複数サイトにまたがる仮想クラスタの構築手法

産業技術総合研究所　情報技術研究部門
広渕 崇宏　横井 威　江原 忠志　谷村 勇輔
小川 宏高　中田 秀基　田中 良夫　関口 智嗣

背景(1)

- 仮想化技術
  - 物理的な計算機資源を論理的に分割・共有
    - 仮想マシン、SAN、VLAN
    - 柔軟な運用による管理コストの低減
  - Cloud Computing
    - 計算資源を必要なときだけ外部から確保
    - 仮想化データセンタ、Amazon EC2 (1VM = $0.1/hour)
- 仮想クラスタ
  - 仮想ノード群からなる大規模なアプリケーション実行環境
  - ユーザ独自の大規模な実行環境を迅速に構築
    - 物理ハードウェア管理から解放
  - 物理資源運用の効率化
    - 遊休資源の低減

背景(2)

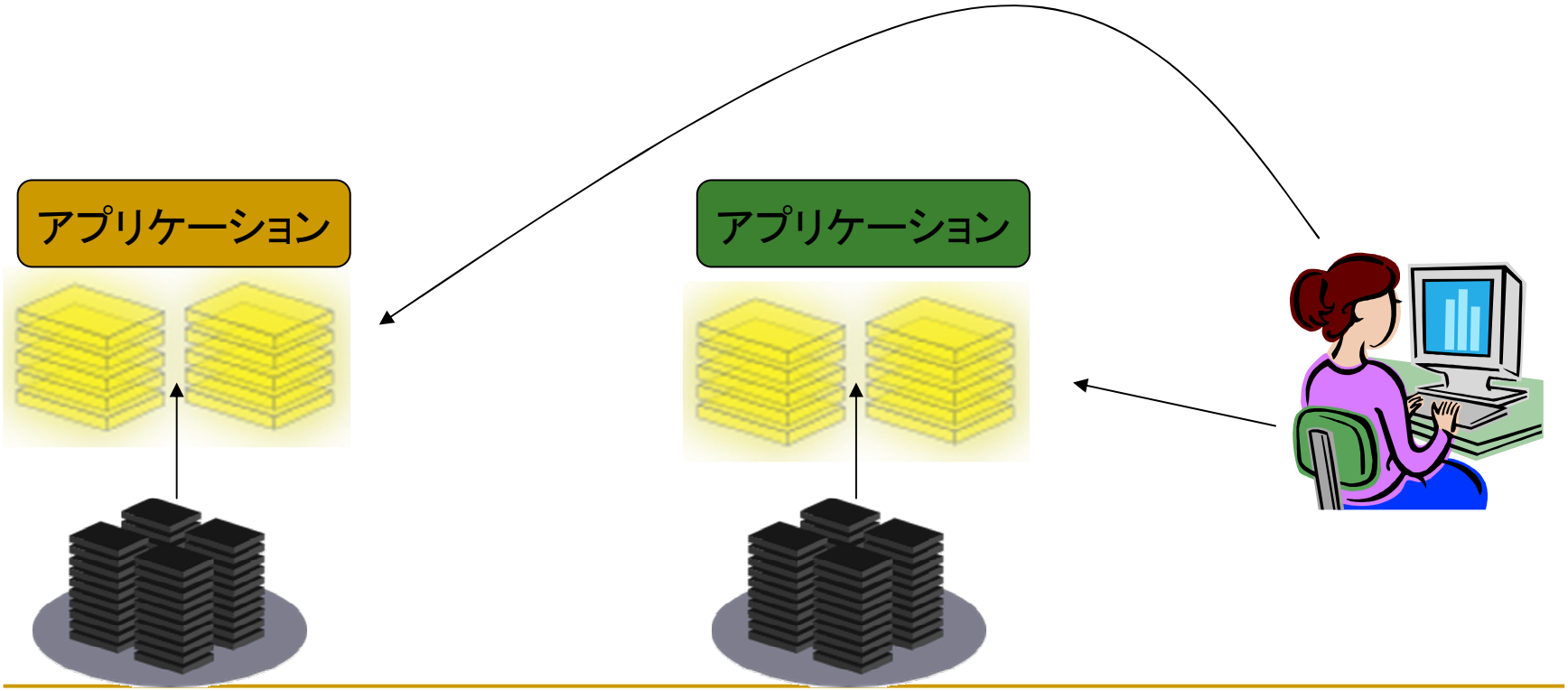
- 仮想クラスタ管理システムの開発
  - 物理クラスタの仮想化キット
  - 簡単導入、低コスト
  - 誰でも独自に仮想クラスタを構築可能に
- 仮想クラスタの大規模化における問題点
  - 単一拠点に存在する物理資源量の限界
    - スケーラビリティ、柔軟性の限界
    - => **多拠点化**
  - 分散仮想資源の管理コスト増大
    - 仮想ノードの分散拠点配置
    - システムソフトウェアレベルからの管理コスト X 分散拠点

目的と成果

- 分散する仮想化資源をもとにした大規模アプリケーション実行環境の実現・管理・運用手法として、マルチサイト仮想クラスタを提案
  - 複数拠点にまたがって単一実行環境の仮想クラスタを構築
    - 単一拠点の計算資源量に束縛されずに大規模な仮想クラスタを構築可能
    - 管理コストが拠点数やノード数の増加に伴って増大せず、運用が極めて容易
    - 迅速な構築と容易なカスタマイズ
- 成果
  - マルチサイト仮想クラスタシステムの設計と実装
  - WAN環境に対する提案手法の妥当性検証
    - 管理にともなうWAN経由トラフィックの最小化
    - 物理クラスタと同等のノードセットアップ時間

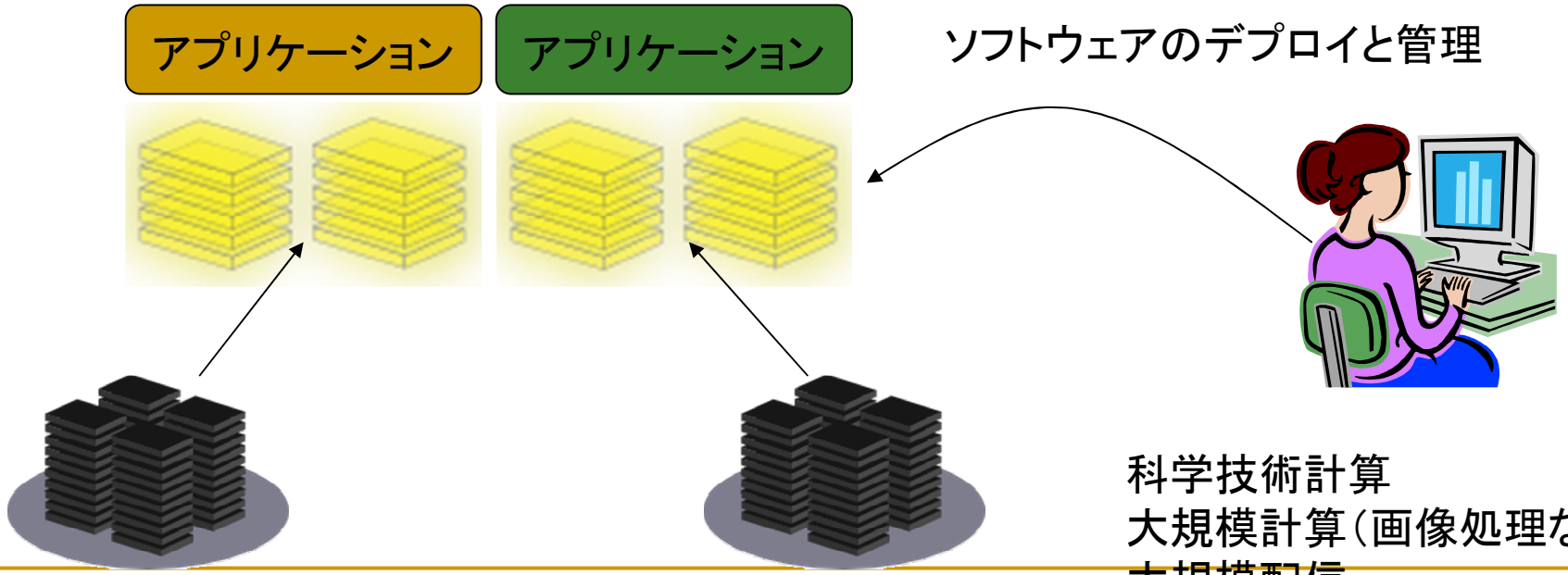
管理形態の比較
（拠点ごとの仮想化）

分散拠点に対するソフトウェアのデプロイと管理はとても面倒
ソフトウェアアップデート、ノードカスタマイズ



管理形態の比較
（マルチサイト仮想クラスタ）

多拠点へのソフトウェアデプロイと管理を、
単一ビューの仮想クラスタを通して行える
=> 多拠点からなる大規模仮想クラスタを容易に構築・運用可能に



科学技術計算
大規模計算（画像処理など）
大規模配信

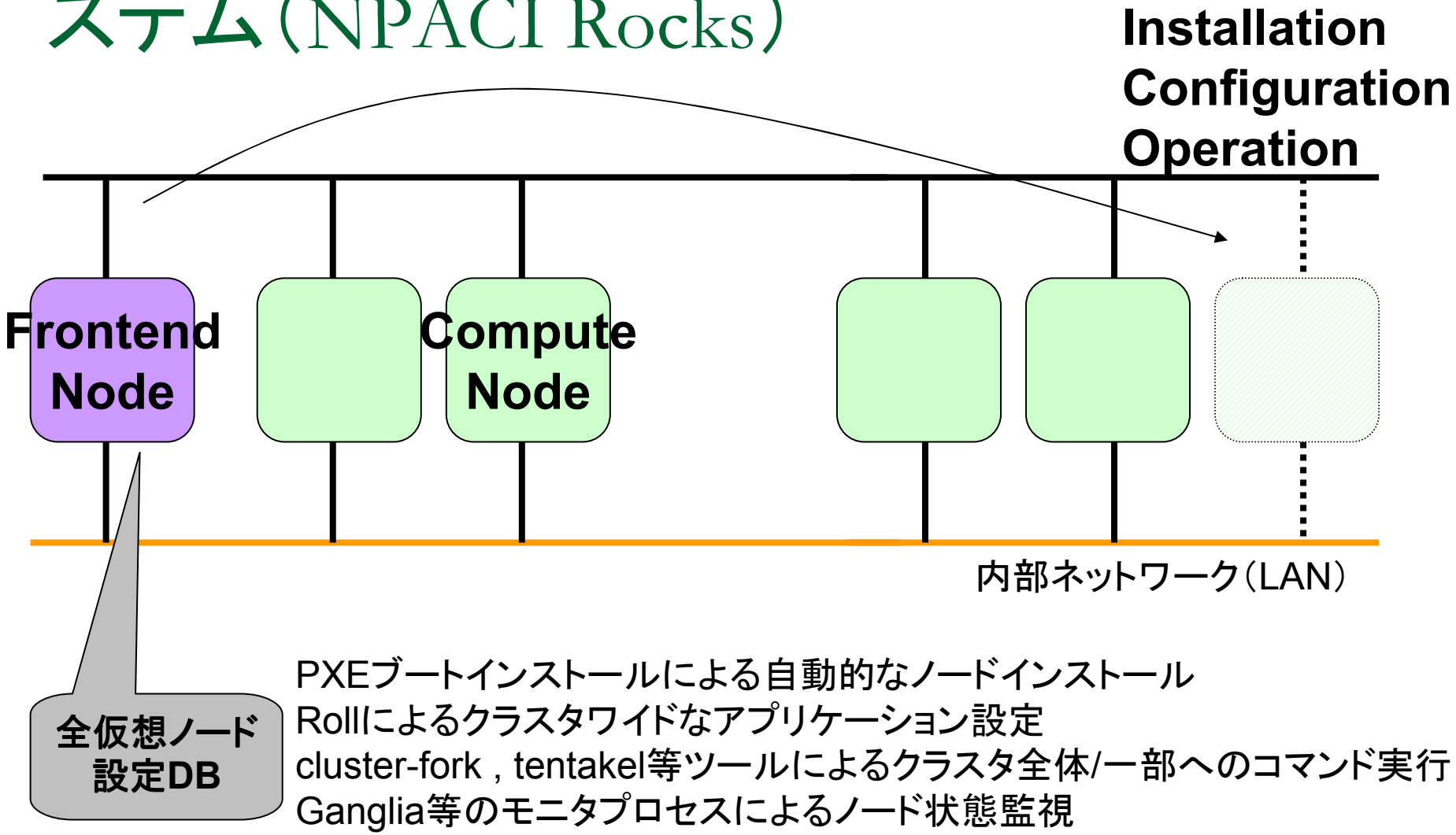
要求事項

- マルチサイト仮想クラスタ管理システムへの要求
  - 大規模ノードに対する強力な管理機能
    - ロバスト（大規模、ノード構成変化）
    - 仮想ノードごとのきめ細かいカスタマイズ
    - 多拠点環境のヘテロジニアスティに対応
  - 管理者にとって直感的で使いやすい分散ノード管理
  - 迅速な設定、管理にともなうトラフィックの最小化

実現手法

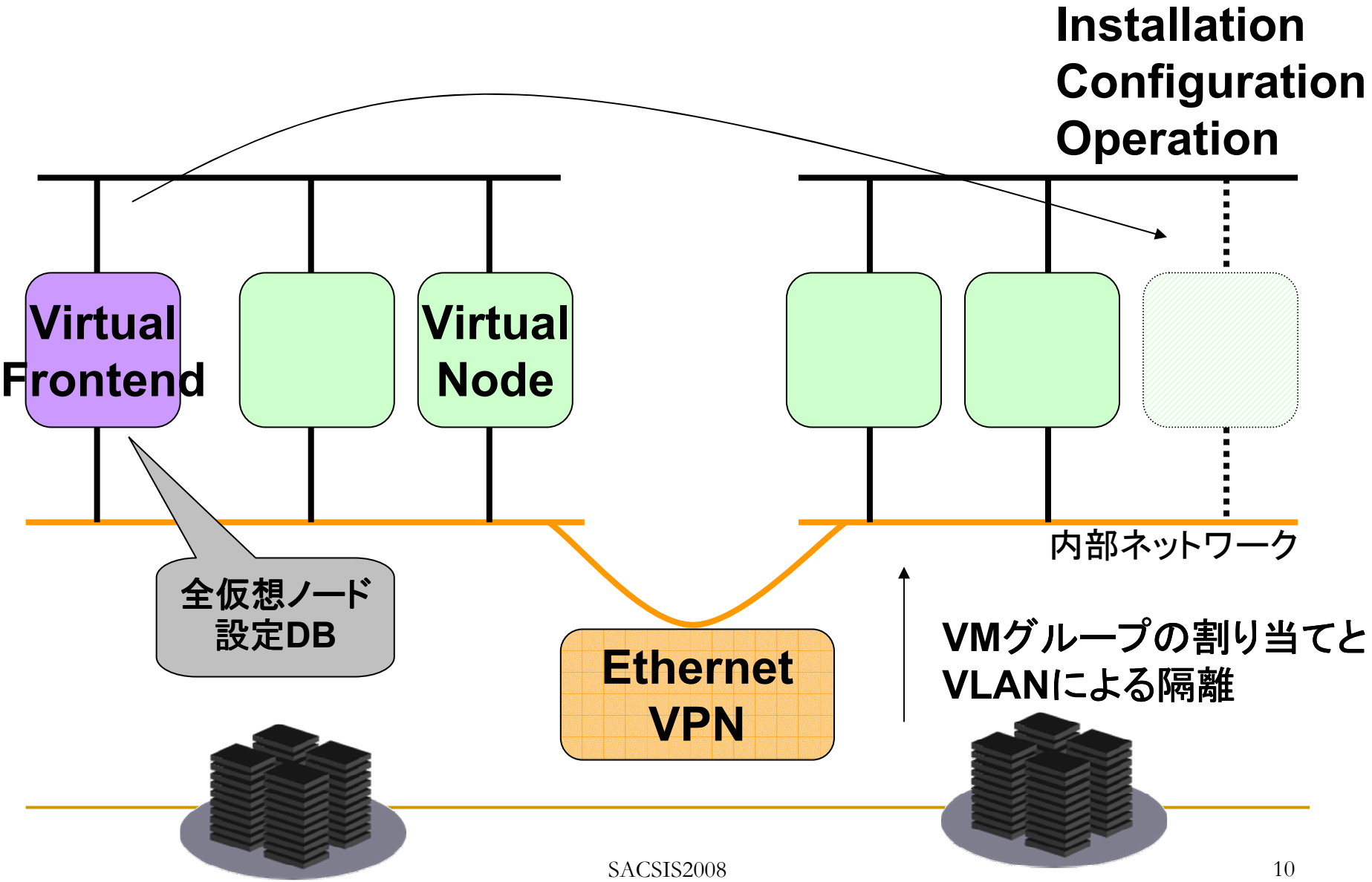
1. マルチサイト仮想クラスタ内部に物理クラスタ向けの大規模クラスタ管理システムを導入
   - 分散拠点の仮想ノード管理（追加・削除、再設定）
2. イーサネットVPNによる仮想クラスタ内部ネットワークの結合
   - 物理クラスタにおけるクラスタ管理資産の再利用
   - ユーザに対する透過性
3. 拠点ごとの透過的パッケージキャッシュ
   - パッケージインストールベースのクラスタ管理システム
   - 仮想ノードの迅速なインストール、再設定
   - きめこまかいカスタマイズ

物理クラスタ向け大規模クラスタ管理システム（NPACI Rocks）

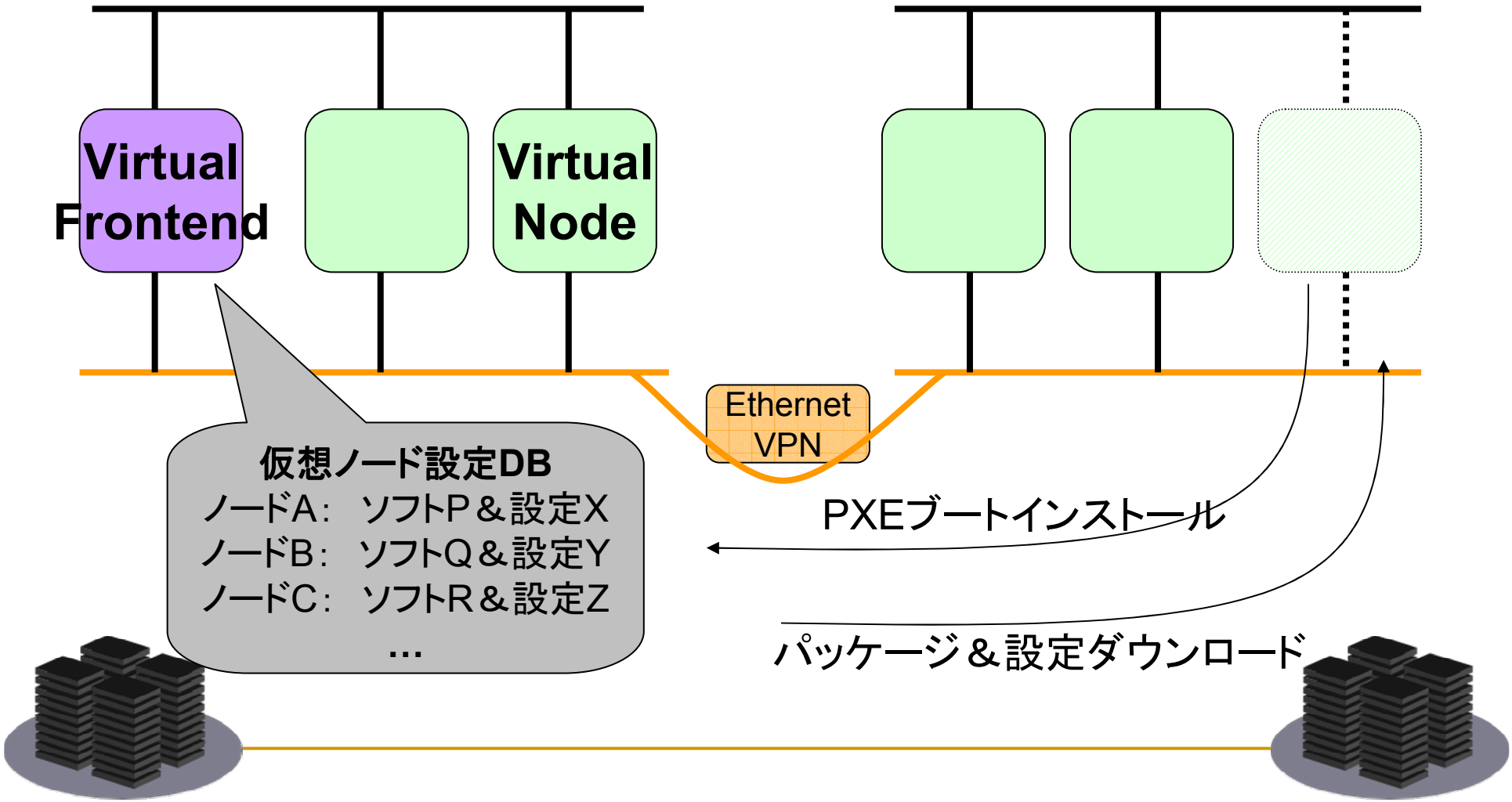


PXEブートインストールによる自動的なノードインストール
Rollによるクラスタワイドなアプリケーション設定
cluster-fork , tentakel等ツールによるクラスタ全体/一部へのコマンド実行
Ganglia等のモニタプロセスによるノード状態監視
ノードダウン時の自動復旧

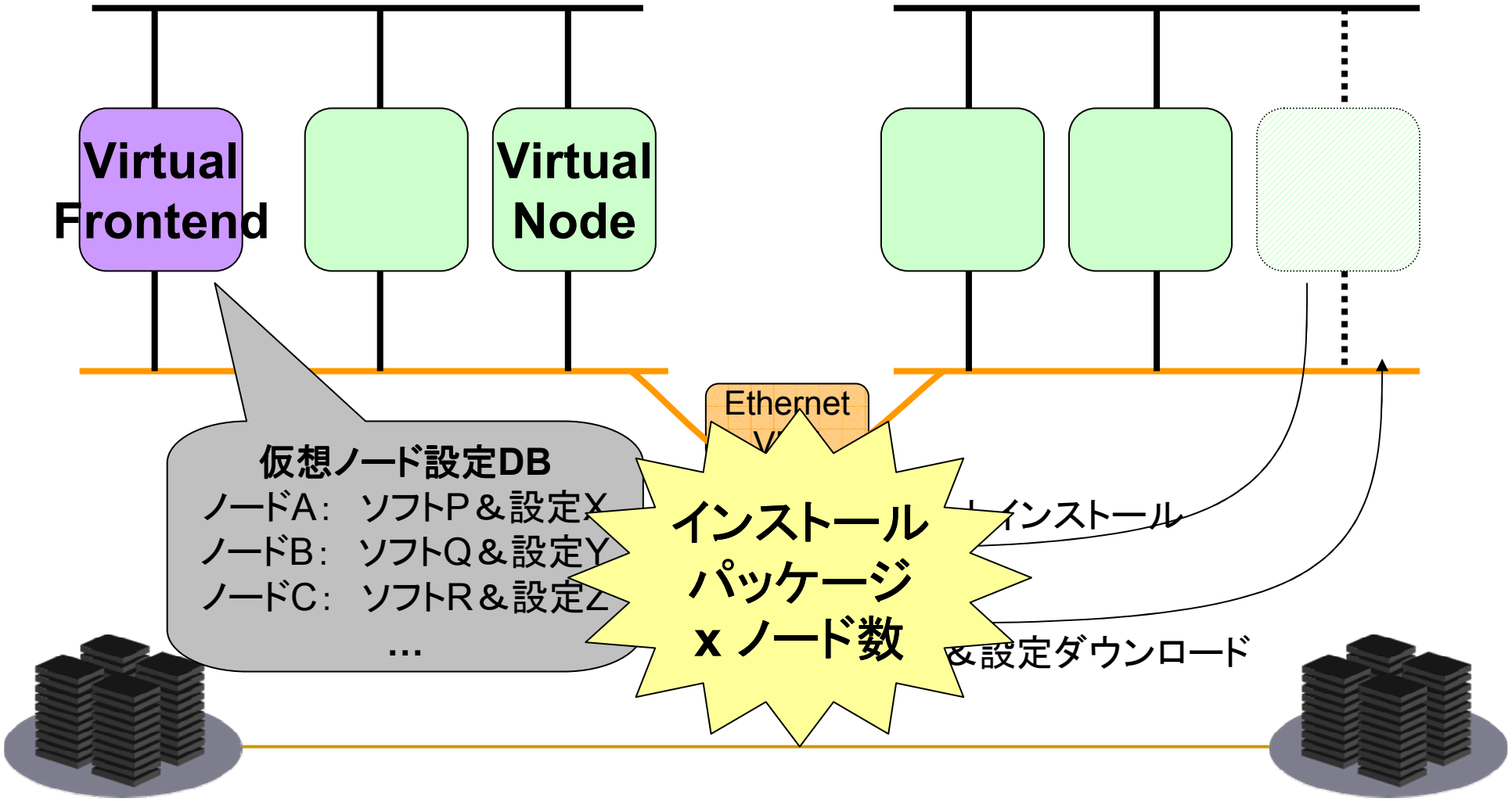
仮想クラスタ内部管理


Installation
Configuration
Operation
Virtual Frontend
Virtual Node
内部ネットワーク
全仮想ノード設定DB
Ethernet VPN
VMグループの割り当てとVLANによる隔離

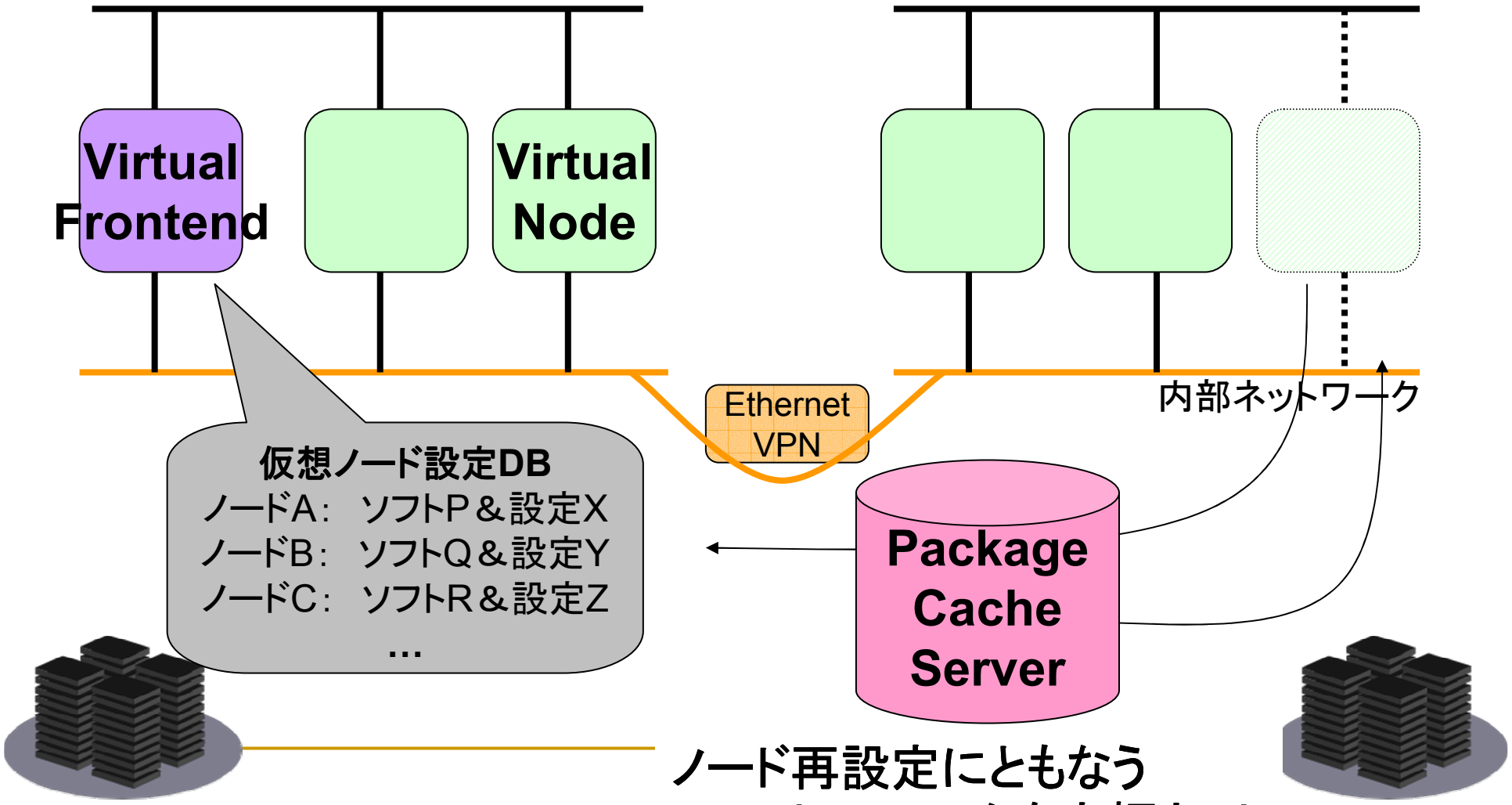
仮想ノード(再)設定と パッケージキャッシュ機構(1)



仮想ノード(再)設定と パッケージキャッシュ機構(2)



仮想ノード(再)設定と パッケージキャッシュ機構(3)

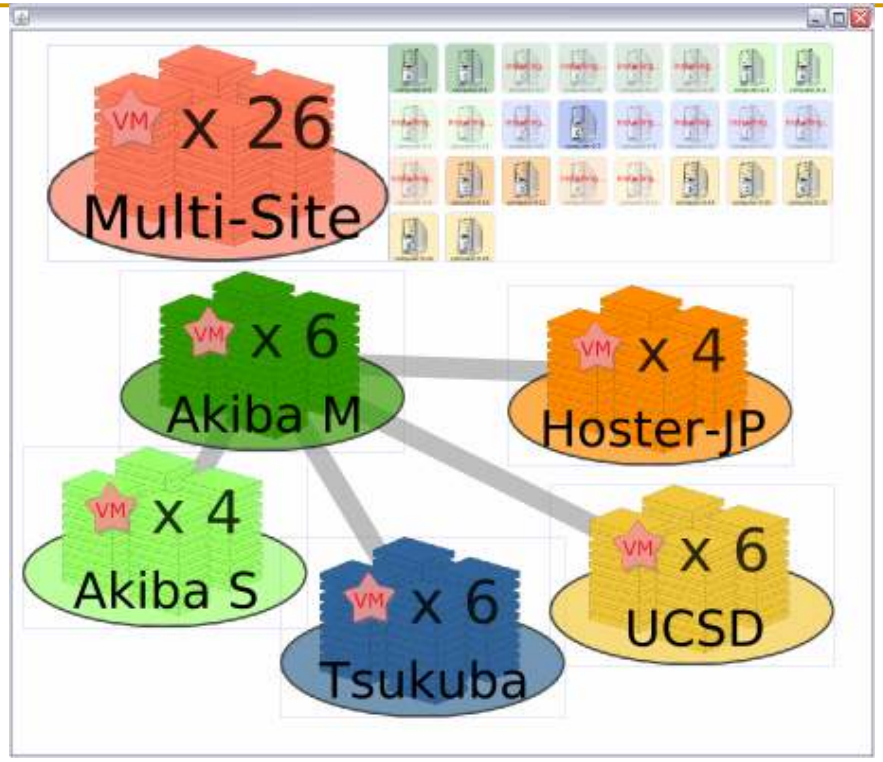


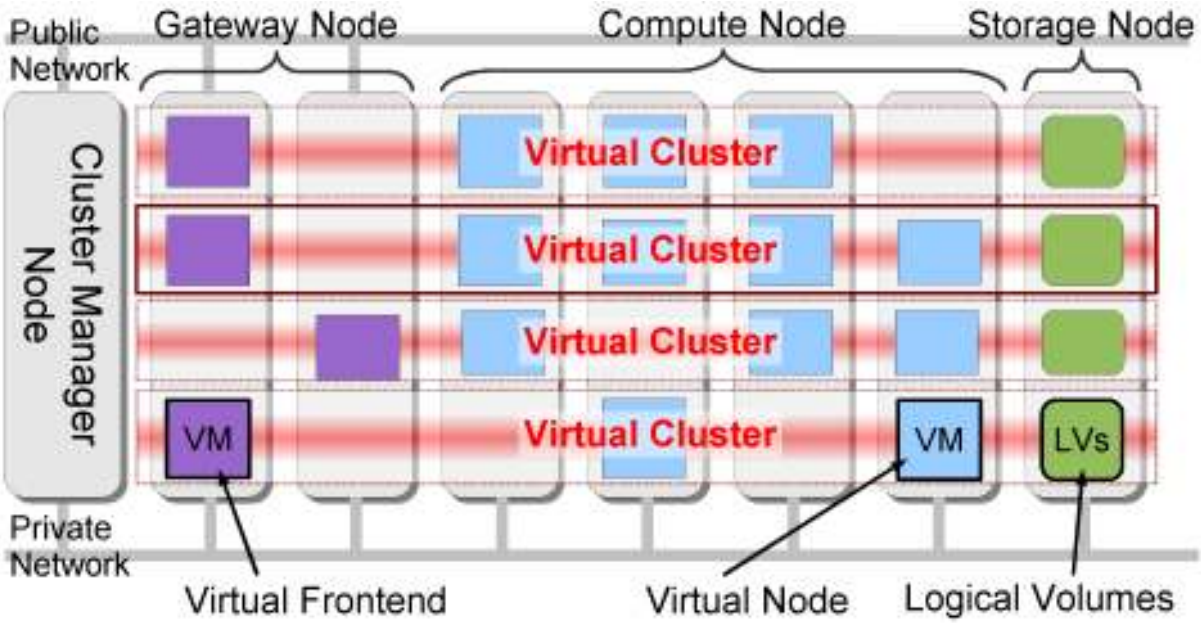
ノード再設定にともなう
WANトラフィックを大幅カット

実装

- **RESTベースAPI**
  仮想クラスタ割り当て
    VMの追加・削除
    VPN開始・停止
- **予約ポータル**
  各サイトの資源把握
  予約リクエスト発行

- NPACI Rocks 4.2
- OpenVPN 2.0
- Squid 3.0





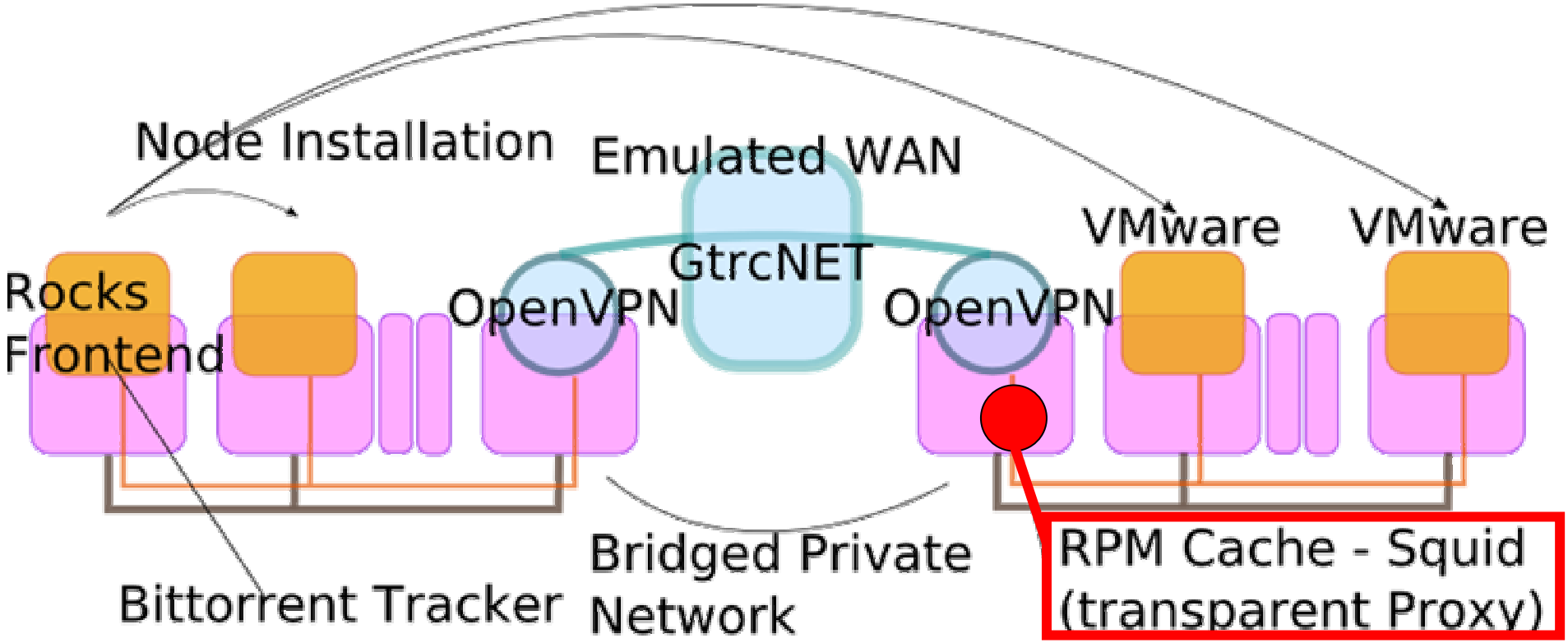
評価

- 動作状況
  - 分散仮想ノードの自動インストール
  - cluster-fork & tentakel クラスタワイドなコマンド実行ツール
  - Ganglia ノード状態のモニタリング

OK！

- 検証ポイント
  - 提案手法のWAN環境での妥当性
    - 大規模仮想ノード群の再設定
      - 仮想ノードの追加・離脱
      - 仮想ノードごとのカスタマイズ
    - 再設定に要する時間、トラフィック

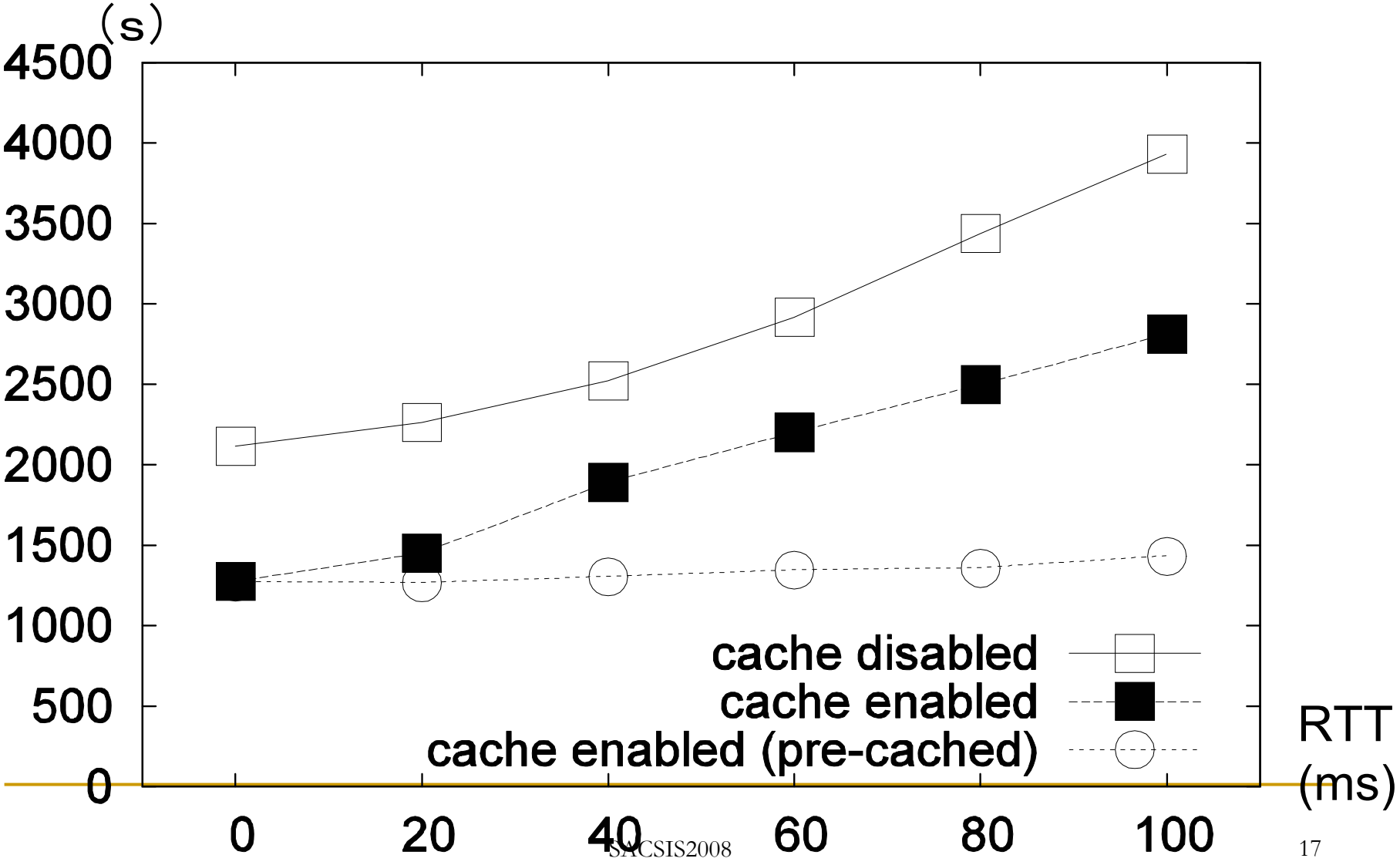
実験環境



AMD Opteron 244, 3GBメモリ, Gb Eth x2

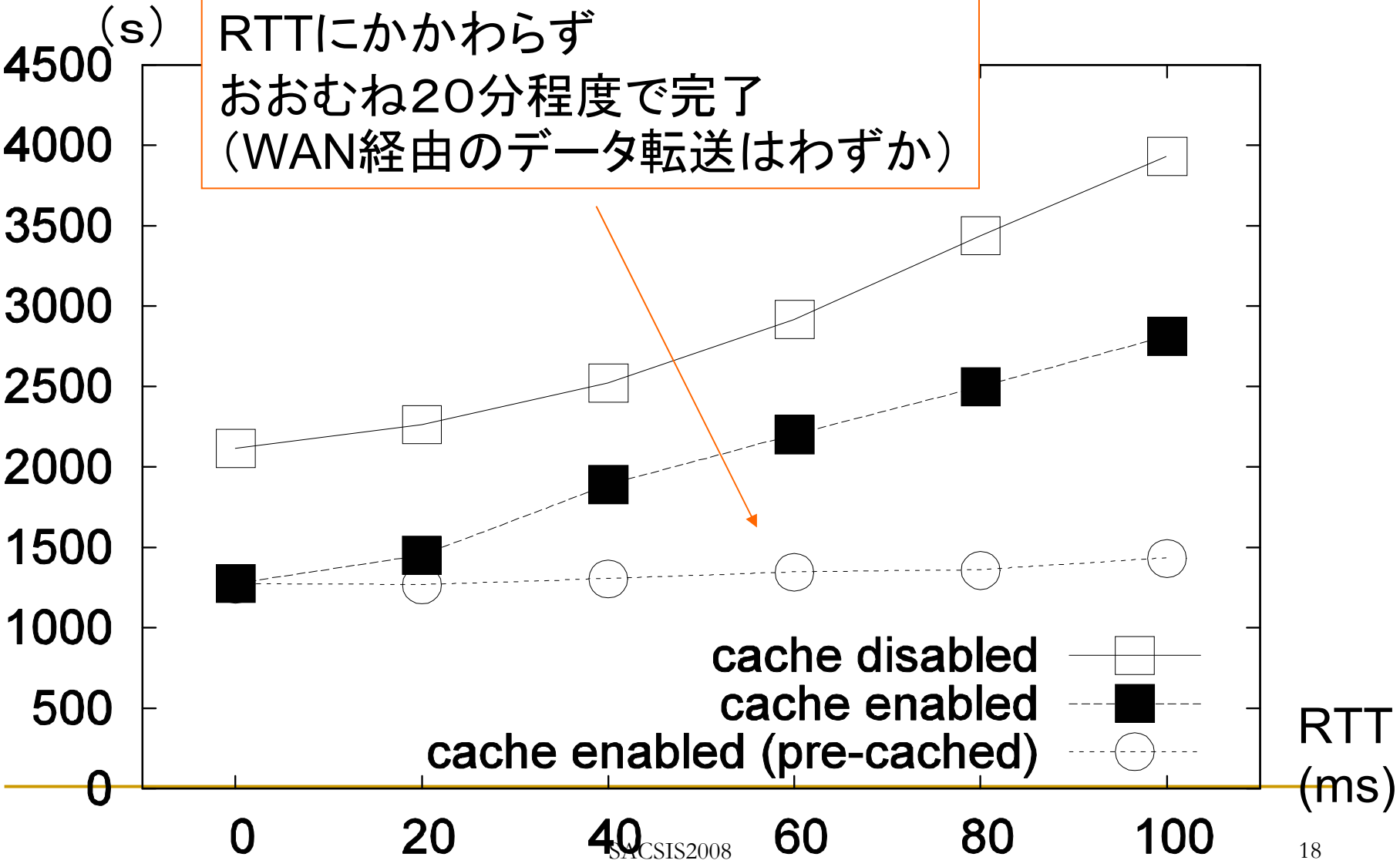
AMD Opteron 246, 6GBメモリ, Gb Eth x2

仮想クラスタ構築時間（遠隔134ノードの再構築）

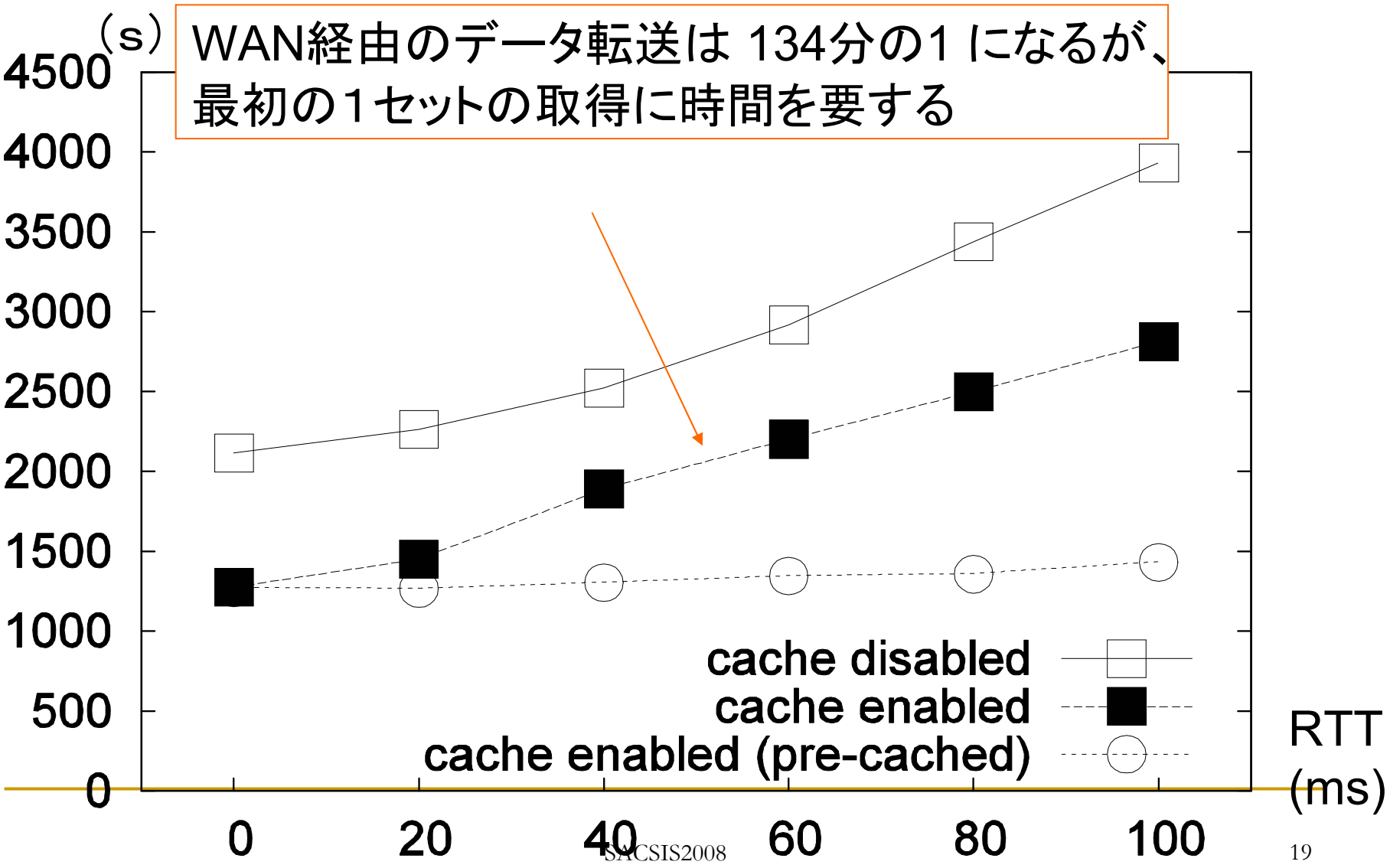


仮想クラスタ構築時間（遠隔134ノードの再構築）

あらかじめキャッシュ済みであれば、
RTTにかかわらず
おおむね20分程度で完了
（WAN経由のデータ転送はわずか）

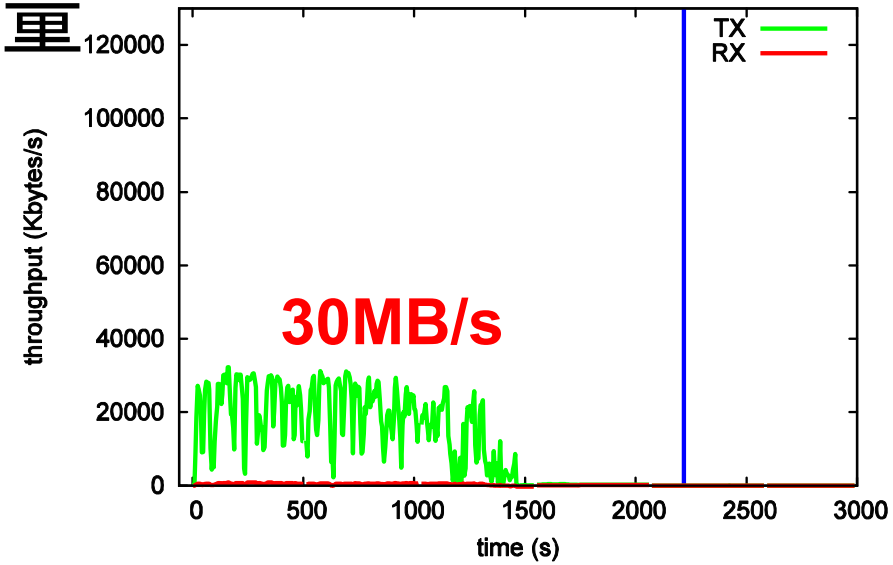


仮想クラスタ構築時間（遠隔134ノードの再構築）

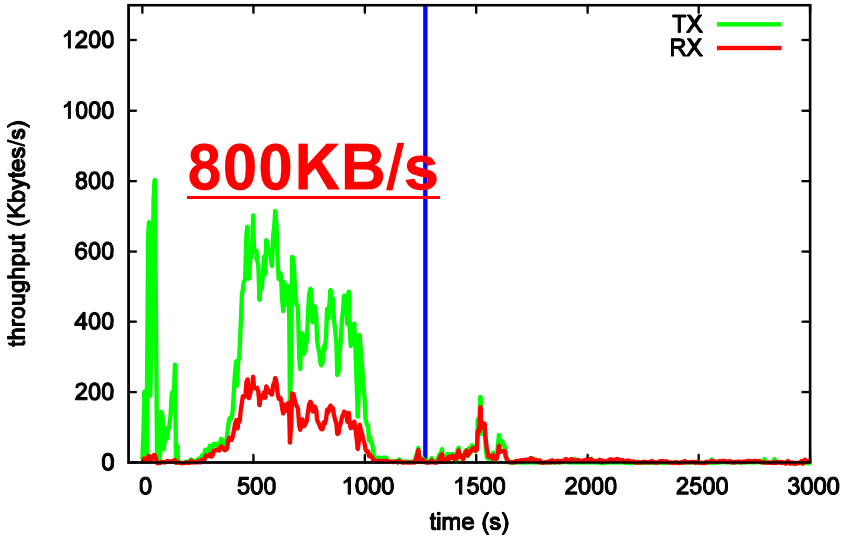


WAN経由のデータ転送量（RTT20msのとき）

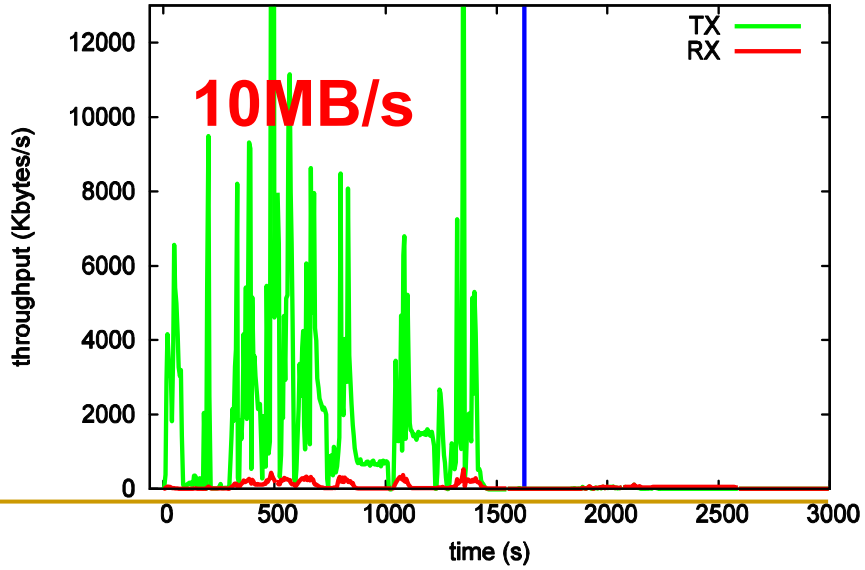
キャッシュ無効



キャッシュ済み



キャッシュ有効



キャッシュ機能により
WAN経由トラフィックは極小化

考察

- キャッシュにより仮想ノード数の増加に対してもWAN経由トラフィックはほぼ一定
  - 1ノードにインストールされるパッケージの総サイズ
  - あらかじめキャッシュしておけばさらに極小化
- 全ノードの再構築時間は20分程度
  - 約900MB分のパッケージをインストール
    - あらかじめキャッシュされている場合
    - RTT 20ms程度までの場合
  - 物理クラスタに対しても15-20分程度
    - Rocksの改良により短縮は可能

関連研究

- 多拠点にわたる仮想ノードを、物理クラスタ管理システムの枠組みで取り扱うものはない
- 仮想クラスタ＋VPN
  - Virtuoso
  - VioCluster
- Virtual Workspace, Amazon EC2
- InTrigger, Puppet
- PlanetLab

結論

- 仮想化による大規模アプリケーション実行環境の構築
  - 多拠点化
  - 管理コストの増大
  - 多様な設定
- マルチサイト仮想クラスタ
  - 仮想クラスタ内部へのクラスタ管理システムの導入
  - イーサネットVPNによる単一ビュー
  - パッケージベースインストーラとパッケージキャッシュ
- 評価
  - 大規模仮想ノード群を迅速に設定可能
  - WAN経由のデータ転送も極小化

今後の課題

- 仮想クラスタ管理システム
  - Xen対応作業中
  - 遠隔マイグレーション
  - ディスクレスブート
  - マルチOS
- 成果物
  - http://code.google.com/p/grivon/
  - 徐々に公開中