

マルチベンダー環境の運用事例 - 理化学研究所を例として -

理化学研究所 情報基盤センター
姫野 龍太郎・重谷 隆之・鶴岡 信彦

[キーワード] マルチベンダー、運用、責任体制、切り分け、大規模システム

1. はじめに

理化学研究所(理研)の情報基盤センターの役割は一般の計算機センターと同じで、理研全体にネットワークと共同利用計算機のサービスを提供している。共同利用計算機として、最大のものはスーパーコンピュータのシステムである。今回のマルチベンダー環境の運用事例としては、このスーパーコンピュータ・システムとネットワークシステムの二つの事例を紹介する。

2. スーパーコンピュータ・システム

理研の研究分野の広がりユーザーニーズの変化からスーパーコンピュータは一種類の大きなシステムで構成することは効率的ではない。このことから、大きく分けて三社(富士通、NEC、IBM)の製品を集めたシステム構成になっている。それぞれ、富士通:PCクラスタ、NEC:ベクトル並列機、IBM:HPPSS(アーカイブシステム)である。この他、細かく見るとたくさんの会社の製品で構成されている。

マルチベンダーでの運用の問題は、システム障害時の原因の特定が困難で、対応に関する責任の所在があいまいになるところにある。この対策として、プライムを設定して、システム運用の全体を把握、対応体制(特に障害時)の明確化、責任の所在の明確化を依頼している。しかし、プライムだけでは解決できない問題もあり、定期的なミーティングを開催している。このミーティングでは、全ベンダーが出席し、積み残した障害・対応状況の確認、問題の共有化を図っている。担当部分だけでは見えない部分を共有することで、システム運用全体の中で可能な問題解決手段を選択可能となる。理研では月に1回(約2、3時間)開催している。

利用者にはシステム毎にコマンドオプションが違ったり、実行モジュールが異なることによる不便をなくすために、システム側で吸収している。

3. ネットワーク

ネットワーク機器はベンダーによりそれぞれ独自の仕様で通信している部分があるので、混成するとトラブルが多発した。現在、理研では6つの支所それぞれは単一ベンダーの製品に統一したが、理研全体としてはマルチベンダーとなっている。同一支所内で複数のインテグレータと契約したことがあり、これは苦勞が多い。基本的な問題は契約範囲外への無関心が原因で、インテグレータ間の連携が必要となる。これについての誠意を期待してはいけない。独自の監視網、監視機構を持つ必要がある。

4. まとめ

マルチベンダー環境は機器最適化のため、必然であり、避けて通れない。このため、全体として責任を持つプライム設定が必要となるが、これだけでは万全ではない。解決への熱意を関係者全員が持つことが最も重要である。競合関係にある場合、議論が困難なこともあるが、ユーザーサービスを優先し、我々の場合、問題解決のために特殊仕様となる開発も依頼している。



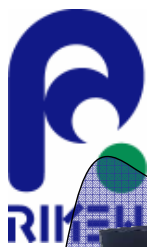
マルチベンダー事例

1. スーパー・コンピュータ・システム
2. ネットワーク・システム

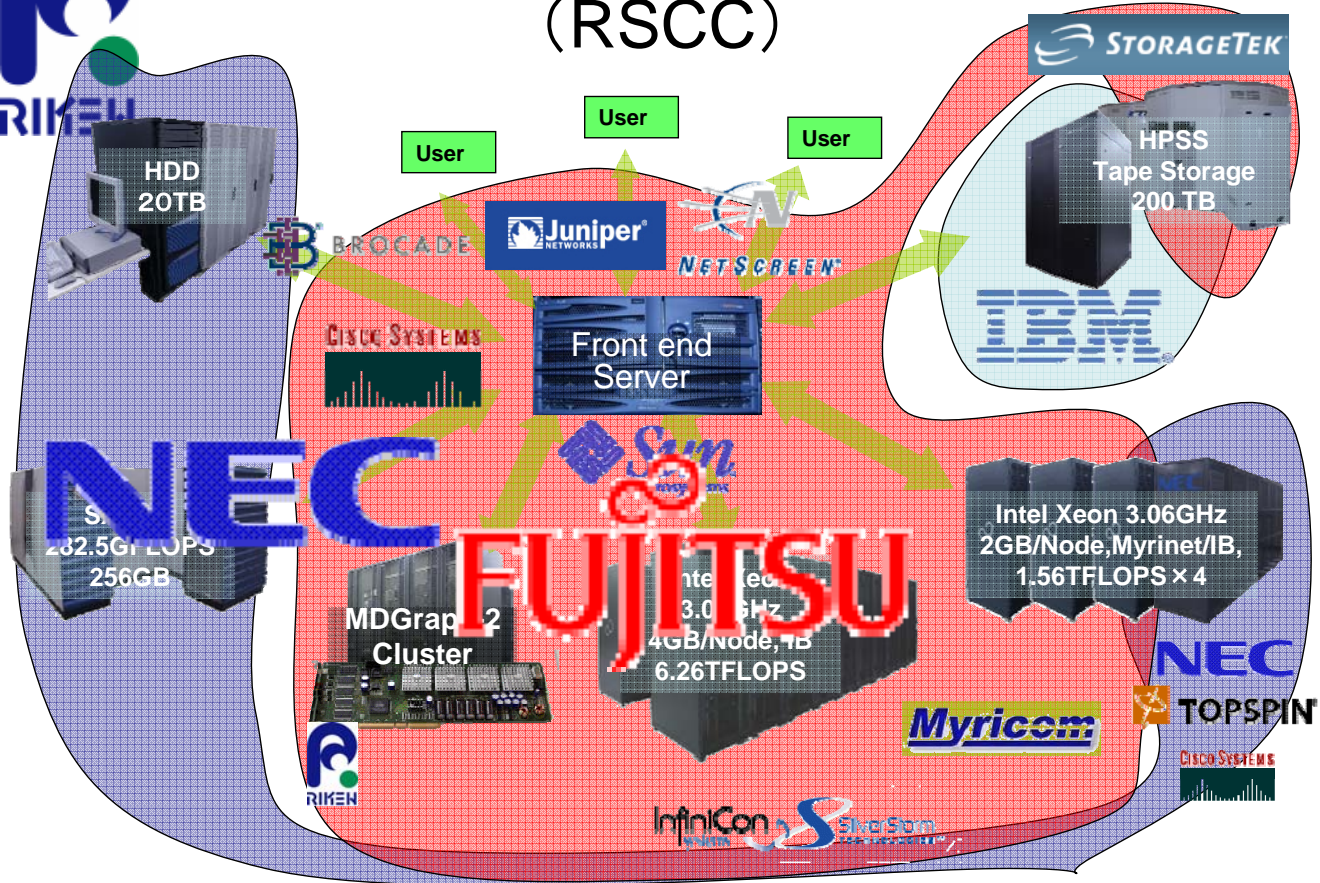


マルチベンダー環境事例1

スーパー・コンピュータ・システム
重谷



1.1 RIKEN Super Combined Cluster (RSCC)



1.2 マルチベンダーの必要性

- 一社でカバーできない
 - ベクトル機、PCクラスタ、HPSS
- ベクトル機の必要性
 - 過去の資産 (Fujitsu VPP700E/160) の継承
 - ベクトル化されたプログラム
 - 並列化できないプログラム
 - 1CPUで大容量のメモリが利用できる環境が必要
- 新規の研究分野 & ユーザーの要求
 - スカラー向きソフトの増加
 - フリーソフトの利用希望の増加
- 既存システムとの互換性
 - BNLでの実験データ解析ではHPSSを利用
 - 共用できるのであれば2つ以上のアーカイブシステムは無駄



1.3 マルチベンダー構成による RIKEN Super Combined Cluster

- 富士通
 - フロントエンド計算機
 - [SUN Microsystems](#)
 - 管理用ネットワーク
 - [Cisco systems](#)、[Netscreen Technologies](#)(現[Juniper Networks](#))、[Fujitsu](#)
 - Linuxクラスタ(512ノード、128ノード×3)
 - インターコネクト
 - [Myricom](#)、[InfiniCon System](#)(現[SilverStorm Technologies](#))、
 - テープ装置
 - [StorageTek](#)(現[SUN Microsystems](#))
- NEC(SX-7)
 - 大容量メモリ計算機
 - Linuxクラスタ(128ノード×1)
 - インターコネクト
 - [TopSpin Communications](#)(現[Cisco systems](#))
 - ディスクストレージ
 - [Brocade](#)
- IBM
 - テープアーカイブサーバ(HPSS)
 - MD計算加速ボード(RIKEN(manufactured by IBM))



1.4 マルチベンダーでの問題と対策(1)

- システム障害時における原因の特定が困難
 - 責任の所在が不透明
 - ⇒ プライムを設置
 - システム運用の全体を把握
 - 対応体制(特に障害時)の明確化
 - 責任の所在の明確化
 - プライムだけでは解決できない問題も
 - ⇒ 定期的なミーティングを開催
 - 積み残した障害・対応状況の確認
 - 問題の共有
 - 全ベンダーが参加することが重要
 - 担当部分だけでは見えない部分を共有
 - システム運用全体の中で可能な手段を選択可能
 - 理研では月に1回(約2、3時間)
 - 定期的ミーティングだけでは、時間的な制約もある



1.4 マルチベンダでの問題と対策(2)

具体的な障害対応では他社には知られたくない情報と密接な関係がある場合も

⇒ プライムと担当ベンダー3者、もしくは担当ベンダーとの個別ミーティングも重要

- コマンド・オプションが異なる
- 同じPCクラスタでもインターコネクトが違くと実行モジュールが異なる
 - 利用者にマルチベンダーであることを意識させたくない

⇒ システム側で吸収



1.5 具体的な問題と解決事例(1)

◎ フロント・エンドからテープ・アーカイブシステムへのファイル転送の速度低下

関係するシステム【ベンダー】:

- フロント・エンド: Solarisサーバ【富士通】
- ファイル・システム: GFSによるSAN【NEC】
- アーカイブ・システム: AIXサーバ+HPSS【IBM】

⇒ 3社が協力して原因を究明

– GFSクライアントを構築したSolarisの仕様と判明

⇒ GFSクライアントの修正により解決



1.5 具体的な問題と解決事例(2)

◎ クラスタにおける実行モジュールの共通化

- Linux クラスタは5つに分割して運用
 - インターコネクトは3種類: Mirinet、InfiniBand × 2
- ⇒ 3種類のインターコネクト全てに対応する実行モジュールを作成
- ⇒ 利用者は1度実行モジュールを作成すれば、どのクラスタでも同じモジュールを利用可能

◎ コンパイラ、バッチ・ジョブ投入などコマンドがクラスタとSX-7で異なる

- ⇒ wrapperにより隠蔽し、利用者は同じコマンドが利用可能



マルチベンダー環境事例2

理研ネットワークの場合



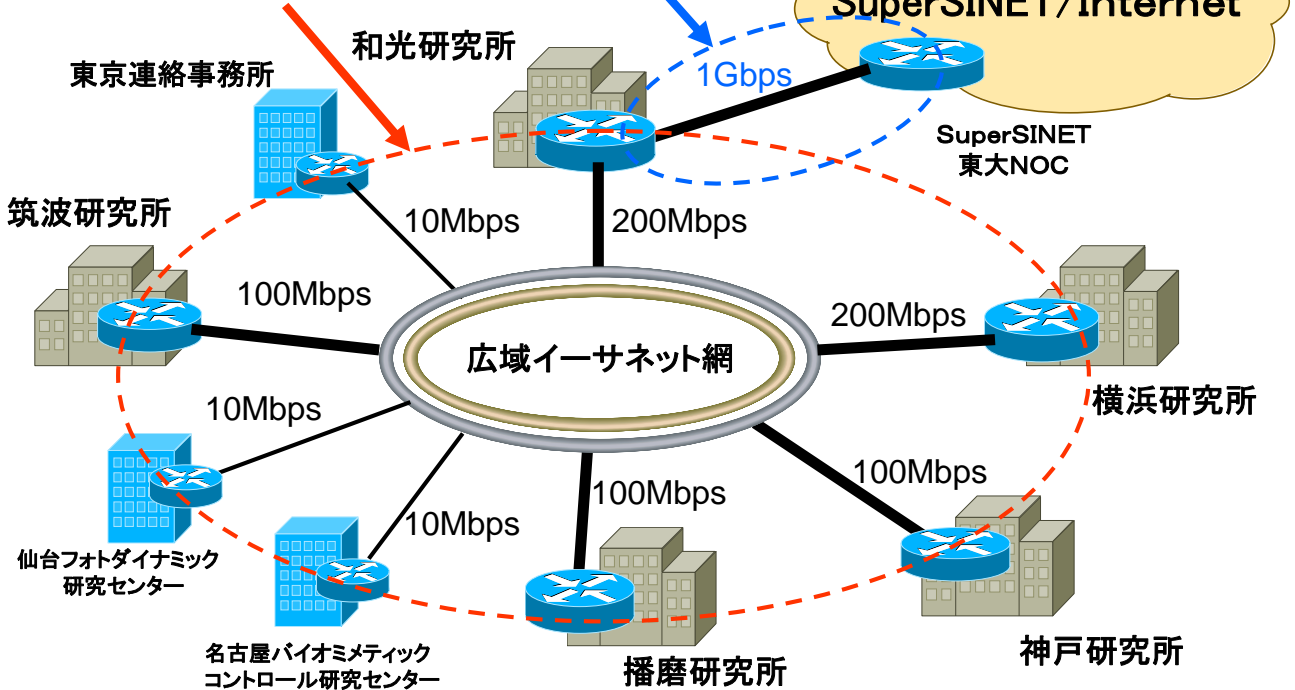
2.1 理研ネット関連ベンダー



SuperSINET回線及び全理研ネットワーク

全理研ネットワーク

SuperSINET回線



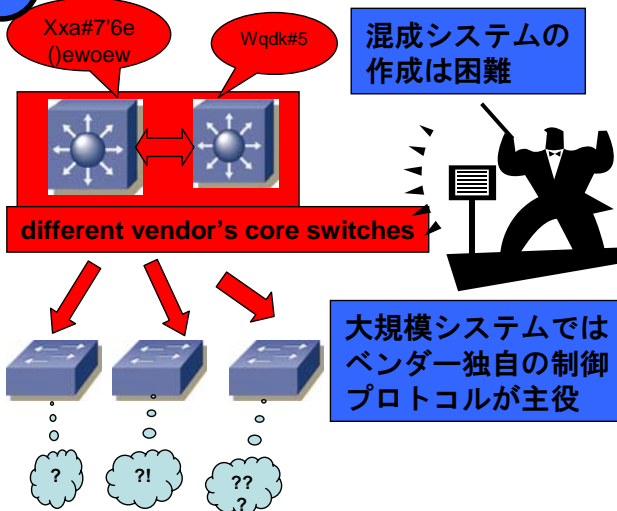


2.2 マルチベンダー機器 混ぜると危険！



2.3 同種のシステムは混成困難

1



2

エンジニアの知識は一人
ほぼ一社製品程度

A社 switch

B社 router



I know only A's



I know B's, but am not sure about A's



Nobody knows about if A+B works!!

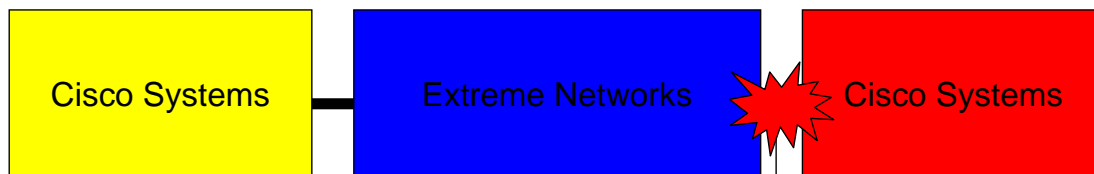


2.3 同種のシステムは混成困難

3

障害が発生する（実例）

物理的に同一ベンダースイッチが隣にしていると判断



隣のCiscoを探すためのCisco
コントロールパケットがExtremeを通過

Extremeには適用できない制御を開始、ネットワークダウン



2.4 我々の取った方法

- 各支所内は単一メーカー
- 理研ネットワーク全体はマルチベンダー
- 異種の機器は混在OK

支所ごとにLANメーカーを統一



各支所とも異種機器については混成



和光本所



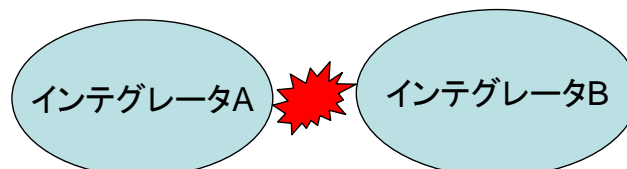
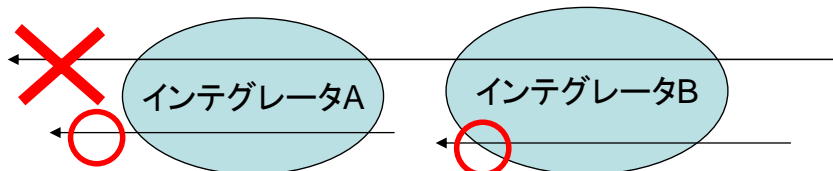
2.5マルチインテグレータ (インテグレータ複数と契約してしまった！)

- 百害あって一利なし
- 基本的な問題は契約範囲外への無関心
- インテグレータ連携について誠意を期待してはいけない



実例1 障害対応

- ① 両インテグレータを通過しないトラフィックについては障害の発見すらできない
- ② 発見しても、相手の故障と主張し相互に責任回避する
- ③ 境界に近い障害報告が行われず・二重に行われる(夜中に2回起こされる)





実例2 問題解決能力

- インテグレータ営業同士の話し合いに異常に時間がかかる(契約期間終了待ち?)
- 技術者が対話しない(させない)ので解決が遅い
- 結果として理研が予算供出、調整、解決方法提案等を行うことに



我々のとっている方法

- 契約課を説得し、長期的な視点でマルチインテグレータ体制を極力なくす
- インテグレータの監視範囲、対応範囲を極力明確にする
- 独自の監視網、監視機構を持つ



まとめ

1. マルチベンダは機器最適化のため、必然
2. プライム設定は当然だが万全ではない
 - 問題意識のずれ(それは仕様です！)
 - 問題の共有化 (ユーザーが困っていることが問題)
 - 解決への熱意を関係者が全て持つことが重要
 - 競合関係にある場合、議論が困難なことも
3. 最悪の場合、ユーザーサービスを優先
 - 解決のために特殊仕様となる開発も依頼