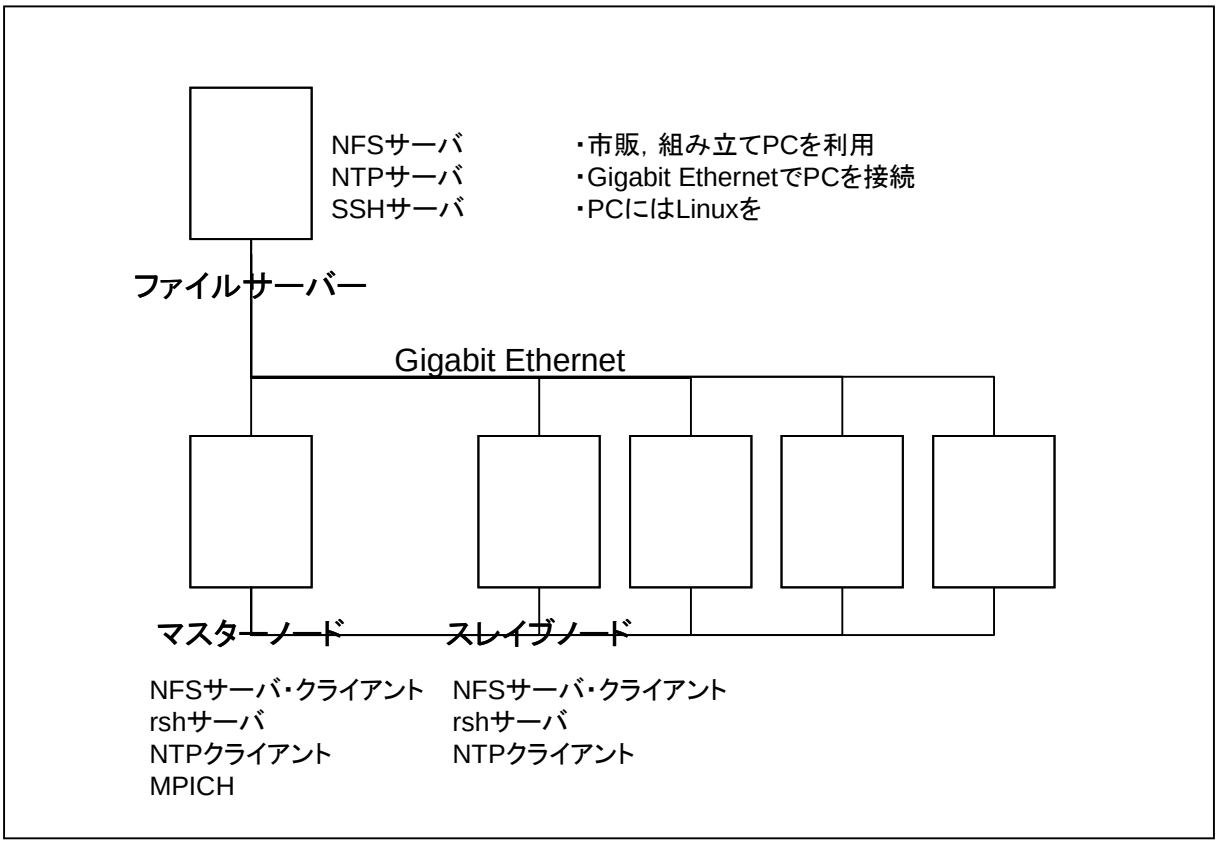

図 2-6-応-6.　PC クラスタの構築方法

【解説】

1) Beowulf PC クラスタとは

＊ Beowulf とは Linux マシンをノードとする分散メモリ型の PC クラスタシステムである。各 PC は Ethernet を介して接続され、NFS 経由で/home を共有し、rsh で通信を行う。

＊ 非常に安価にスーパーコンピュータと同等の演算能力を持つクラスタを構築することができる。

＊ Beowulf は、以下のノード構成となっている。

- ファイルサーバ
- マスタ演算ノード
- スレーブ演算ノード

＊ ファイルサーバは以下のサーバの設定が必要である。

- NFS サーバ
- NTP サーバ
- SSH サーバ

＊ マスタ演算ノードは、以下のサーバおよびクライアントの設定が必要である。

- NFS サーバ、クライアント
- rsh サーバ
- NTP クライアント
- MPICH

＊ スレーブ演算ノードは、以下のサーバおよびクライアントの設定が必要である。

- NFS クライアント
- rsh サーバ
- NTP クライアント

2) Beowulf におけるコンパイル、実行

＊ Beowulf における並列プログラミングは MPI を利用することが多い。たとえば、オープンソースの MPICH を利用すると、MPI 環境を非常に安価に構築することができる。

＊ Beowulf における並列プログラミングのコンパイル・実行例については、「MPI における並列プログラミング」と同じである。

3) PC クラスタのベンチマーク

＊ PC クラスタのベンチマークには以下のものがよく使われる。中でも姫野ベンチは短時間で実速度を求めることが出来るので、国内ではよく使われる傾向がある。

- 姫野ベンチ
  http://w3cic.riken.go.gjp/HPC/HimenoBMT
- ScaLAPACK
  http://www.netlib.org/scalapack/
- NAS Parallel Benchmarks
  http://www.nas.nasa.gov/Softwrare/NPB/

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	応用
習得ポイント	2-6-応-7.　HPC クラスタ向けユーティリティの利用	
対応する コースウェア	第 13 回 HPC クラスタの周辺技術	

2-6-応-7.　HPC クラスタ向けユーティリティの利用

バッチ処理システム、システム監視ツール、並列処理ライブラリ、並列プロファイラ、並列プログラム向けデバッガ、並列プログラム開発環境、並列プログラムのベンチマークソフトウェアなど、HPC クラスタで効果的に活用できる各種のユーティリティソフトウェアの機能と特徴を説明する。

【学習の要点】

＊ HPC クラスタを利用するためには各種のユーティリティを利用すると有効である。

＊ HPC のユーティリティとしては、バッチ処理システム、システム監視、並列ライブラリ、並列デバッガ・プロファイラ、開発環境、ベンチマークソフトウェアなどがある。

HPCクラスタ向けユーティリティ

ユーティリティ種類	
バッチ処理	OpenPBS TORQUE GridEngine PBS LSF
システム監視	Ganglia
並列ライブラリ	ScaLAPACK NAG Parallel Library
並列デバガ、プロファイラ	Intel Trace Coolector / Analyzer TotalView
開発環境	PTP(Parallel Tools Platform)
ベンチマークソフトウェア	**姫野ベンチ** ScaLPACK NAS Parallel Benchmarks SKaMPI Intel MPI Benchmarks

図 2-6-応-7. HPC クラスタ向けの主なユーティリティ

【解説】

1) バッチ処理システム

* OpenPBS（オープンソース）
  http://www.pbsgridworks.com/PBSTemp1.3.aspx?top_nav_name=Products&item_name=OpenPBS&top_nav_str=1&AspxAutoDetectCookieSupport=1
* TORQUE
  http://www.clusterresources.com/
* GridEngine
  http://gridengine.sunsource.net/
* PBS
  http://www.pbsgridworks.com/Default.aspx
* LSF
  http://www.platform.com/Products/platform-lsf-family/platform-lsf/product

2) システム監視

* Ganglia　http://ganglia.info/

3) 並列ライブラリ

* ScaLAPACK　http://www.netlib.org/scalapack/
* NAG Parallel Library　http://www.nag.co.uk/numeric/fd/fddescription.asp

4) 並列デバガ、プロファイラ

* Intel Trace Collector/Analyzer
  http://www.intel.com/cd/software/products/asmo-na/eng/306321.htm
* TotalView　http://www.totalviewtech.com/index.htm

5) 開発環境

* PTP (Parallel Tools Platform) : Eclipse のプラグイン　http://www.eclipse.org/ptp/

6) ベンチマークソフトウェア

* 姫野ベンチ　http://w3cic.riken.go.gjp/HPC/HimenoBMT
* ScaLAPACK　http://www.netlib.org/scalapack/
* NAS Parallel Benchmarks　http://www.nas.nasa.gov/Softwrare/NPB/
* SKaMPI　http://liinwww.ira.uka.de/~skampi/
* Intel MPI Benchmarks
  http://www3.intel.com/cd/software/products/asmo-na/eng/219848.htm

スキル区分	OSS モデルカリキュラムの科目	レベル
システム分野	2-6　クラスタサーバシステム構築に関する知識	応用
習得ポイント	2-6-応-8.　グリッドコンピューティング	
対応する コースウェア	第 14 回　グリッドコンピューティング	

2-6-応-8.　グリッドコンピューティング

グリッドコンピューティングの概念と各種のグリッドコンピューティングについて説明する。また米国、英国、欧州、アジア、そして日本におけるグリッドコンピューティングに関連する様々なプロジェクトを紹介する。

【学習の要点】

* グリッドコンピューティングは、並列分散処理の特殊な形態の一つである。コンピュータクラスタに比べて、より粗に結合され、地理的に離れた多様なコンピュータ同士が協調して計算するアーキテクチャである。
* グリッドコンピューティングの大ざっぱな分類としては、プロセッシンググリッド、データグリッド、ビジネスグリッドとなっている。
* グリッドコンピューティングはビジネスでの適用はもちろんのこと、重要な国家プロジェクトとして、世界各国で競争が行われている分野である。

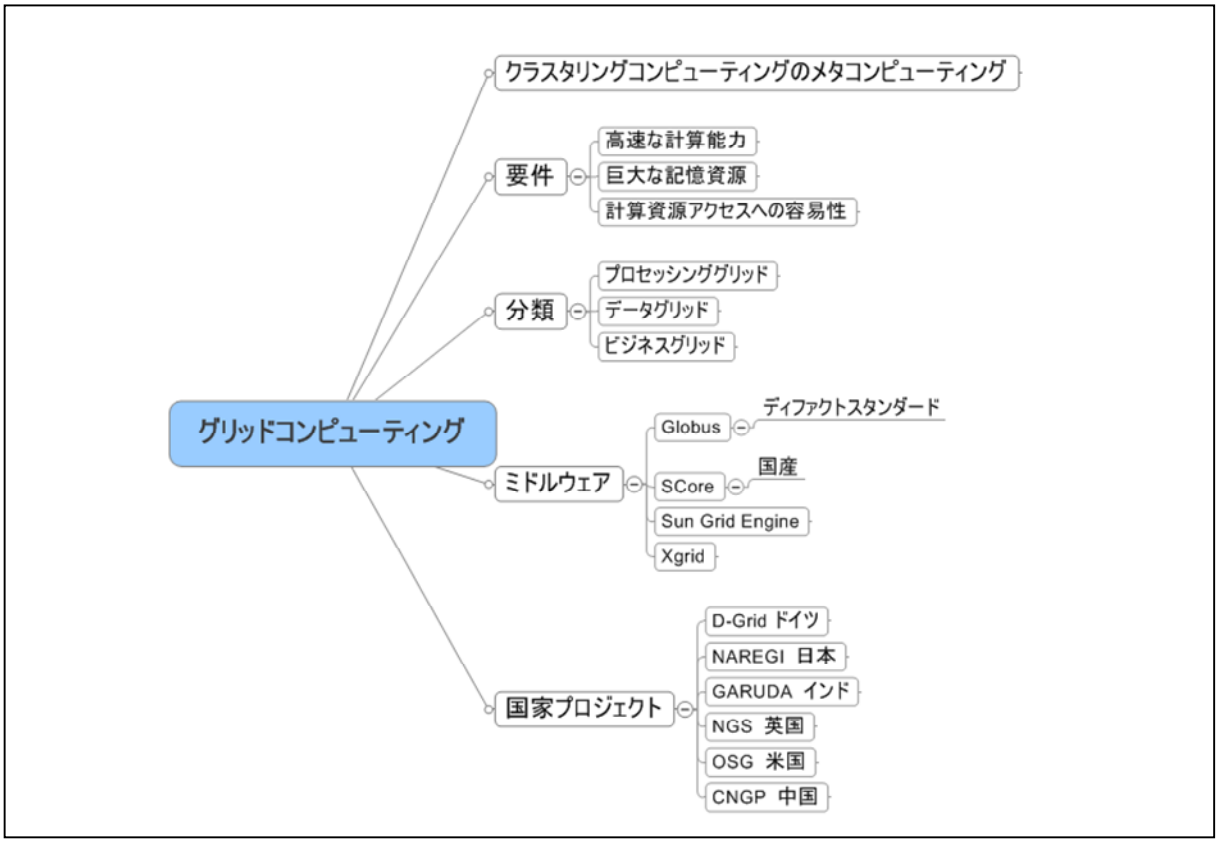

図 2-6-応-8.　グリッドコンピューティングのポイント

【解説】

1) グリッドコンピューティングとは

＊ クラスタリングコンピューティング（並列コンピューティングそして分散情報システム）とは、複数の計算機を組み合わせて、計算処理の向上を示すものである。これに対してグリッドコンピューティングとは、違ったクラスタリングコンピュータ同士さえも接続して、巨大なクラスタリング・クラスタリングコンピューティングを構築する仕組みである。

＊ グリッドコンピューティングは OS や物理的計算機の違いを吸収するために、ミドルウェアによって実現されることが多い。

＊ グリッドコンピューティングに必要な要件を以下に示す。

- 高速な計算能力
- 巨大な記憶資源
- 計算資源へのアクセスの容易性

2) グリッドコンピューティングの分類

＊ グリッドコンピューティングは、以下の種類に分類される。

- プロセッシンググリッド：計算処理をグリッド化
- データグリッド：データストレージをグリッド化
- ビジネスグリッド：ビジネスシステムグリッド化

3) グリッドコンピューティング ミドルウェア

＊ Globus ツールキット　　http://www.globus.org/

- 事実上のグリッドコンピューティングミドルウェアのデファクトスタンダードである。オープンソースである。
- プロトコルとしては、資源管理(GRAM、Grid Resource Management Protocol)、 情報サービス(MDS － Monitoring and Discovery Service)、データ移動と管理 (GASS － Global Access to Secondary storage)、そして FTP である GridFTP を提供する。

＊ SCore
「II-10-8 SCore の導入」を参照のこと。

＊ Sun Grid Engine

- Sun Microsystems が主導するオープンソースのグリッドコンピューティングミドルウェア

＊ Xgrid

- Apple が開発するプロプライエタリなミドルウェア。MacOSX 専用である。

4) 国家プロジェクトとしてのグリッドコンピューティング

＊ D-Grid (ドイツ)　　http://www.d-grid.de/

＊ National Research Grid Initiative（日本）　　http://www.naregi.org/

＊ GARUDA（インド）　　http://www.garudaindia.in/

＊ The National Grid Service (英国)　　http://www.grid-support.ac.uk/

＊ Open Science Grid(米国)　　http://www.opensciencegrid.org/

＊ China National Grid Project(中国)　　http://www.cngrid.org/web/guest/home