

Linuxクラスタ導入とそのキーポイント

NTT コムウェア株式会社
Linuxセンタ
チーフアーキテクト

大木 一浩

ooki.kazuhiro@nttcom.co.jp



NTT COMWARE

はじめに

- Linux はLinux Torvalds の米国およびその他の国におけるの登録商標あるいは商標です。
- UNIX は 米国およびその他の国における米国X/Open, Inc.の登録商標です。
- その他、記載されている会社名、製品名は各社の登録商標または商標です。



Agenda

- 市場動向
- クラスタについて
- システム方式設計に際して
- クラスタ適用事例
- Progart /Local Cluster Enterprise



市場動向

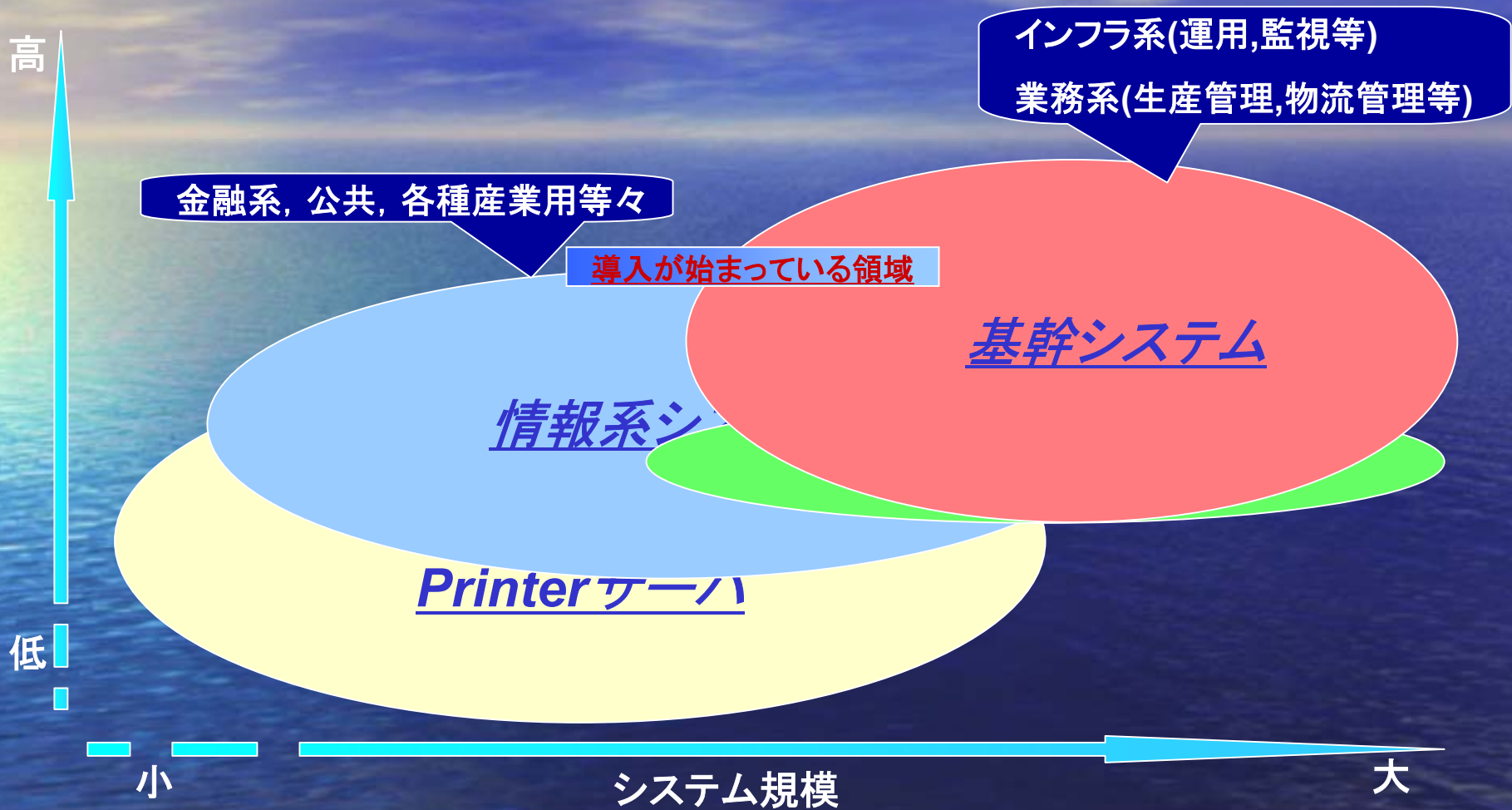
Co-Agenda

- Linuxの適用分野
- 受注内容分析（弊社分析）
- 各分野別企業動向
- 市場要望・要求の傾向
- システムアーキテクチャの動向・傾向
- 市場要望・要求に対する状況



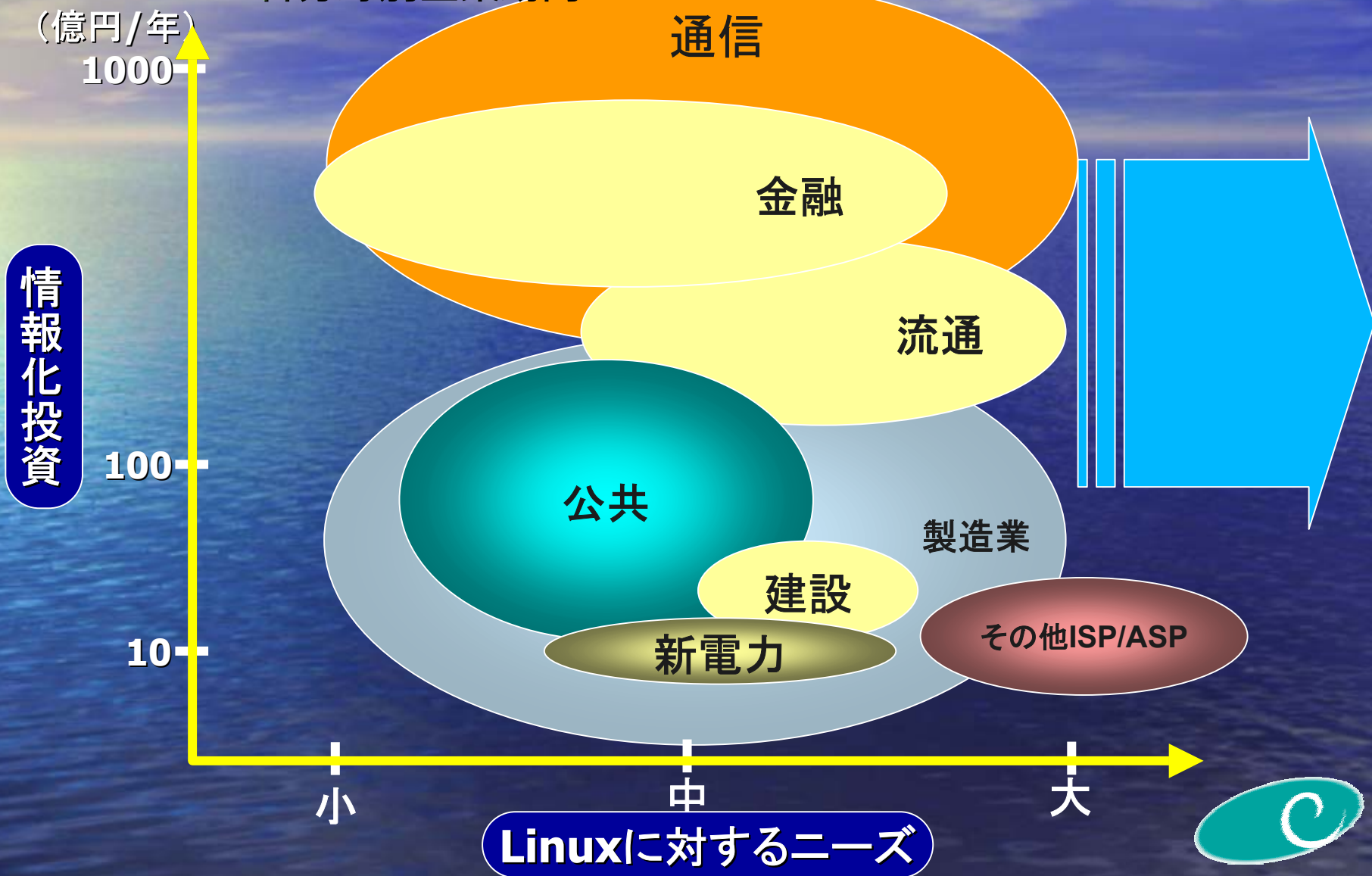
市場動向

—Linuxの適用分野



市場動向

—各分野別企業動向



NTT COMWARE

市場動向

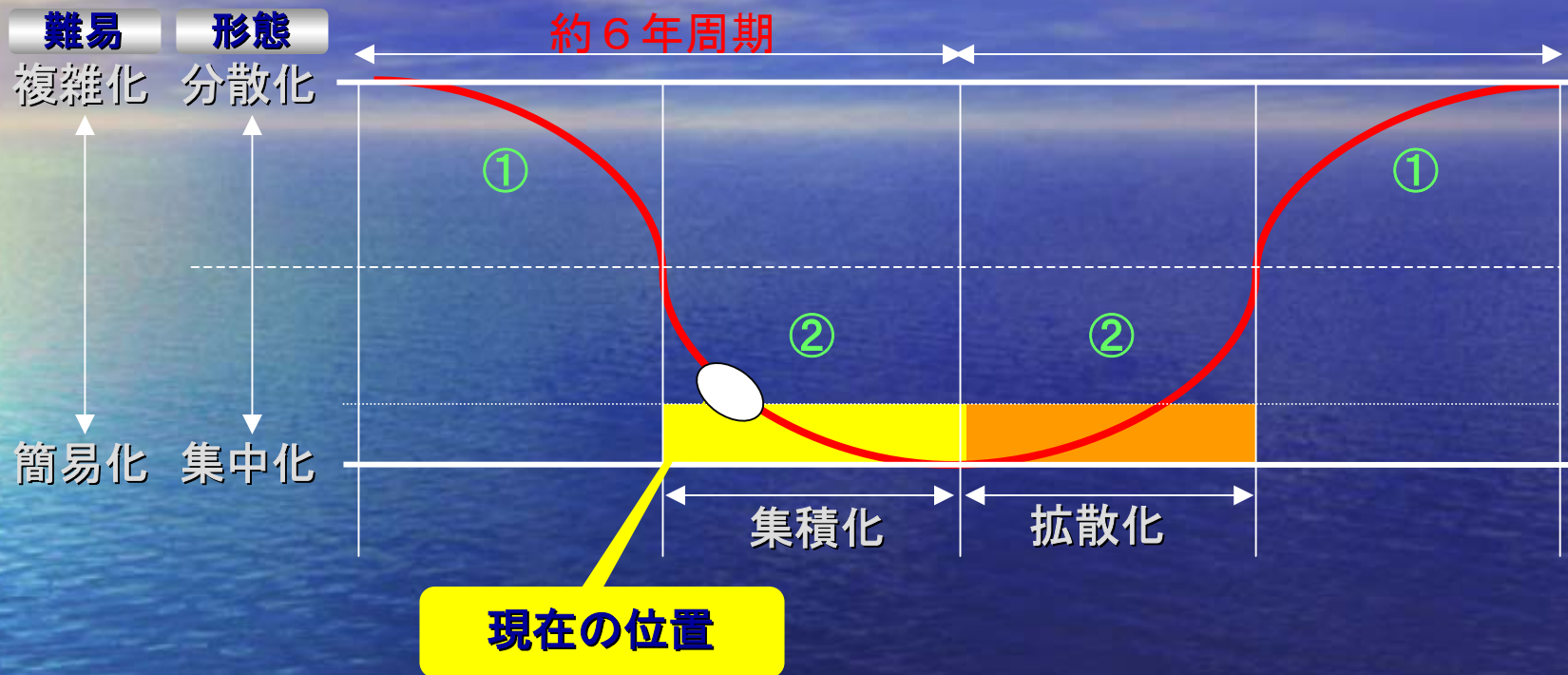
ー市場要望・要求の傾向

- 業務処理のクリティカル化（インフラ的システムの増加）
- 即時処理の主流化，サービス時間帯の延長（24h × 7day）
- 専用装置のオープン・システムへの移行，ダウン・サイジング
- システムライフサイクルの短縮化（システム改修・更改の短期化）



市場動向

ーシステム・パラダイムの動向・傾向



- ① 分散化に伴う問題発生の可能性の増大
構成物点数の増加
ソフトウェアの多様化
データ整合性保証方式の複雑化
マルチベンダ化
システム管理リソースの増大化

- ② 集中化に伴う問題発生の可能性の局所化
構成物点数の減少
ソフトウェアの単一化
データ整合性保証方式の容易化
システム管理リソースの縮小化



市場動向

ー市場要望・要求に対する状況

- コンピュータの信頼性保持レベルの差異，ばらつき
- オープン・システムの価格性能比の向上



クラスタについて

Co-Agenda

- 適用領域拡大への不可欠要素
- クラスタに対する要求
- クラスタの適用分野とメリット
- クラスタの種類



クラスタについて

—適用領域拡大への不可欠要素

中～大規模Webサーバ，基幹業務系/基幹設備系システム

高速性，高信頼性，高可用性，高拡張性の実現

高速処理への対応

CPUのフォルトトレラント化

ダウンタイムの軽減

柔軟な拡張性

障害に対する担保



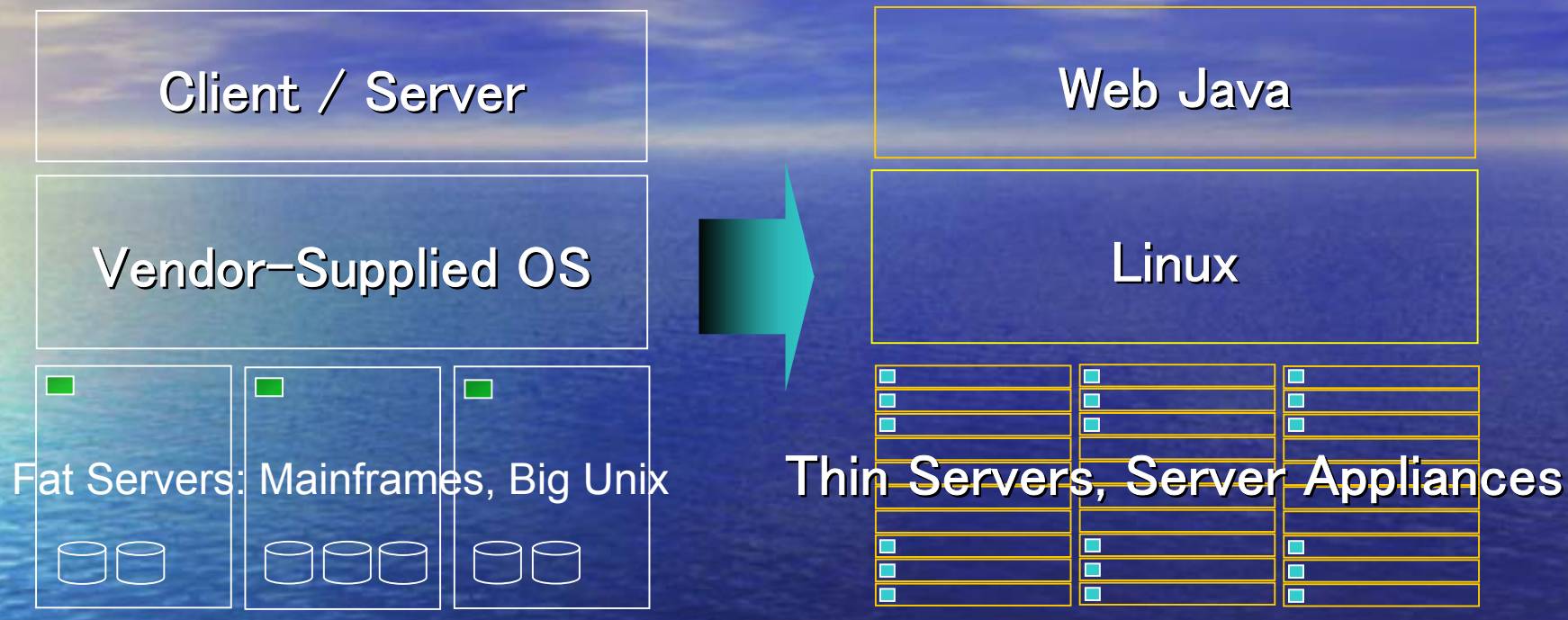
クラスタ技術



NTT COMWARE

クラスタについて

ークラスタに対する要求



高い可用性

高い拡張性

高い操作性



NTT COMWARE

クラスタについて

ークラスタの適用分野とメリット

- マシンのスケーラビリティ，耐障害性の向上手段として注目されている

従来 ： 科学技術計算の高速処理
 レンダリング処理などに適用

近年 ： Webサーバ，ISP，ASP，DBサーバなどさまざま
 な用途に適用

- 対価格性能比のメリット

高価なスーパーコンピュータと同等の性能がPCベースの
Linuxクラスタを適用することで実現、導入コストも
大幅に抑制



クラスタについて

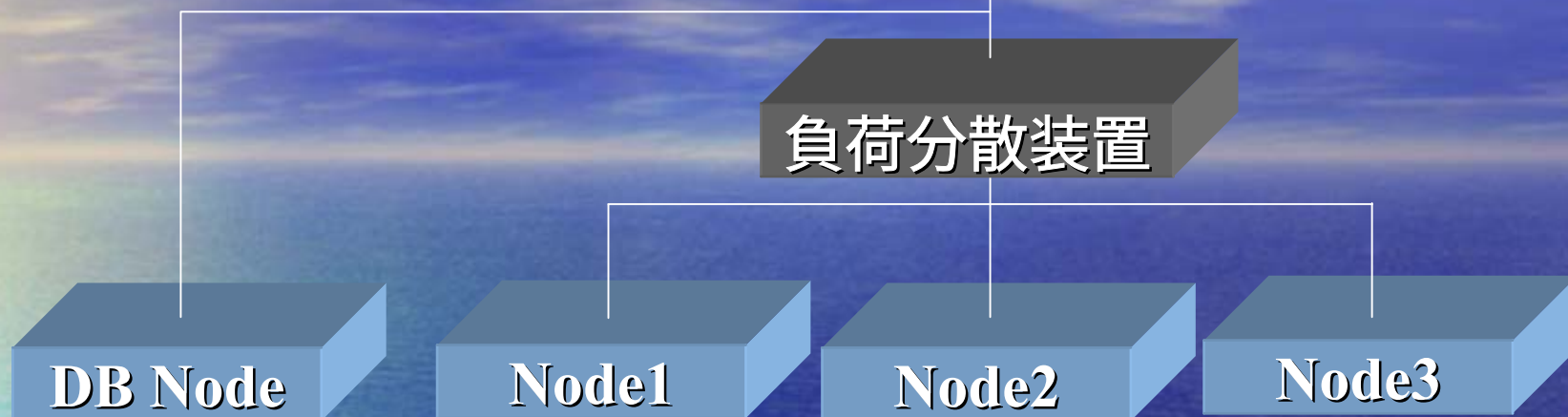
ークラスタの種類

- Load Balancer
- Beowulf type Cluster
- Enfusion type Cluster
- Failover Cluster
- Single System Image type Cluster
- その他



クラスタについて

— Load Balancer(ハードウェア実装)



【特徴 (●メリット・●デメリット)】

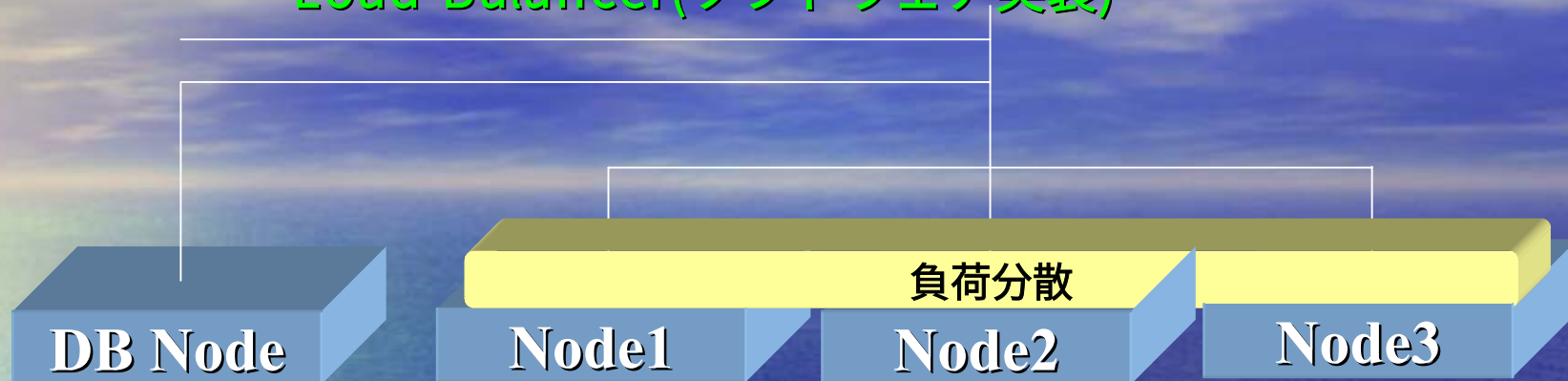
- 負荷の分散により、システム全体としての処理性能が向上
- 複数のサーバを冗長に利用することで高信頼性が得られる
- システム全体の拡張であれば、装置の増設が必要 (高価)
- マシンに非依存
- サーバを切り離しにより、運用中のシステム更改が可能



NTT COMWARE

クラスタについて

ーLoad Balancer(ソフトウェア実装)



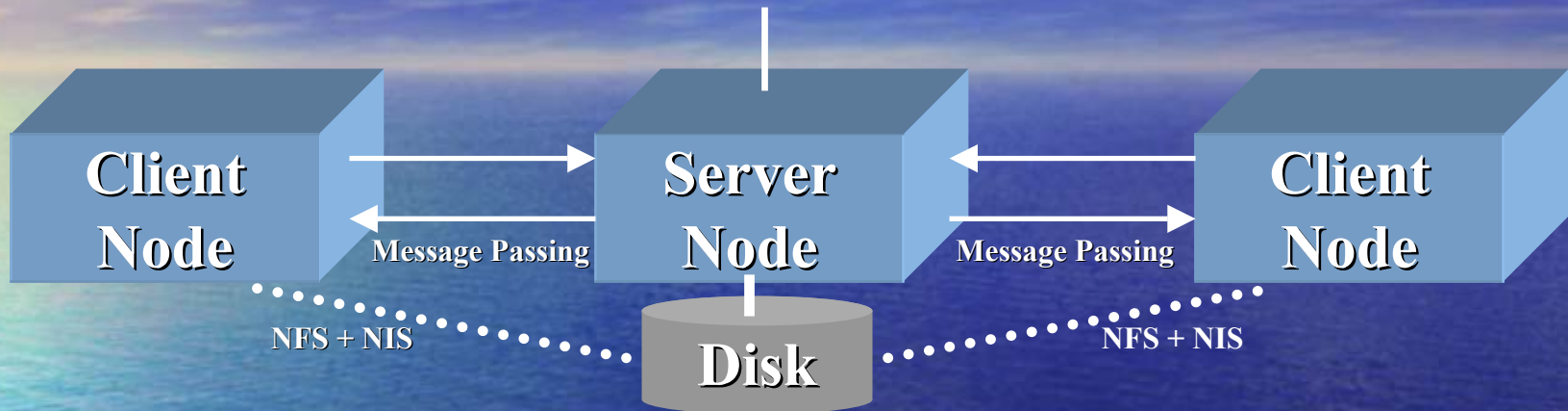
【特徴 (●メリット・●デメリット)】

- 負荷の分散により、システム全体としての処理性能が向上
- 複数のサーバを冗長に利用することで高信頼性が得られる
- システム全体の拡張が容易
- マシン,OSに依存
- サーバを切り離しにより、運用中のシステム更改が可能
- ハードウェアと比較し安価



クラスタについて

— Beowulf Type Cluster



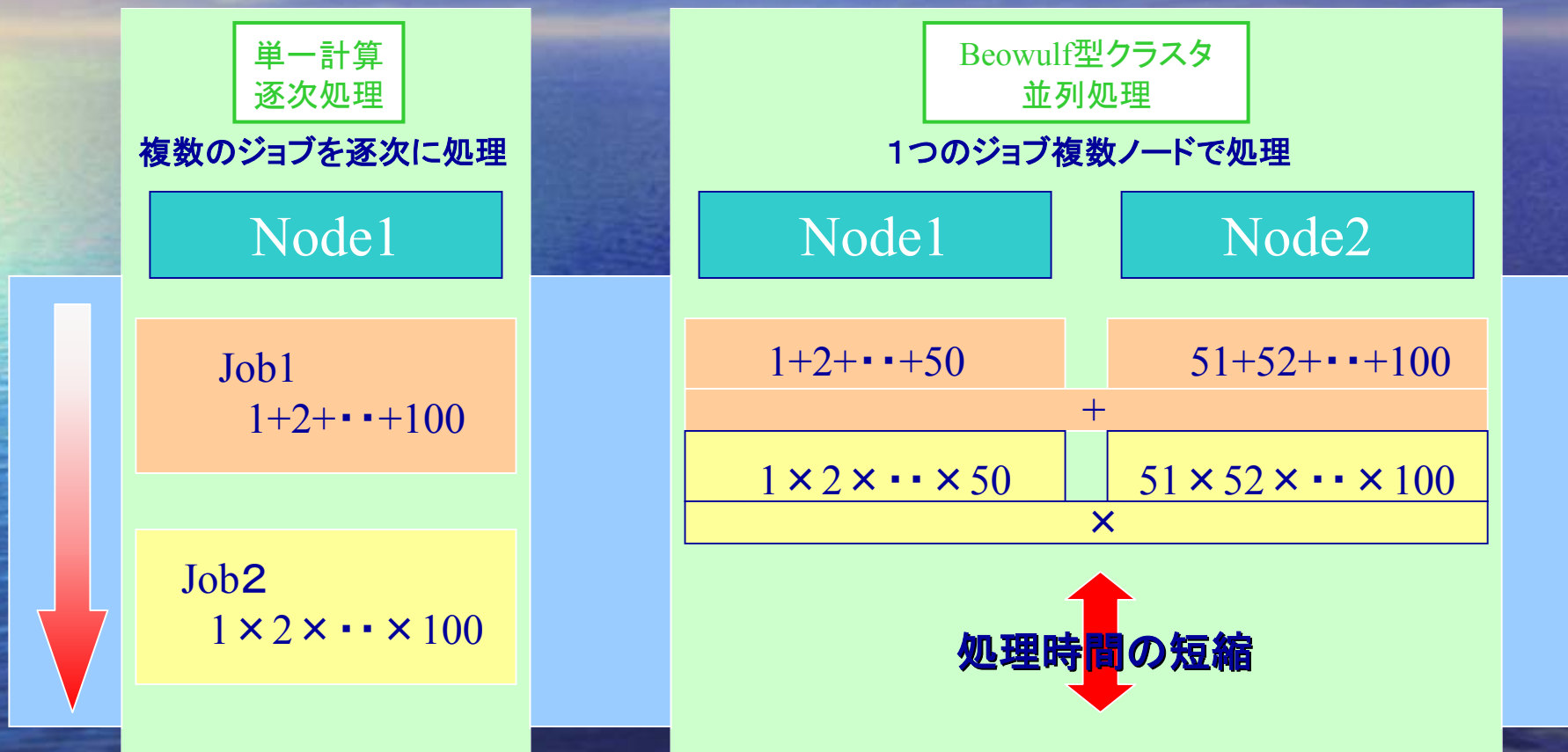
【特徴（●メリット・●デメリット）】

- オープンソースのみで構築が可能
- 計算処理の並列処理化が可能
- *Message Passing Programming Model* に強く依存するため、プログラムの並列化設計が必要
- データ保証機構がない（性能劣化の要因となるため）



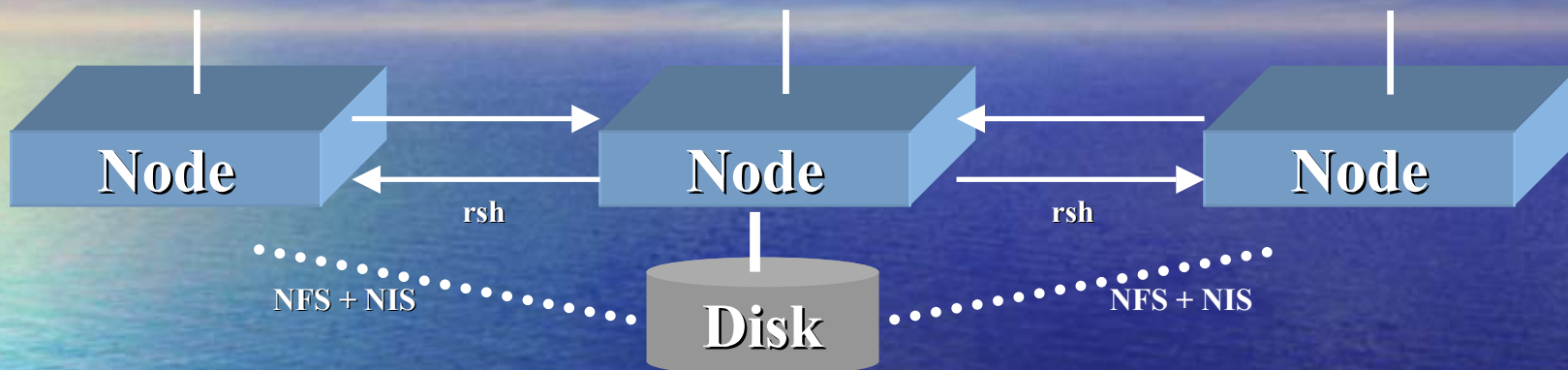
クラスタについて

— Beowulf Type Cluster



クラスタについて

—Enfusion Type Cluster



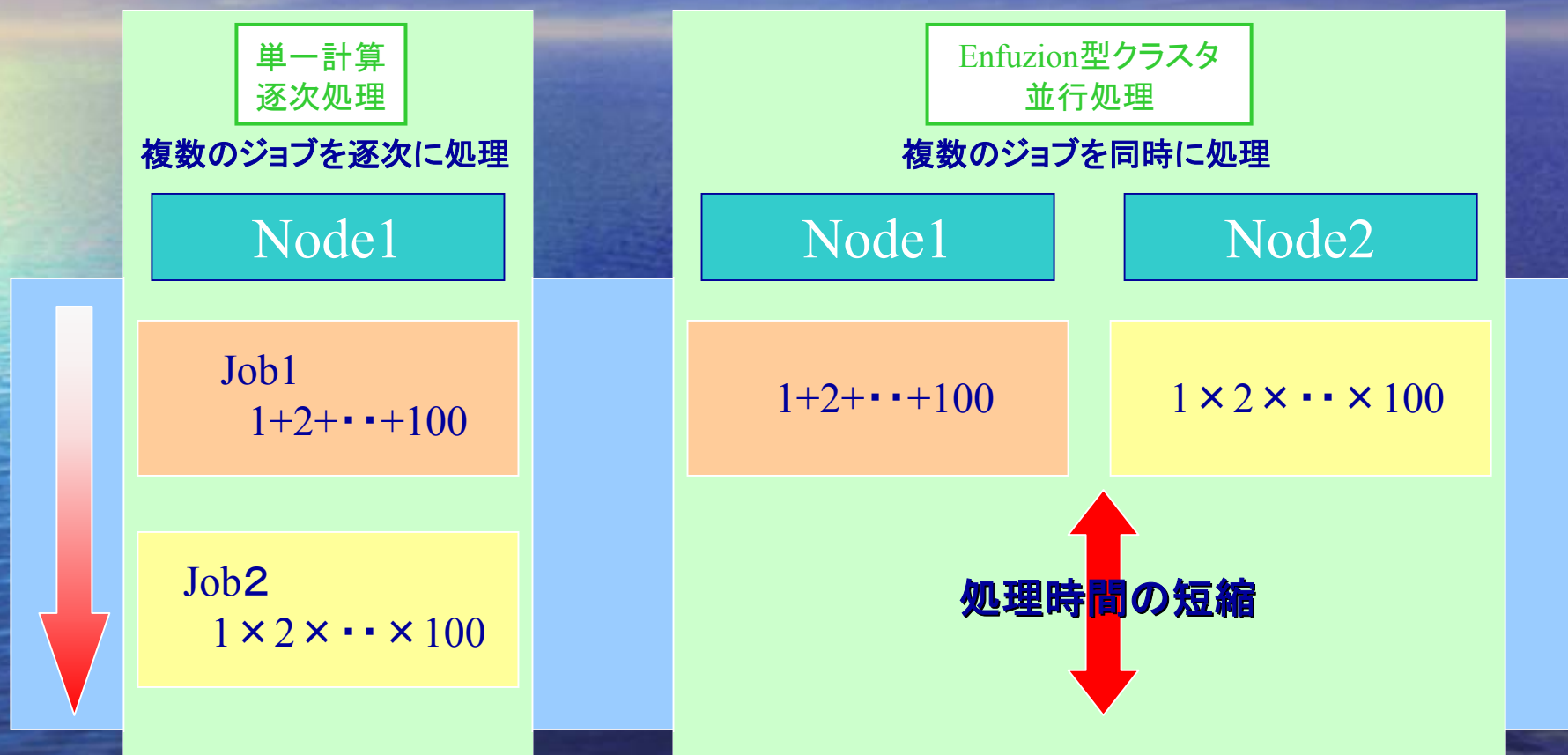
【特徴（●メリット・●デメリット）】

- 計算処理の並行処理化が可能
- データ保証機構がない（性能劣化の要因となるため）



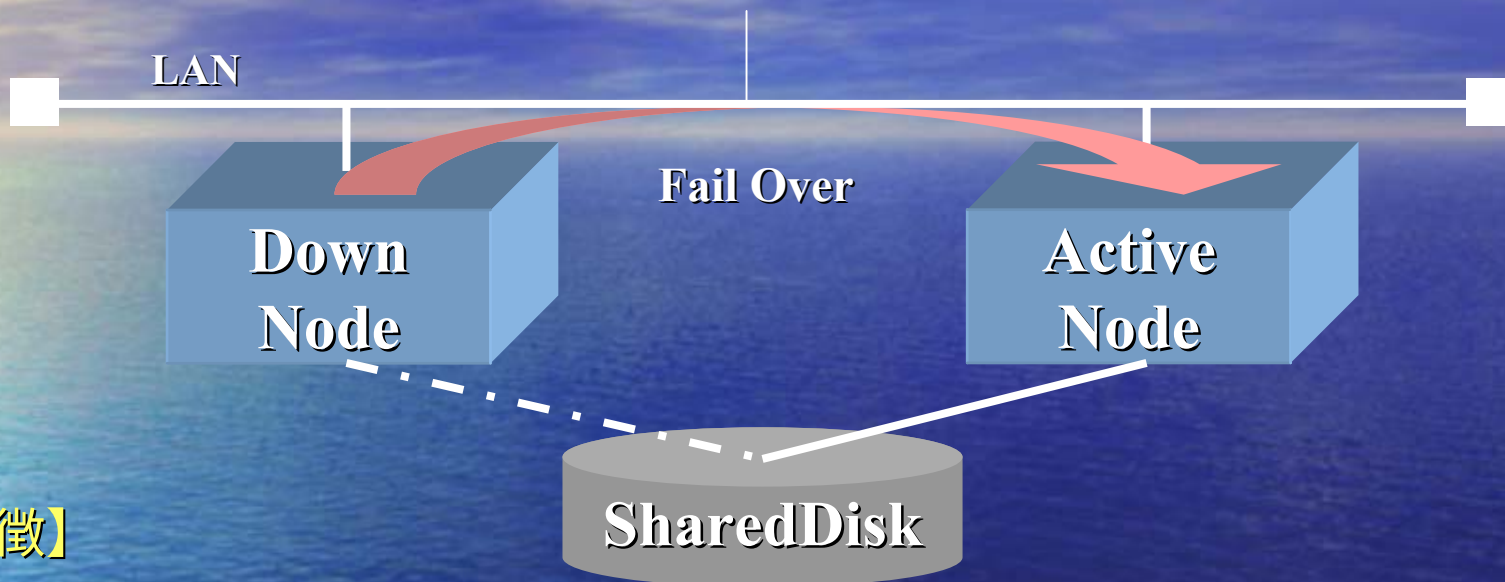
クラスタについて

—Enfuzion Type Cluster



クラスタについて

—Fail Over Cluster (共有ディスク使用)



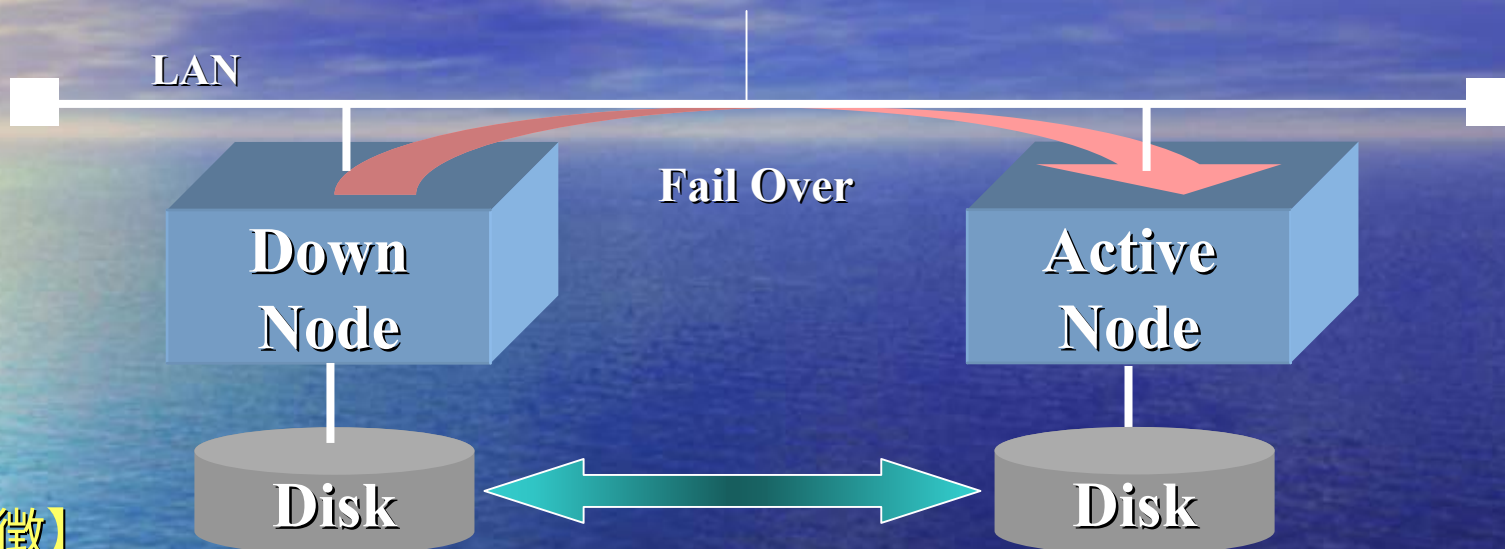
【特徴】

- 自動／手動フェールオーバー機構によりダウンタイムの軽減が可能
- Shared Diskの切り替え機能によりデータの引継ぎが可能
- ハートビート機構により各種の障害を検知可能
- デュプレックス（1＋1）構成を構築することが可能
- データ量が多いシステム向き



クラスタについて

—Fail Over Cluster (レプリケートタ使用)



【特徴】

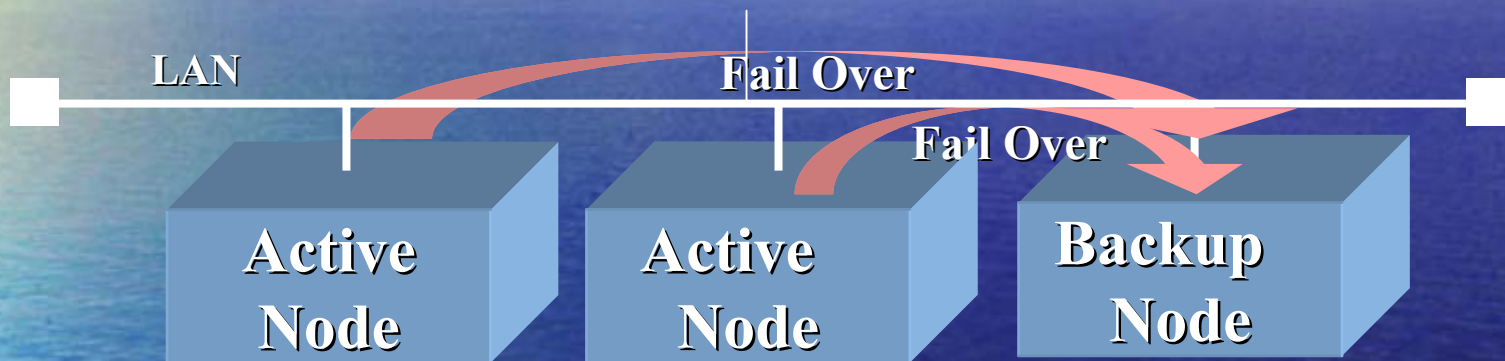
- 自動／手動フェールオーバー機構によりダウンタイムの軽減が可能
- レプリケーション機能によりデータの引継ぎが可能
- ハートビート機構により各種の障害を検知可能
- デュープレックス（1 + 1）構成を構築することが可能
- データ量が多くないシステム向き



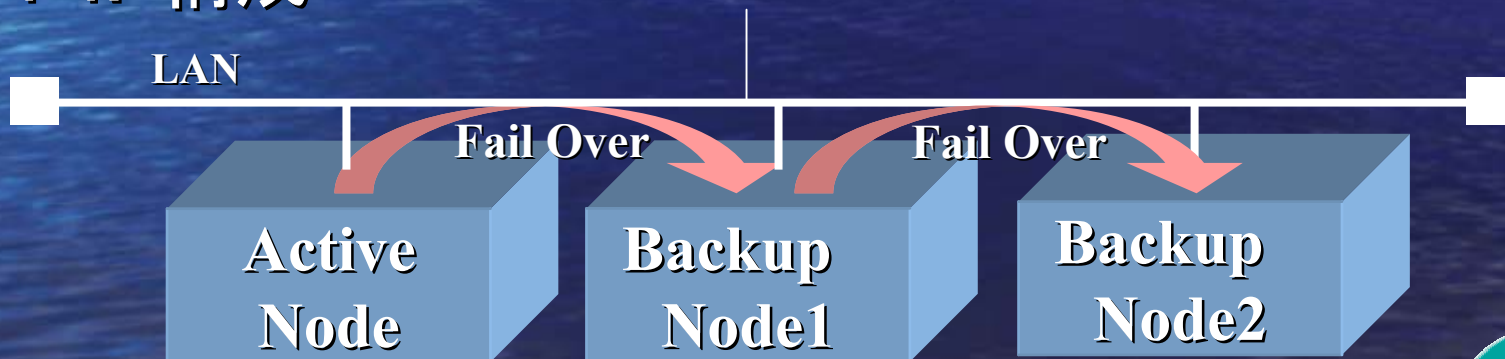
クラスタについて

— Fail Over Cluster (応用：N+1構成，1+N構成)

● N+1 構成



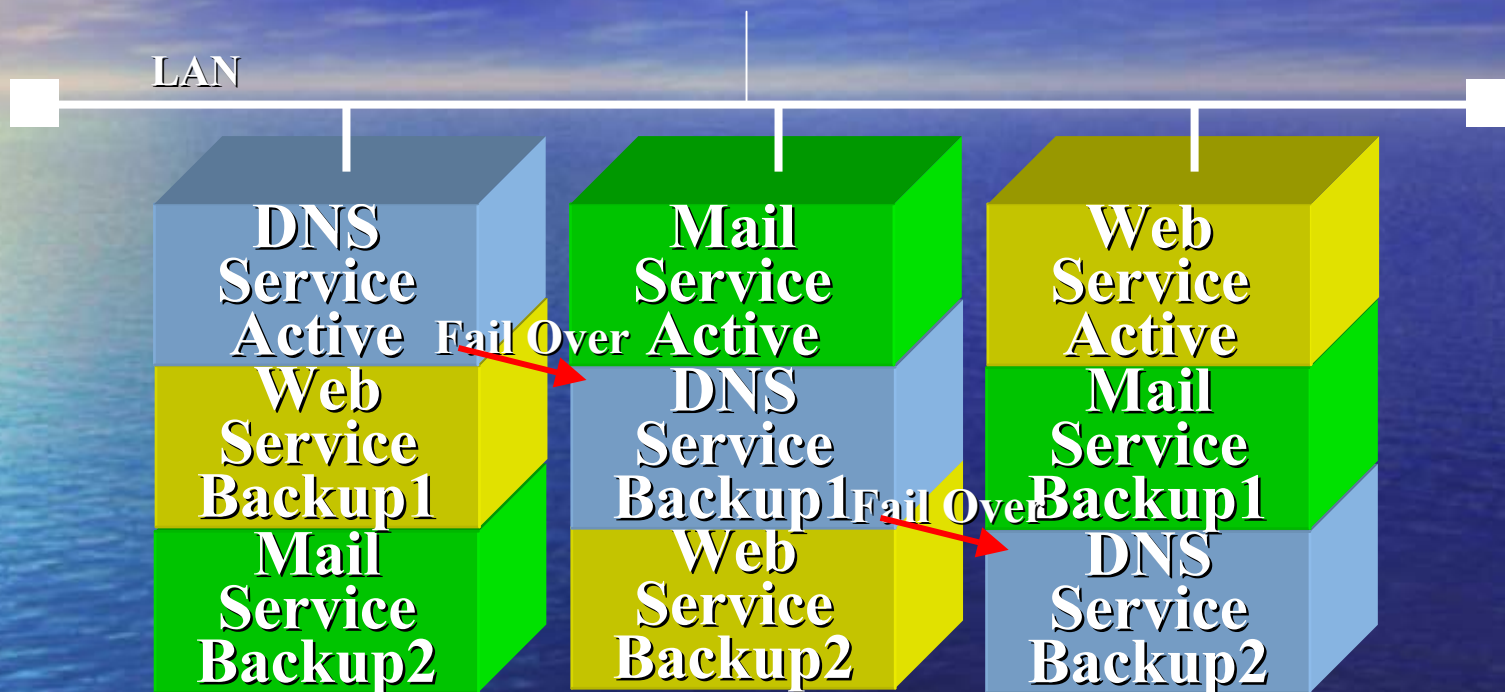
● 1+N 構成



NTT COMWARE

クラスタについて

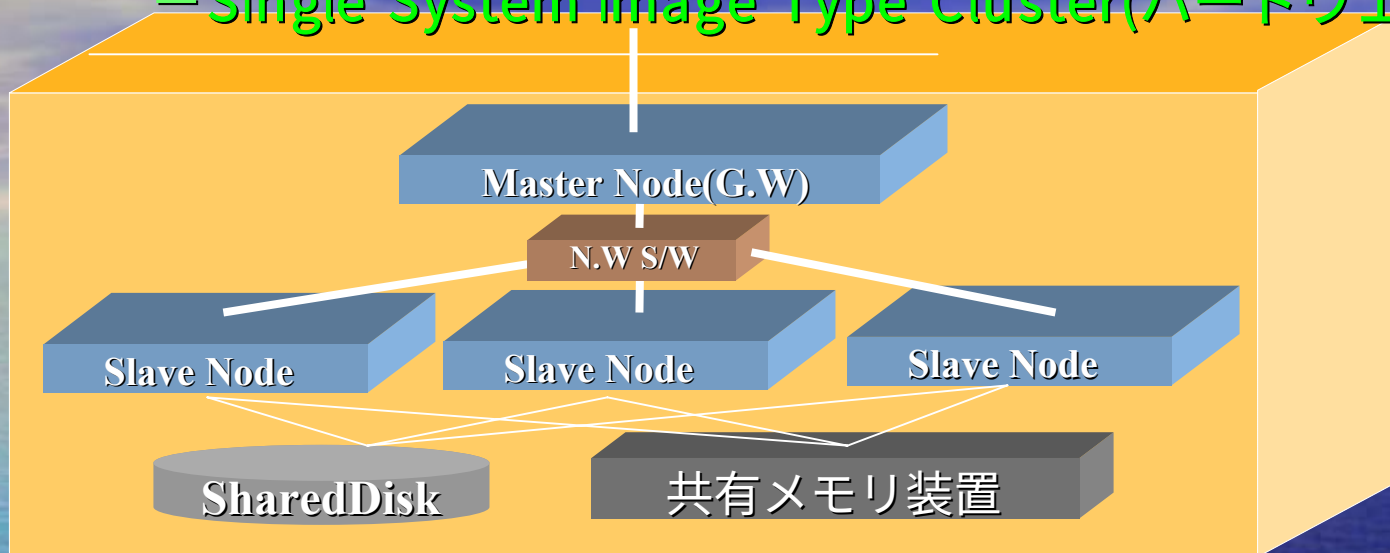
— Fail Over Cluster (応用：N+N構成)



NTT COMWARE

クラスタについて

—Single System Image Type Cluster(ハードウェア実装)



【特徴 (●メリット・●デメリット)】

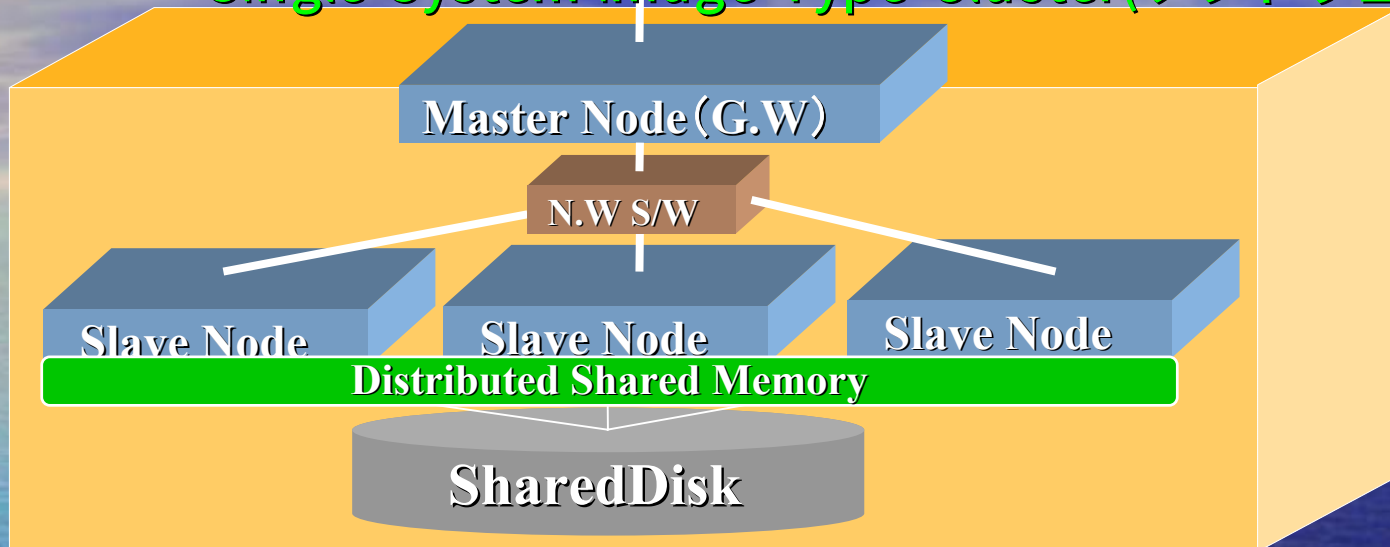
- 共有メモリ装置およびソフトウェアで実現し、外部/内部双方からもシングルマシンとしての操作性を提供
- CPUの効率的な利用により、システム全体としての処理性能が向上
- システム構成にハードウェア装置に依存する制限あり
- システム全体の拡張が容易
- マシン,OSに依存
- 専用ハードウェア必須であるため非常に高価



NTT COMWARE

クラスタについて

—Single System Image Type Cluster(ソフトウェア実装)



【特徴 (●メリット・●デメリット)】

- ソフトウェアのみで実現し、外部/内部双方からもシングルマシンとしての操作性を提供
- CPUの効率的な利用により、システム全体としての処理性能が向上
- 構成の自由度が増す
- システム全体の拡張が容易
- マシン,OSに依存
- ソフトウェアのみの実装により安価



クラスタについて

—その他

- Distributed IPC (DIPC)
- Distributed Shared Memory (DSM)
- NetworkRAM (nearly DSM)
- Score+, Parastation2



クラスタについて

ー 各クラスタの用途

- Load Balancer

I S P, A S P 等の瞬時に大量のトラフィックが発生するサイトでの使用に適している

- Beowulf type Cluster

解析処理, レンダリング等の大量の論理演算処理を必要とするシステムでの使用に適している

- Enfusion type Cluster

多量の論理演算処理を必要とするシステムでの使用に適している

- Fail Over Cluster

システムのダウンタイム軽減を目的とする使用に適している

- Single System Image type Cluster

シングルマシンで実装することが出来る全てのものに適している



システム方式設計に際して

Co-Agenda

- 信頼性に対する考慮
- 可用性に対する考慮
- 性能に対する考慮
- 拡張性に対する考慮
- 運用・保守性に対する考慮
- セキュリティに対する考慮
- 主要特性の相互関係
- システム停止の原因と解決策



システム方式設計に際して

-信頼性に対する考慮

企業の平均システム停止損益 (1,008万円/h)

(単位：百万)

業務	業界	
証券業務	金融	670～870
カタログ販売	小売	7.2～14.4
ペーパービュー	通信	8.1～28
チケット予約	運輸	8.1～14
クレジットカード ／販売承認	金融	260～370

出展：Contingency Planning Research and Dataquest, November 1997.”

信頼性に対する考慮は企業活動に大きく影響



NTT COMWARE

システム方式設計に際して

-信頼性に対する考慮

※ 信頼性 = システムが故障を起こさずに正しく稼働している度合

● ハードウェア障害に対する対策

- オープン製品の適用（特殊なインタフェース仕様を実装している製品の排除）
- 各部位の多重化（Disk, Ether, Server, ネットワーク経路など）
- UPSの適用
- など

● ソフトウェア障害に対する対策

- システムダウンに対する考慮（System Recovery）（ハードと関連）
- プロセスアボートに対する考慮（Process Recovery）（HA機能など）
- トランザクションアボートに対する考慮（Transaction Recovery）（APでの対応）

● データ保持・保証に対する対策

- 更新データの有無
 - データ更新頻度
 - トランザクションデータ欠損の許容可否
-
- ・ スタンバイ形態の選択
 - ・ データバックアップ方式
 - ・ ファイルシステムの選択



システム方式設計に際して

- 可用性に対する考慮

※ 可用性 = システムが利用できる状態を保つことのできる 度合い

● システムに要求されるミッション（可用性の要求レベル）

● ダウンタイムの定義（システム停止に対する考え方）

「停止を許容」

「特定の期間・時間帯のみ停止を許容しない」

「24h × 365dayの継続稼働」

● 許容される復旧時間は？

ゼロ, 数秒, 数十秒, 数分, 数十分

● 環境に関する障害対策

● ロケーション分散配置に対する考慮

● バックアップに対する考慮



システム方式設計に際して

- 可用性に対する考慮

● 運用時間, 必要可用性, 許容ダウンタイム (業種, 企業規模により変動)

	特定用途				汎用用途	
	人事・経理・財務系	コミュニケーション系	対外系	情報系	オンライン系	バッチ系
運用時間	8～22時	24時間	24時間	8～22時	8～20時	20～8時
必要可用性	99.00%	99.90%	99.00%	95.00%	99.90%	90.00%
許容ダウンタイム	4時間	3時間	6時間	2時間	15分	3時間

出展: IAサーバ・HAガイドライン実施手順書 Ver.1, 1999.3, ECA協議会



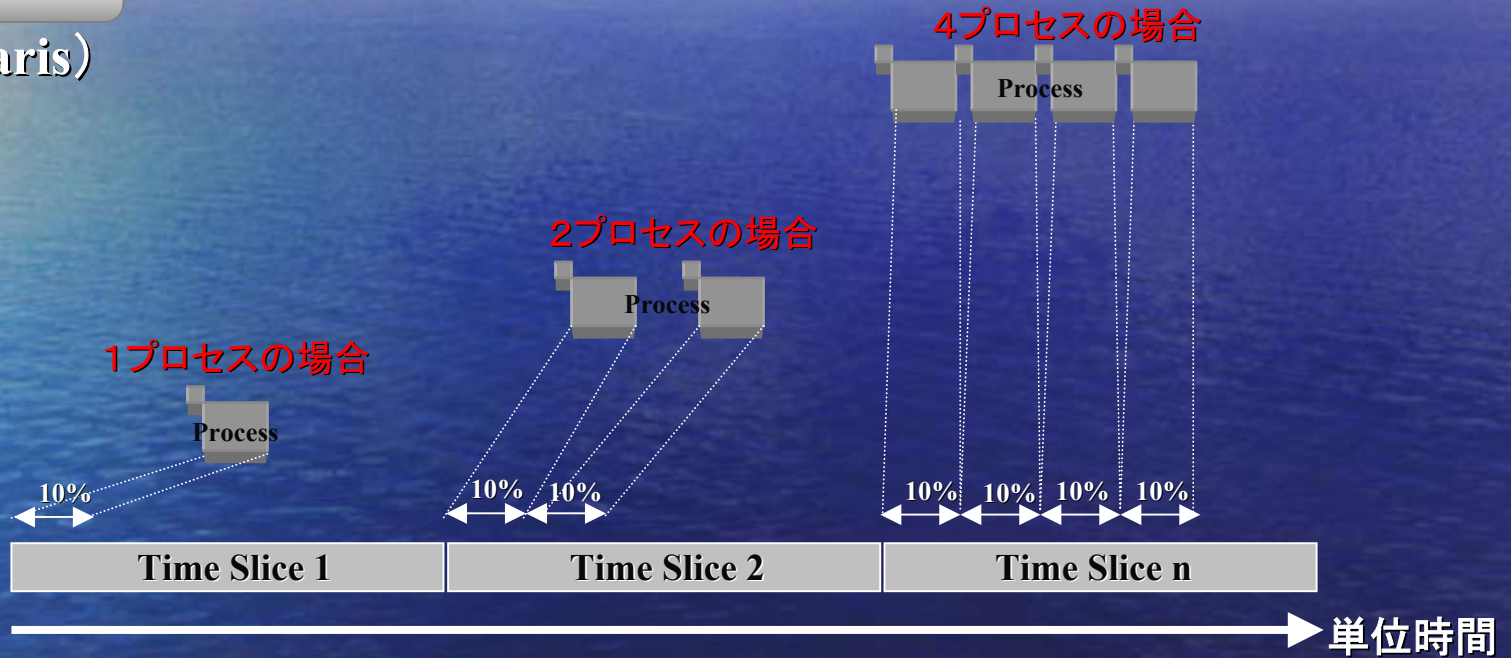
NTT COMWARE

システム方式設計に際して

一性能に対する考慮（プロセススケジューリング方式）

商用UNIX

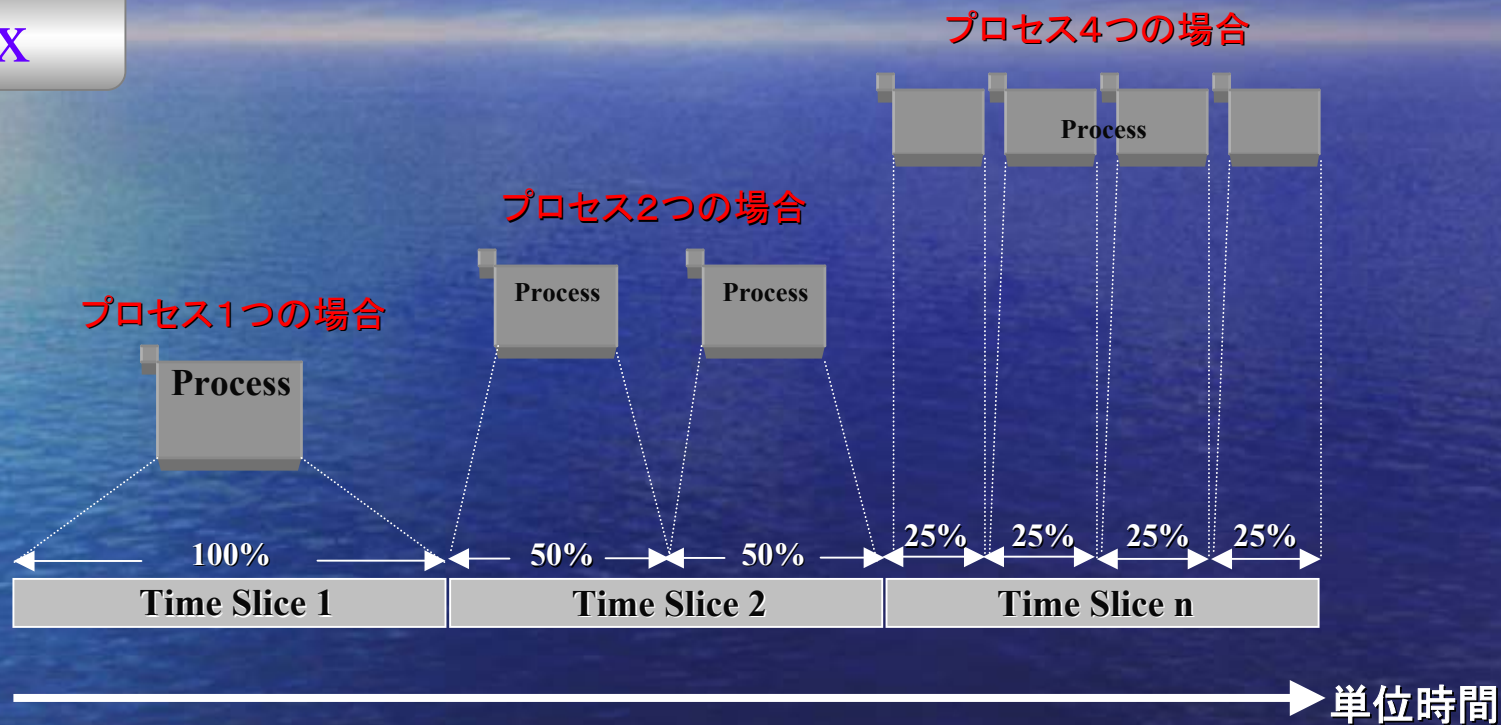
(Ex. Solaris)



システム方式設計に際して

ー性能に対する考慮（プロセススケジューリング方式）

Linux



- ・ CPU時間が均等に割り当てられる
- ・ 速さを追求したプロセッシング処理を指向する方式



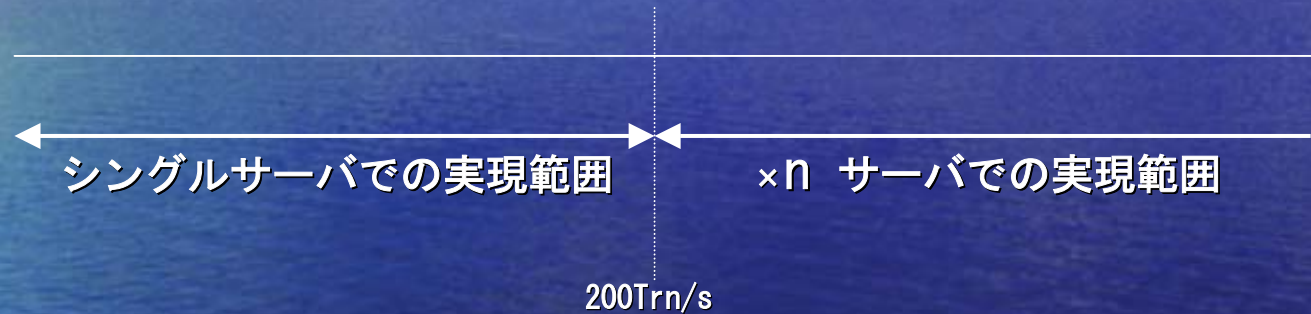
NTT COMWARE

システム方式設計に際して

-性能に対する考慮（システム構成に影響する条件）

●性能要求条件による構成の考慮

[初期状態] OS上のプロセス全体で「約200トランザクション/秒（強）」が
シングルサーバで処理可能な限界



Kernel Attribute Parameterのチューニングにより
シングルサーバでの実現範囲の限界は伸びる
(Parameterには相関関係があるので、注意が必要)

×n サーバでシステムを実現する場合
ハードウェア・ソフトウェア構成に強く依存する



システム方式設計に際して

-性能に関する考慮（チューニング要素）

●Kernel Attribute値 主要項目

●Accounting

RESUME

●Network

MAX_LINX

●Memory Managemnt

freepages

項目名	内 容
項目分類	ファイル管理
定義名	NR_FILE
定義内容	システムで同時にオープンしているファイルの物理上限
初期値	4,096
定義位置(ソース)	include/linux/fs.h
主使用位置(ソース)	fs/file_table.c max_files = NR_FILE;
/proc以下の位置	/proc/sys/fs/file-max
変更の可／不可	可
出力内容	NR_FILE設定値
その他関連情報	/proc/sys/fs/file-nr ファイルで、使用状況の詳細が確認できる
カーネルメッセージ	printk("VFS: file-max limit %d reached\n", max_files);

SHMMAX

AX25_DEF_EWINDOW

appletviewer

SHMMNI

AX25_DEF_CONMODE

Attribute値は相互に密接な関係があるため、変更には注意が必要



NOT COMWARE

システム方式設計に際して

-拡張性に対する考慮

- データ中心アプローチ (DOA)
- 機能ブロック分解能向上方針の明確化 (階層化など)
 - ・ アプリケーションのデータ取り扱い方法およびその機構を概念的にカプセル化することが必要
- デザインのシンプル化
- スケーラビリティに対する機構を有するアプリケーション (またはミドルウェア) の適用



システム方式設計に際して

-運用・保守に対する考慮

- システム運用に必要な体制の検討
- リソースの確定（設備など）
- システム運用ツールの決定
- システム運用における監視対象の決定
- バグ修正パッチ，セキュリティパッチなど、取得ルートの確立
（● パートナー業者の選定）

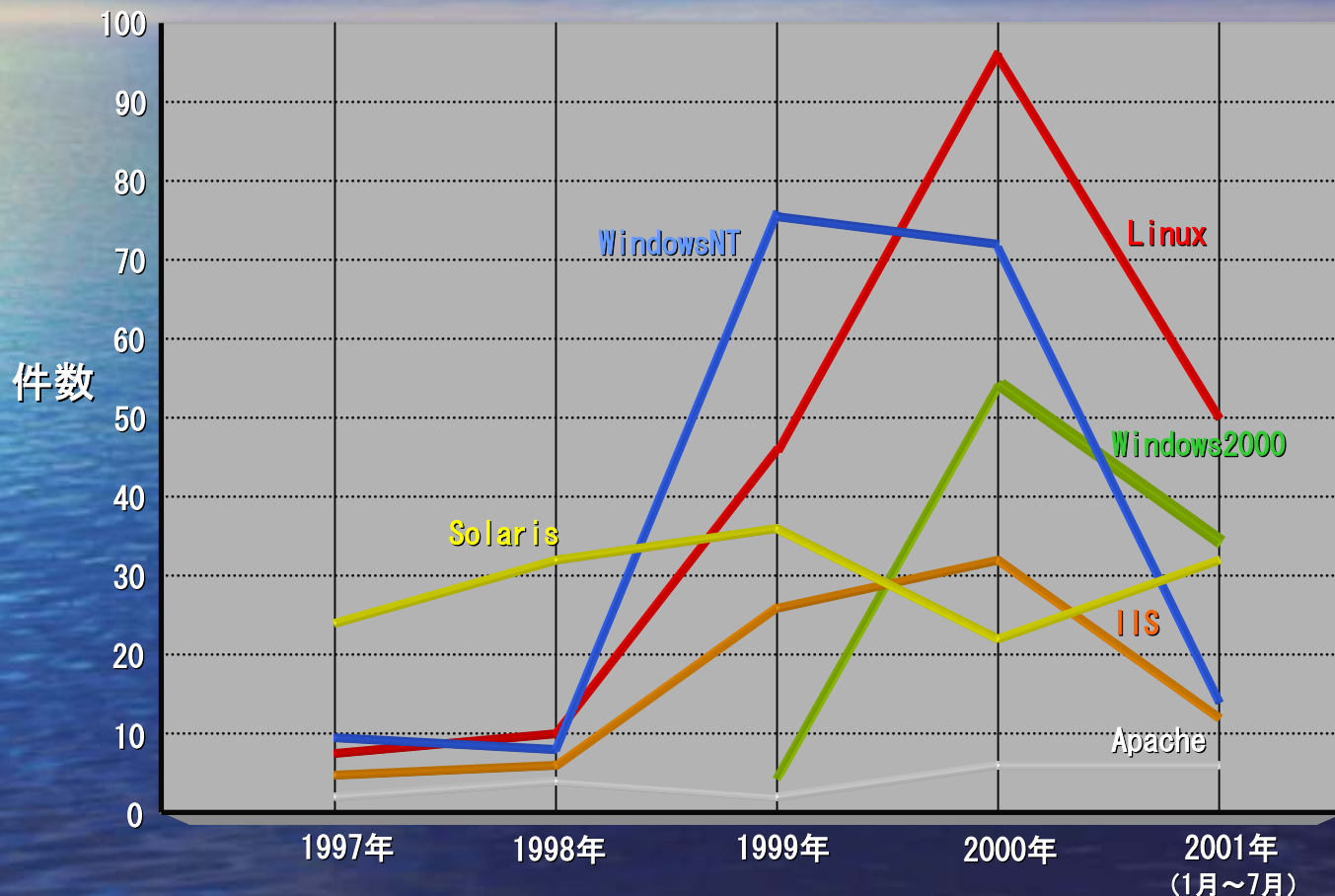


NTT COMWARE

システム方式設計に際して

-セキュリティに対する考慮（セキュリティホール数の推移）

●主要OS, アプリケーションにおけるセキュリティホール数推移



出展：日経コンピュータ（9月号）



NTT COMWARE

システム方式設計に際して

-セキュリティに対する考慮（対象となる事象）

●主要インシデント

インシデント分類		詳細
侵入行為	事前調査	● システム情報調査 ● ポートスキャン ● アカウント名調査
	権限取得	● 一般ユーザ権限取得 ● root権限取得 ● パスワード調査
	不正実行	● ファイル奪取 ● 資源利用 ● 不正プログラム埋め込み ● 中継
	後処理	● アクセスルート作成 ● 証拠隠滅
サービス妨害		● 過負荷の意図的発生 ● セキュリティホールを利用した攻撃
ソーシャルエンジニアリング		● 日常のコミュニケーション手段を利用し、情報を収集



システム方式設計に際して

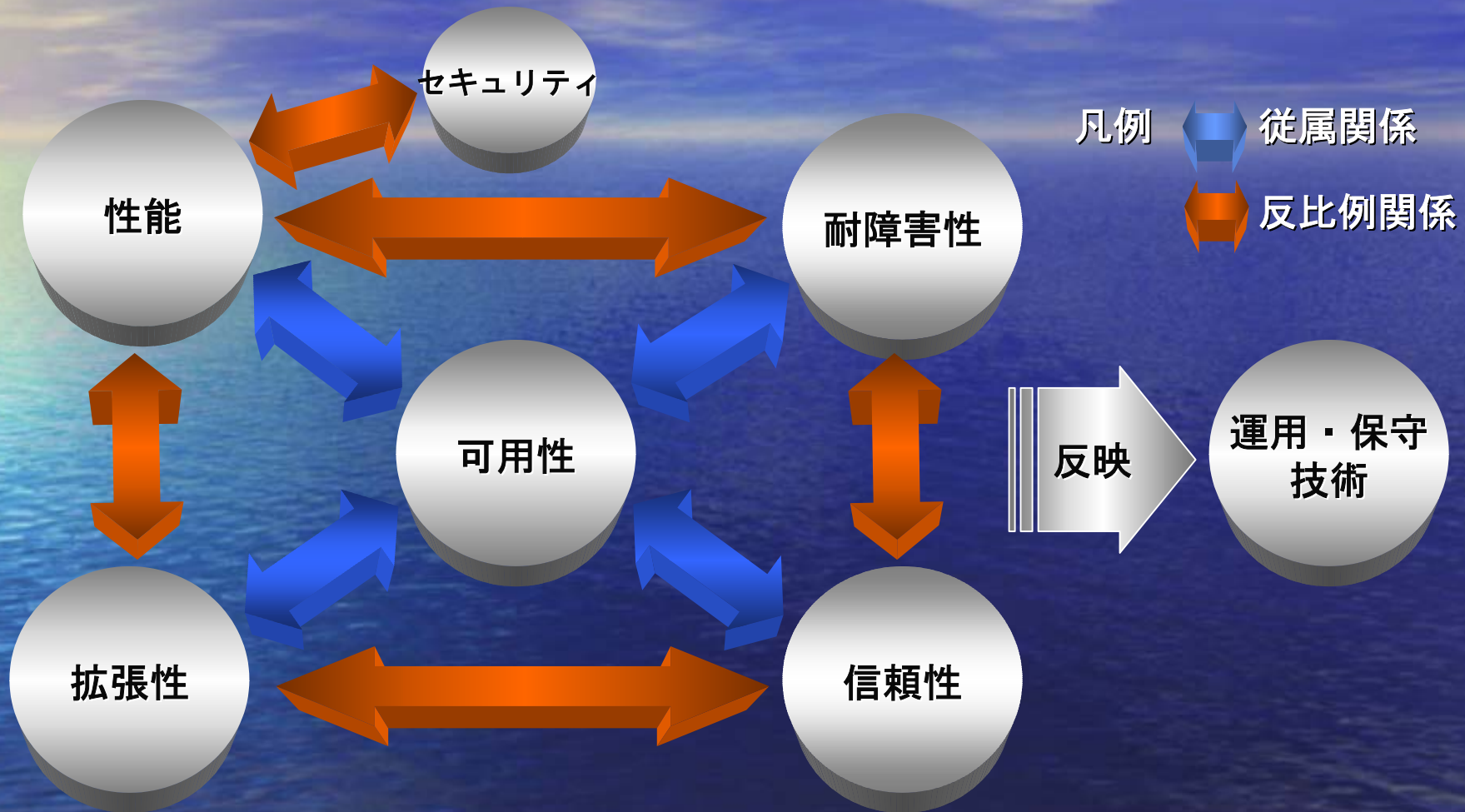
-セキュリティに対する考慮（具体的な方策）

- 不要機能をKernelから排除
- 不要サービス（Daemon），不要ポートを排除
- 不使用サービス，不適切プログラムの排除
- セキュリティホールの排除
- セキュリティ情報の入手ルート確保と周知
- セキュリティ監査手法の確立
- アプリケーション開発の精査（特に言語系：Java, C, C++など）
- 安全なバージョンのソフトウェアを適用
- システム監視（Log, 資源などに対する不正状況自動検知）の強化
- ネットワーク構成全体に対する脆弱性チェック



システム方式設計に際して

—主要特性の相互関係



システムを構築する上でそれぞれの関係が非常に重要



NTT COMWARE

システム方式設計に際して

ーシステム停止の原因と解決策

分 類	詳 細	対 策
ハードウェアに関する障害	●ハードウェアの故障 ●通信設備の障害 ●電源設備などの障害 ●システムの構成変更	●フォールトトレランス ●HA ●データの喪失防止 ●停電対応(UPS) ●ネットワーク経路の多重化 ●オープン製品の適用
ソフトウェアに関する障害	●OSの障害 ●ミドルウェアの障害 ●アプリケーションの障害 ●システムの構成変更	●OSの強化 ●HA ●ソフトウェアの信頼性向上 (例外処理の極小化など) ●リカバリ機能 (ソフトウェア障害からの迅速な復旧)
システムの運用・保守に関する問題 (人的要因)	●オペレーションミス ●メンテナンス作業によるミス ●ハードウェアの拡張および増設 ●ソフトウェアの拡張および追加インストール	●自己診断／故障自動通知機能の充実 ●オンライン中の構成変更・拡張・増設 ●オンライン中のソフトウェア 設定変更・インストール ●ホットスワップ ●オンライン保守機能の充実 ●バックアップ対策
環境に関する障害	●地震など	●遠隔ロケーション間の多重化 ●バックアップ対策



システム方式設計に際して

ーシステムデザインのアプローチ

システムに求められるミッション(可用性の要求レベル)

性能に対する
要件

ハードウェアに
対する要件

ソフトウェアに
対する要件

ネットワークに
対する要件

セキュリティに
対する要件

各項目が相互に影響し合うため
トレードオフによりシステム形態と
トポロジーを勘案しながら
決定していくことが重要

運用に対する要件

保守に対する要件

初期導入コスト

運用コスト



クラスタ適用事例

Co-Agenda

- 高信頼性大規模メールサーバ



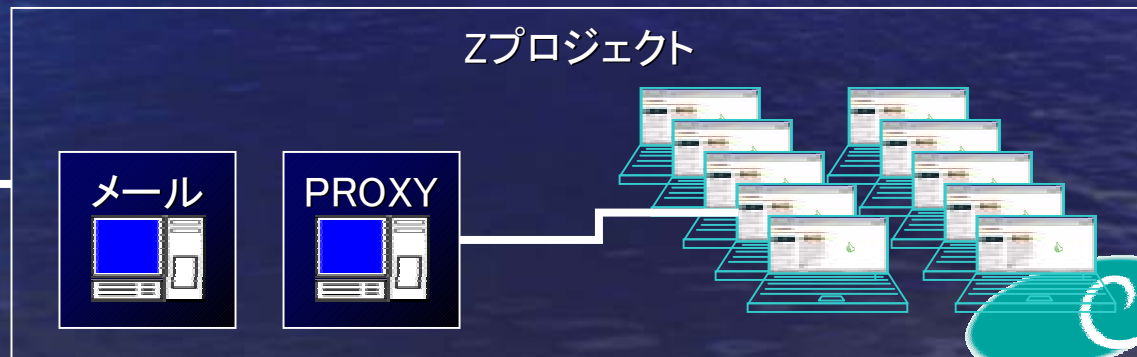
NTT COMWARE

クラスタ適用事例

ー改善前



...

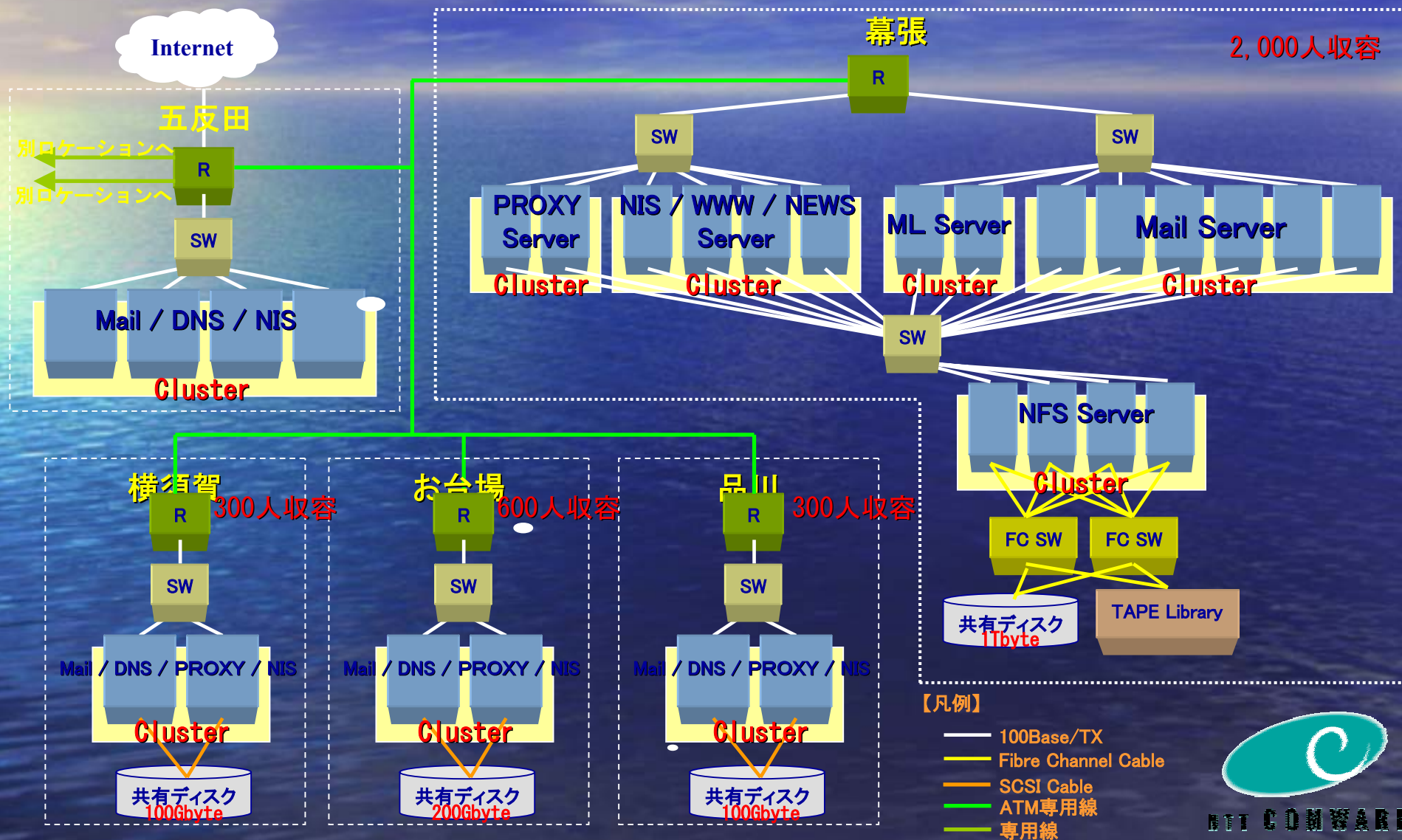


80台のサーバ

3000人のユーザ

クラスタ適用事例

—全体構成



クラスタ適用事例

ソリューション

拡張柔軟性の確保
(需要予測困難)

投資の保護

低コスト

高性能の保持

障害対策



クラスタ化

Linux

SAN etc...



NTT COMWARE

クラスタ適用事例

ソリューション

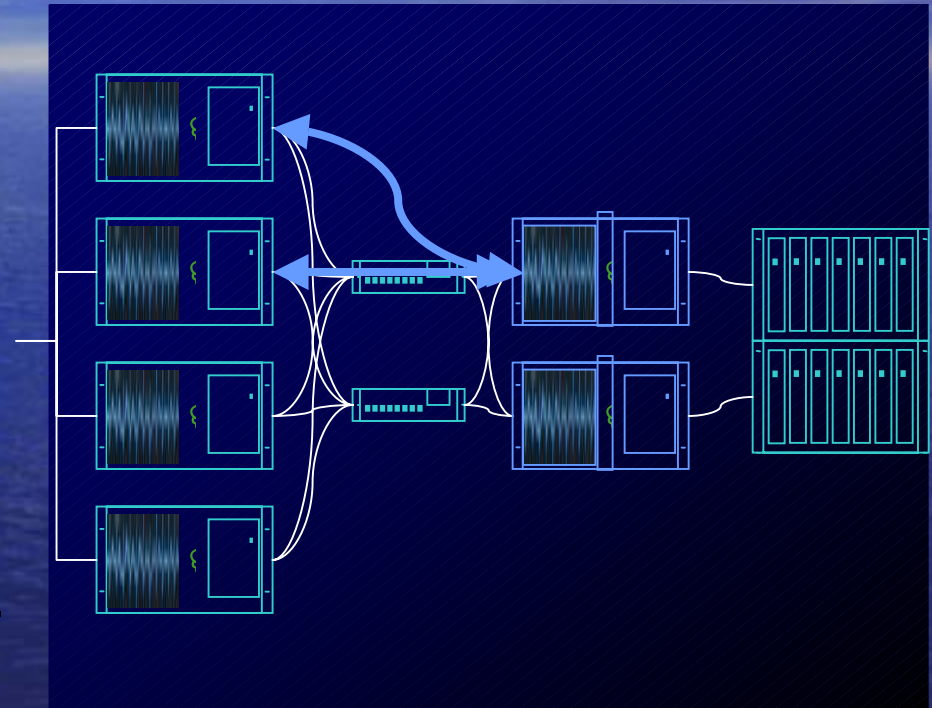
- Progart / Local Cluster Enterpriseによりクラスタ化
 - － 負荷分散(高性能の保持)
 - － 障害対策
- I/O向上のため、Storage Area Network (SAN) を採用
 - － 拡張柔軟性
- Intelligent Cacheの導入
 - － I/O性能向上
- IAサーバ (VA Linux) の導入
 - － 信頼性保持, 低コスト



クラスタ適用事例

—Intelligent Cache

- サーバとディスクの間に位置するサーバ
- RAIDカード, SCSIカードが持つキャッシュメモリだけでは不足する大容量ディスク用のキャッシュメモリを実現
- データのキャッシングやデータ書き込み時のライトバックを行うすることにより、ディスクI/Oの高速化を実現



クラスタ適用事例

ーソフトウェア構成

qmail

bind

squid

Progart / Local Cluster Enterprise

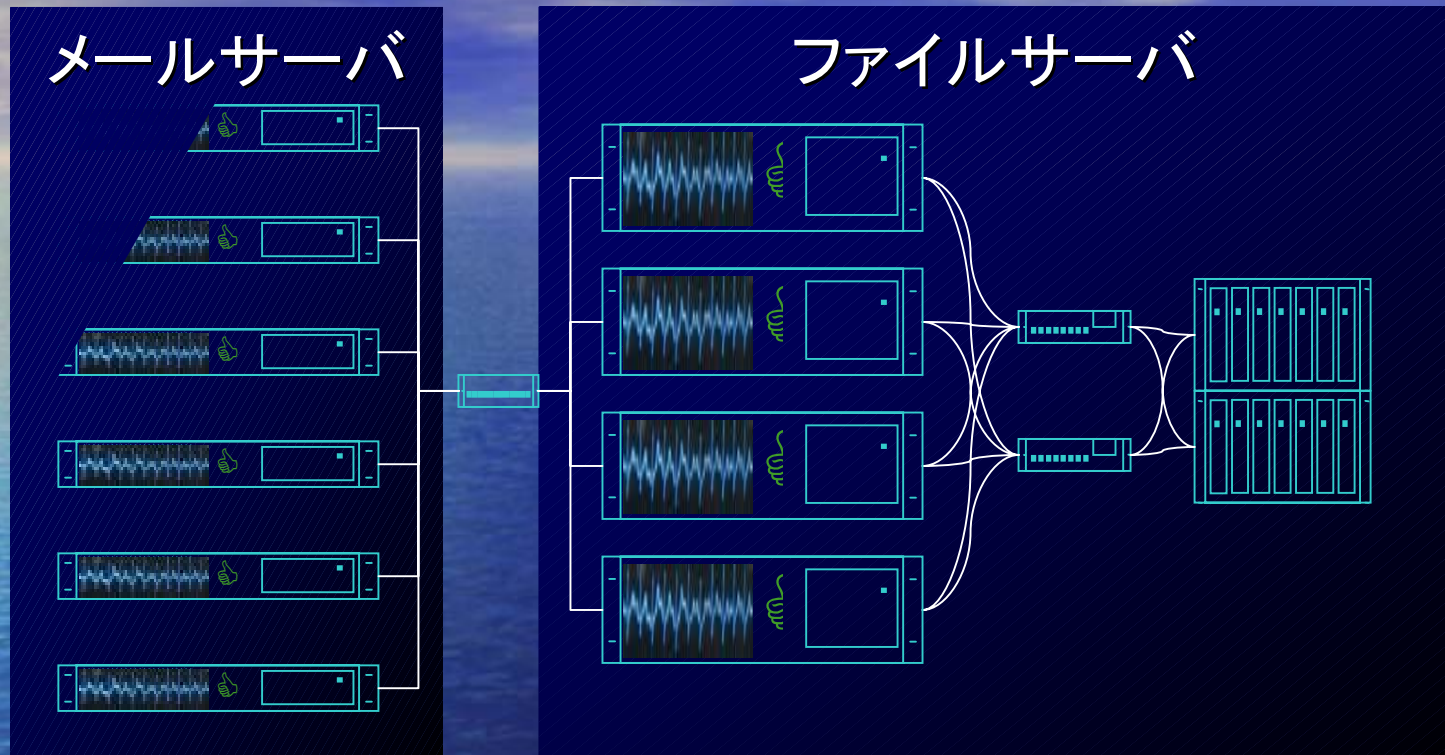
VA Linux



NTT COMWARE

クラスタ適用事例

一幕張 メールサーバ



Progart /
Local ClusterTM Enterprise
Powered by  **PolyServe**

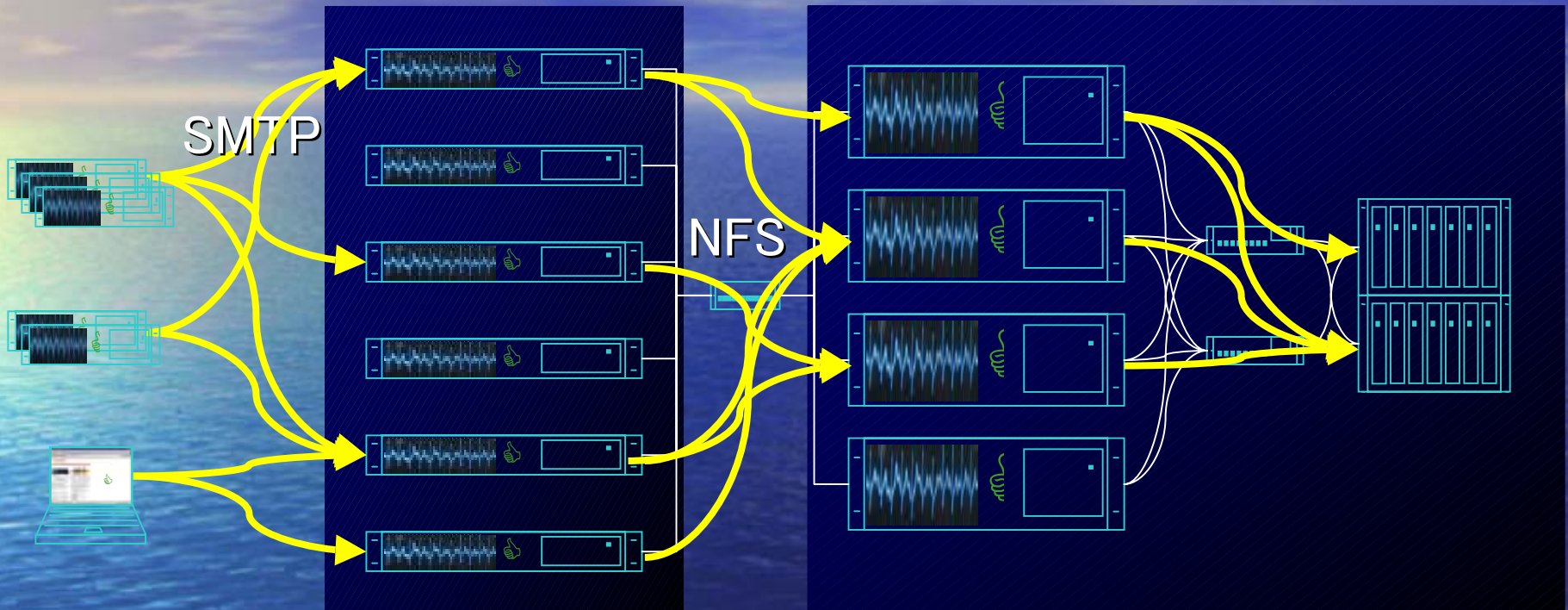
Progart /
Local ClusterTM Enterprise
Powered by  **PolyServe**



NTT COMWARE

クラスタ適用事例

—SMTP



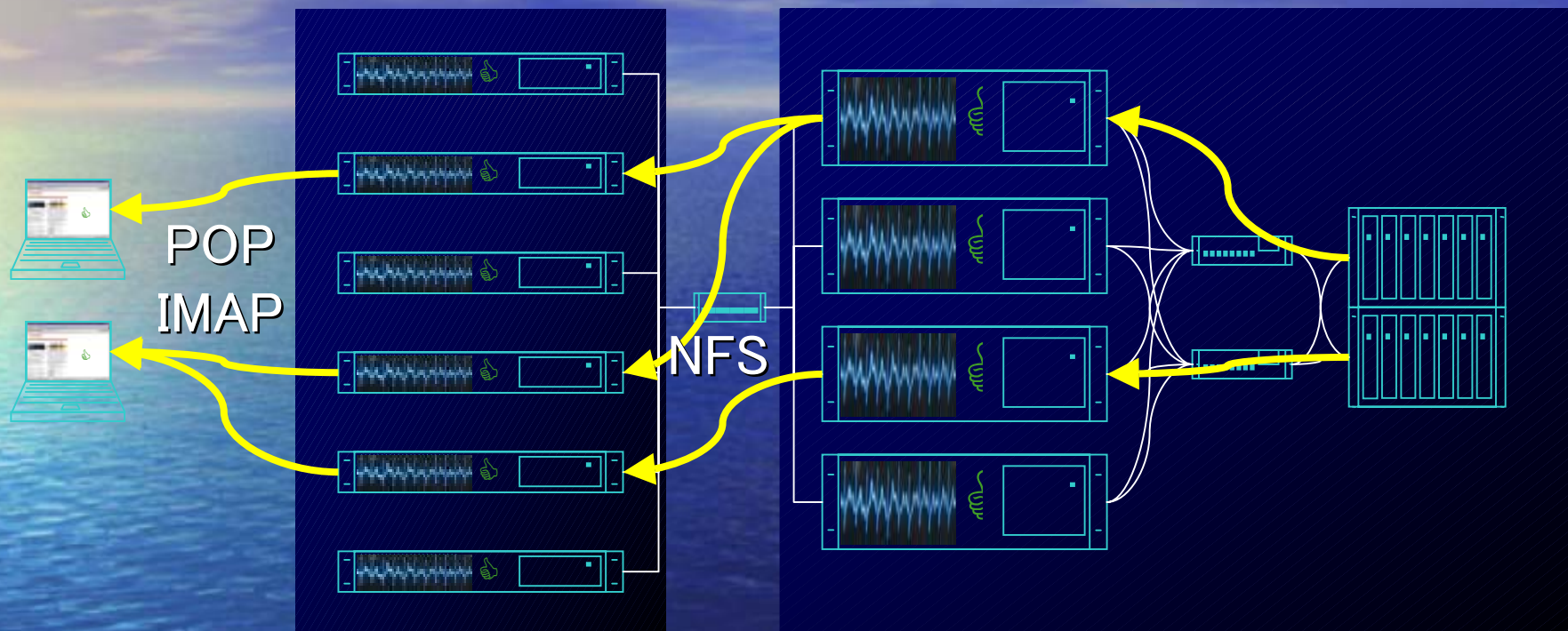
SMTP 7200アクセス/時



MTI CONWARE

クラスタ適用事例

—POP, IMAP



POP 24,000アクセス/時

IMAP 750アクセス/時



NTT COMWARE

ProgartTM / Local ClusterTM Enterprise

Powered by  **POLYSERVE[™]**



NTT COMWARE

概要

- 2台以上のクラスタ構成
- 共有ディスクへ対応
- 高度なフェイルオーバー機能、共有ディスクのサポートにより高い可用性をサーバに付加
- アプリケーションとの密接な連携が可能



動作環境

● Linux

- RedHat Linux 6.x 7.x
- VA Linux
- MIRACLE LINUX

● Solaris

- Solaris 7 (SPARC)
- Solaris 8 (SPARC)

● Windows

- Windows NT 4.0
- Windows 2000

● CPU

- Intel 486相当以上

● メモリ

- 64MB

● ディスク

- 30MB



適用範囲

● Webサーバ

- ー 共有ディスクを用いた構成
- ー データレプリケーションにより各サーバを同期

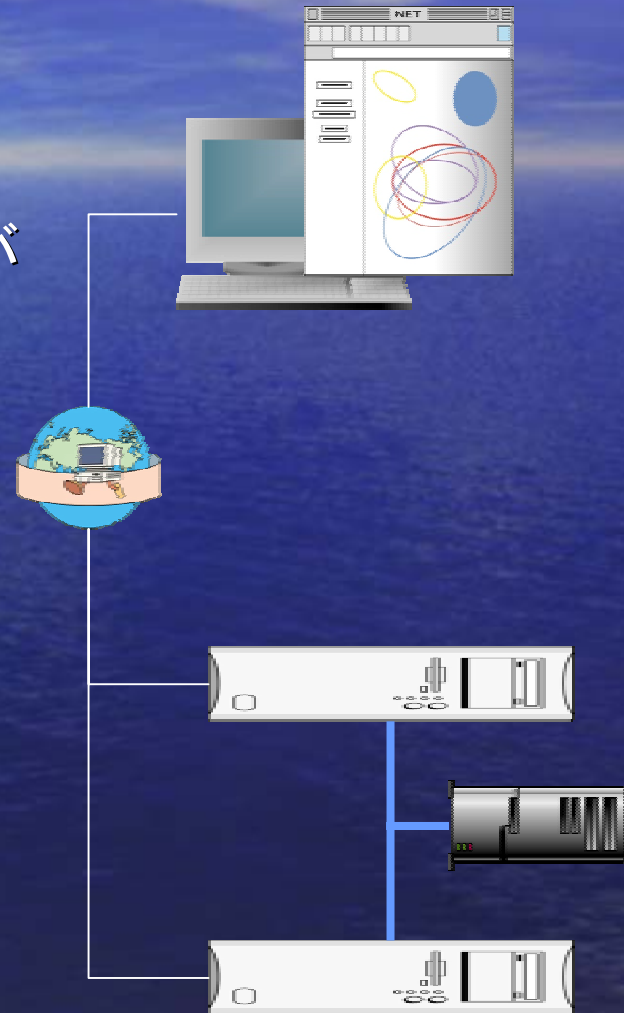
● DBサーバ

- ー 共有ディスクによりデータの一貫性を保証

● メールサーバ

● DNSサーバ

etc. . .

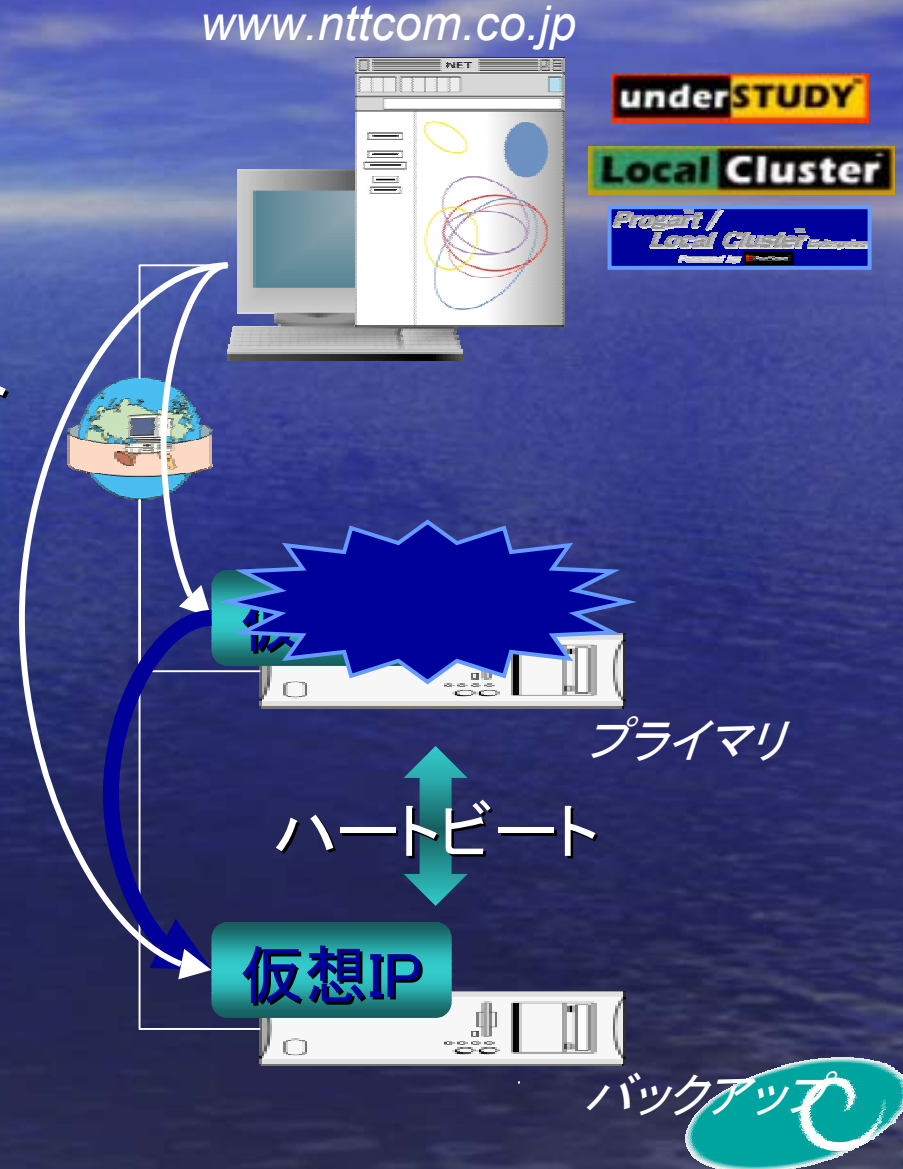


NTT COMWARE

Fail Over

- ソフトウェア、ハードウェアの故障を検出し、正常なサーバへサービスを移行
- 監視対象のサービスポート

HTTP
SMTP
FTP
POP3
IMAP4
NFS
DNS
NIS
任意のTCP/IPポート



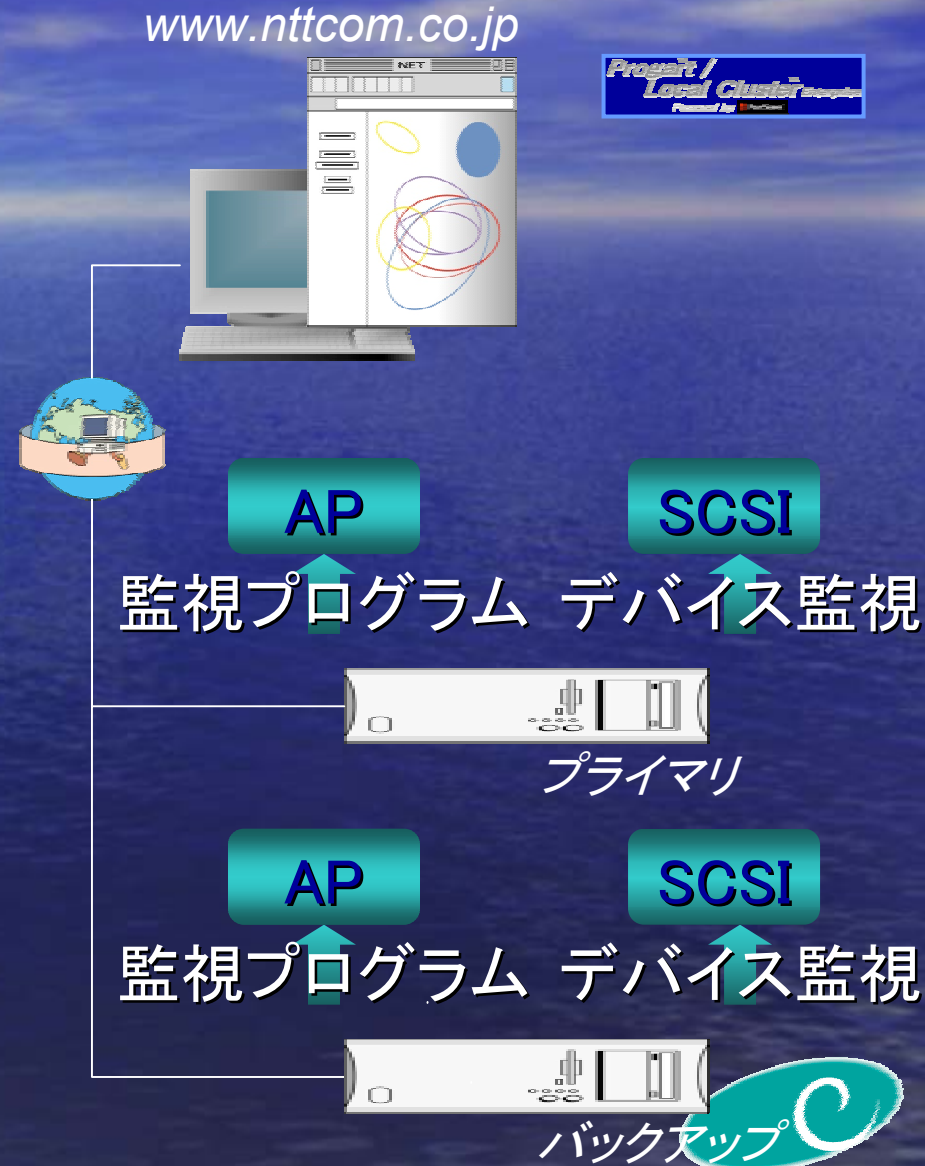
Fail Over 複数ノード

- クラスタを構成するノード数
 - 2～1024台
- サービスごとにフェイルオーバー構成を設定可能
 - 1+1 デュープレックス構成
 - N+1 バックアップ構成
 - N+N マルチバックアップ構成



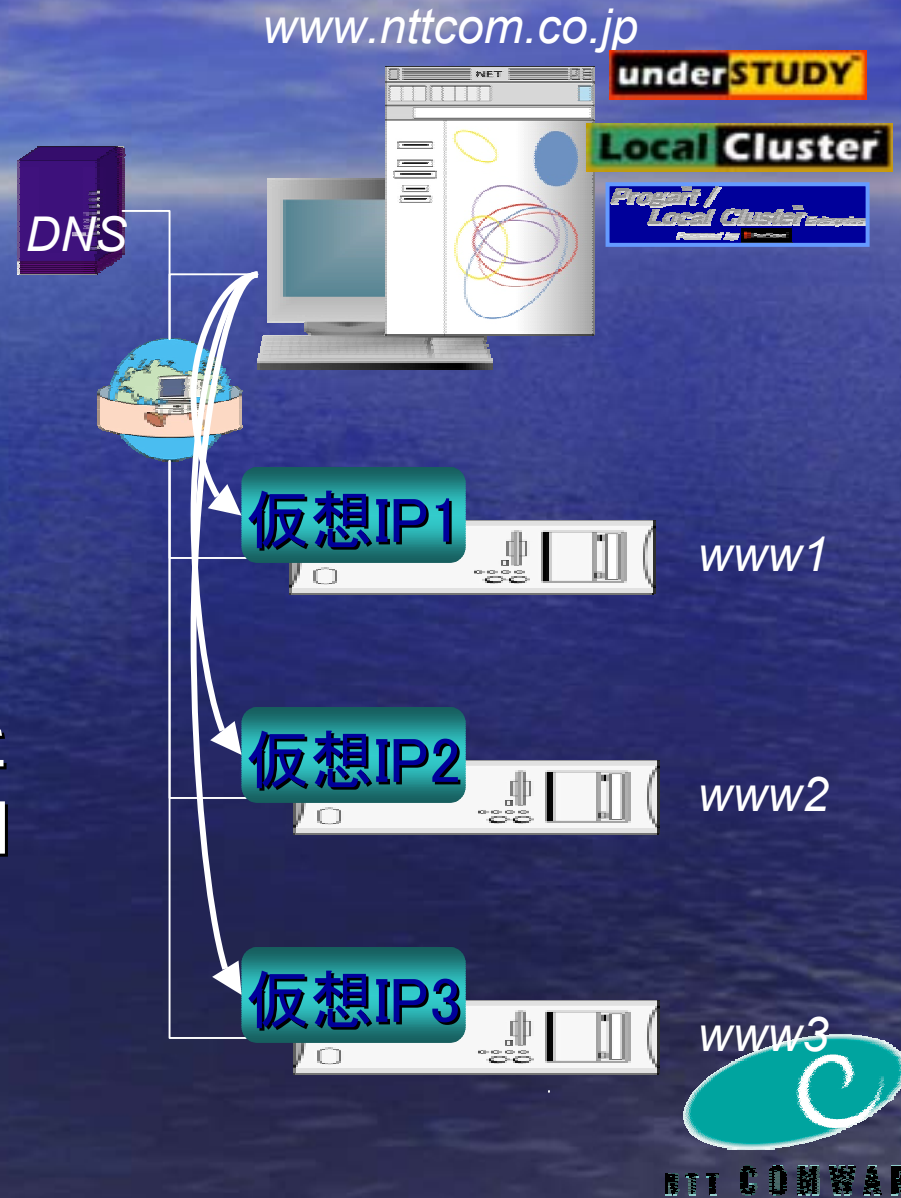
監視対象

- サーバ本体
- 様々なTCPポート
- SCSIデバイス
- 任意のデバイス
- アプリケーション
 - ー 監視対象に合わせて作成する監視プログラムを用いる



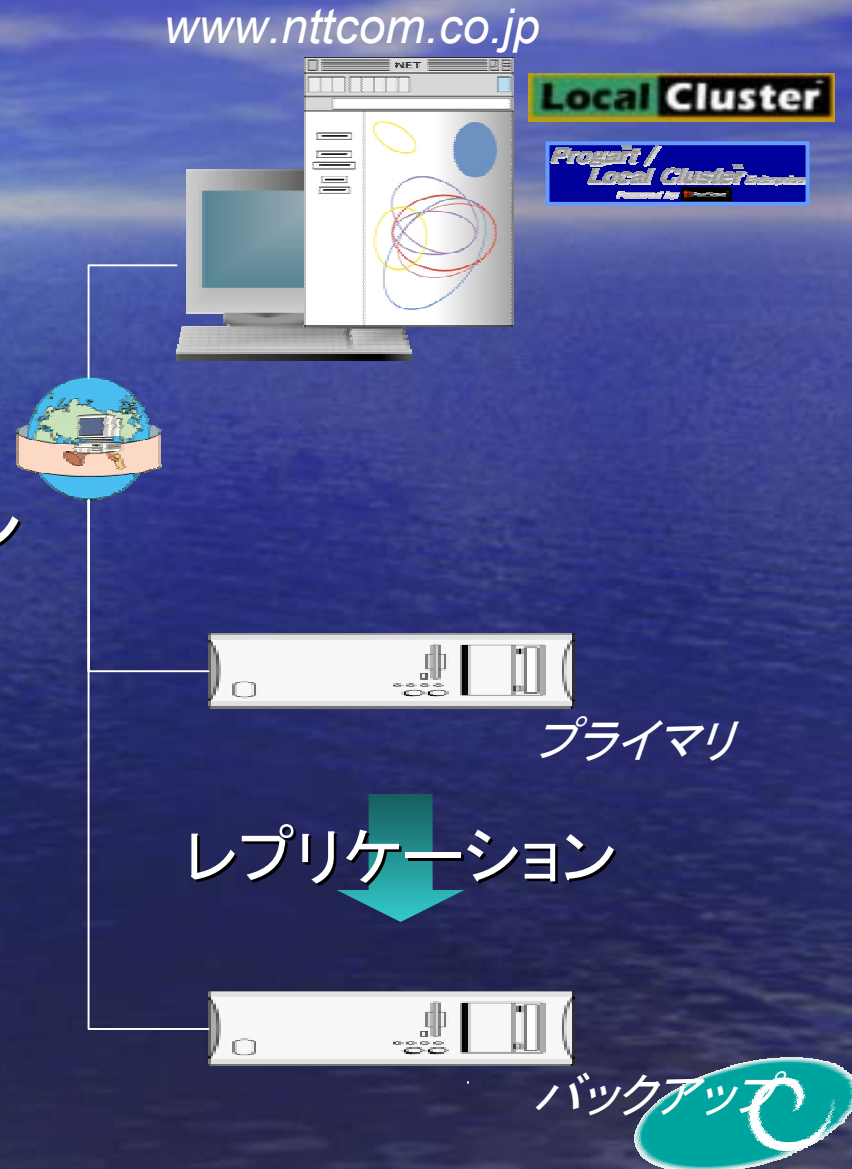
負荷分散

- DNSラウンドロビンを用いて要求を2台のサーバへ振り分け
- フェイルオーバー機能を組み合わせることによりDNSラウンドロビン使用時に問題となる「故障が発生したサーバに要求を送り続ける」事象を解決



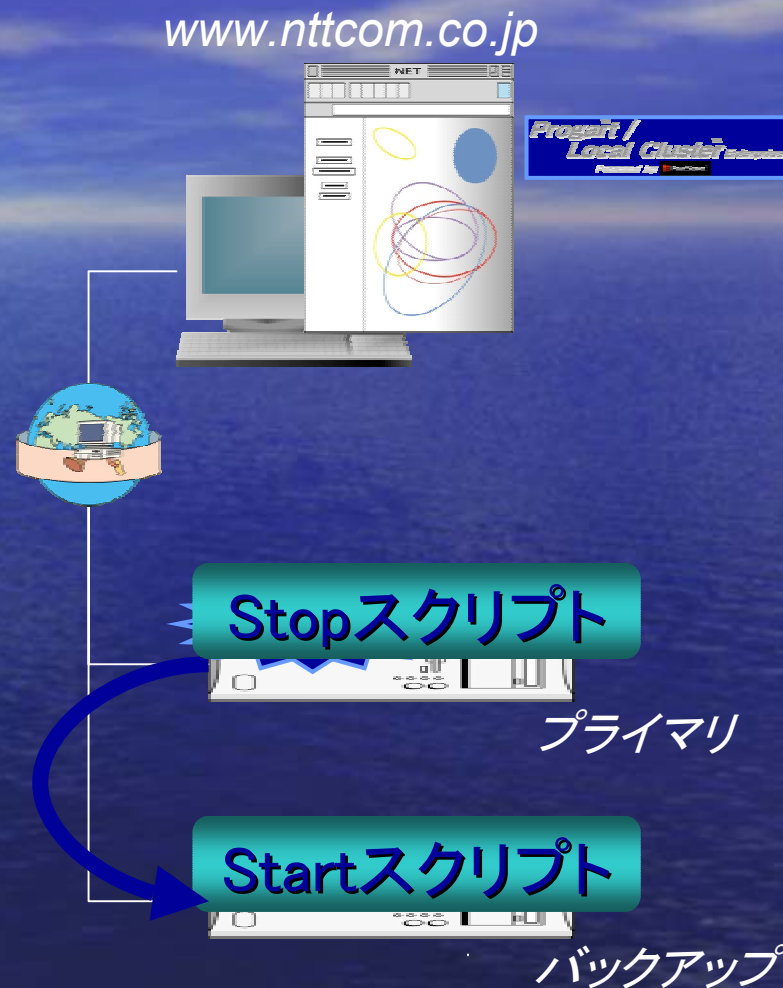
レプリケーション

- 変更されたディレクトリ, ファイルのみをコピーすることによりレプリケーションのトラフィックを抑制
- レプリケーションとアプリケーションの連携
 - アプリケーションを停止した後にレプリケーションを実施
 - レプリケーション完了後、アプリケーションを起動



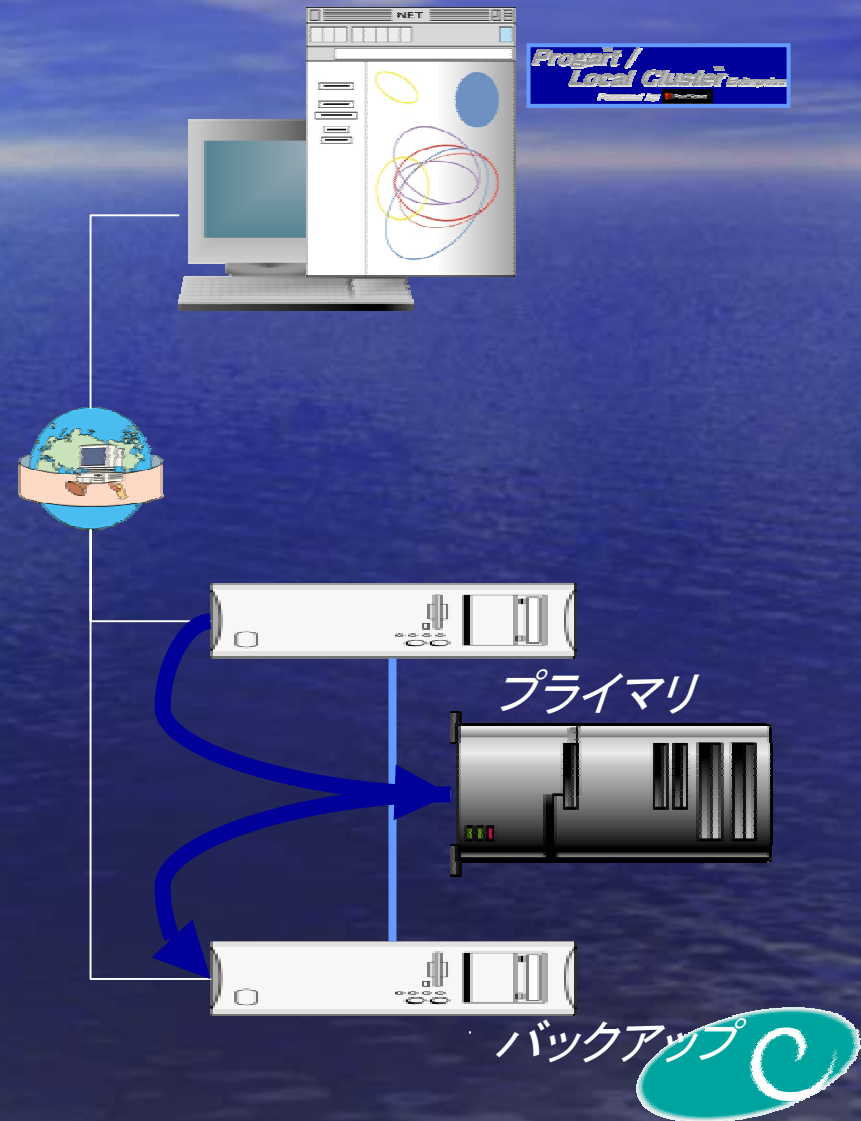
アプリケーションとの連携

- スクリプトインターフェースを提供
 - アプリケーションを監視するスクリプトを設定することによりアプリケーションの障害を検出
 - フェイルオーバー時にアプリケーションの停止、起動処理を実行
- スクリプト
 - Startスクリプト
 - Stopスクリプト
 - recoveryスクリプト



共有ディスク

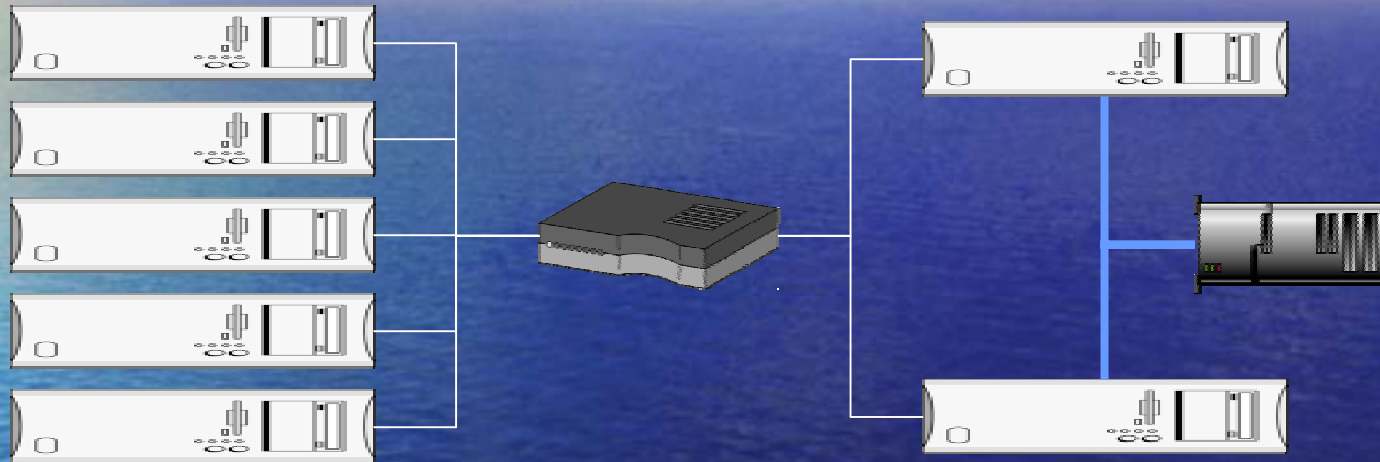
- 共有ディスクを使用することによりデータのー貫性を保つことが可能
- SAN (Storage Area Network) により大規模システムへも対応



クラスタソリューション

Web サーバ
アプリケーションサーバ

データベースサーバ
ファイルサーバ



underSTUDY™

Local Cluster™

*Progart /
Local Cluster™ Enterprise*
Powered by  PolyServe

*Progart /
Local Cluster™ Enterprise*
Powered by  PolyServe



NTT COMWARE

製品に関するお問い合わせ

● お問い合わせ

- <http://www.linux.nttcom.ne.jp>
- Email: progart@linux.nttcom.ne.jp

● 試用版のダウンロード

- <http://www.linux.nttcom.ne.jp>
 - underSTUDY
 - Local Cluster
 - Porgart / Local Cluster Enterprise
 - 試用期間30日



NTT COMWARE

ありがとうございました

NTTコムウェア株式会社 Linuxセンタ

www.linux.nttcom.ne.jp

TEL:(03)5769-7055

progart@linux.nttcom.ne.jp

ooki.kazuhiro@nttcom.co.jp



NTT COMWARE