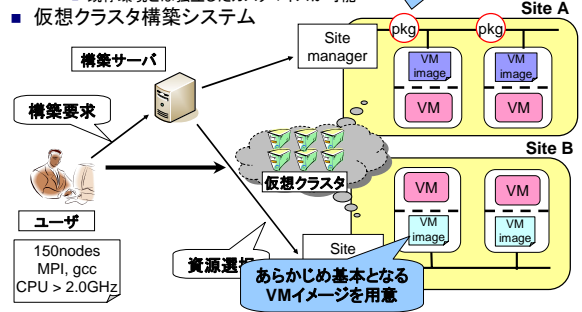


「仮想クラスタ構築時間のモデリングおよびその最適化」

学籍番号 03-25794 山崎翔平
指導教員 松岡 聡 教授

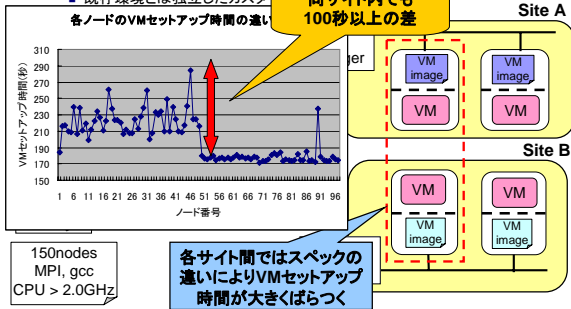
背景：既存の仮想クラスタ構築システム

- 仮想クラスタ
 - 仮想計算機(VM)を計算基盤として活用
 - 既存環境とは独立したカスタマイズが可能
- 仮想クラスタ構築システム



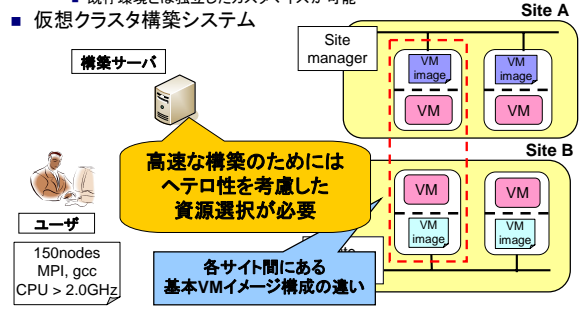
背景：既存の仮想クラスタ構築システム

- 仮想クラスタ
 - 仮想計算機(VM)を計算基盤として活用
 - 既存環境とは独立したカスタマイズが可能
- 仮想クラスタ構築システム



背景：既存の仮想クラスタ構築システム

- 仮想クラスタ
 - 仮想計算機(VM)を計算基盤として活用
 - 既存環境とは独立したカスタマイズが可能
- 仮想クラスタ構築システム



関連研究

- 仮想クラスタ構築プロジェクト
 - 例 VMPlants[Krsul et al., '04]、[Nishimura et al. '07]
 - 各マシンの性能差によるインストール時間の差は考慮していない
- 構築時間の予測、モデリング
 - VMイメージ転送時間予測[Sotomayor et al. '06]
 - イメージ転送時間だけのモデリング
 - ヘテロな資源を対象とする割り当ては考慮していない

提案

- 仮想クラスタ構築時間のモデリング
 - 実資源のヘテロ性を考慮
 - マシンスペック
 - 基本となるVMイメージ
- モデルを用いた資源選択による仮想クラスタ構築時間の高速化

本研究の目的と成果

■ 目的

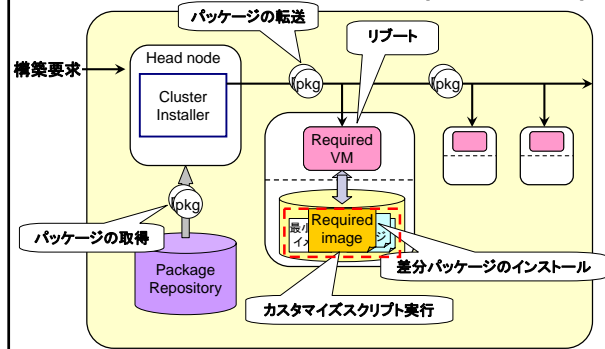
- 大規模環境における最適な仮想クラスタ構成資源の選択

■ 成果

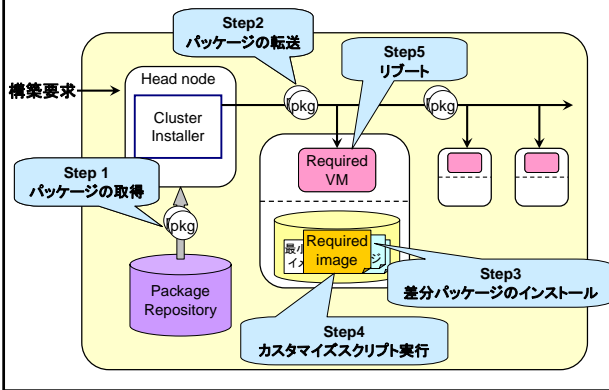
- 仮想クラスタ構築時間のモデリング手法
- モデルベースの複数サイトにまたがる仮想クラスタ構築シミュレータ
- モデルに基づいた資源選択手法の有効性をシミュレーションにより確認
 - 構築時間が最大33.7%減少

対象仮想クラスタ構築システムの概要

[Nishimura et al., '07]



VMセットアッププロセスのステップ分け



各VMセットアップ時間のモデル

Step1	パッケージダウンロード	$\alpha_1(PkgSize) + \beta_1$
Step2	パッケージ転送	$\alpha_2(PkgSize) + \beta_2(TransferOrder) + \gamma_2$
Step3	パッケージインストール	$\alpha_3(PkgSize) + \beta_3(CPUfreq)^{-1} + \gamma_3(DiskWrite)^{-1} + \delta_3$
Step4	カスタマイズスクリプト実行	$\alpha_4(PkgSize) + \beta_4(CPUfreq)^{-1} + \gamma_4(DiskWrite)^{-1} + \delta_4$
Step5	リポート	$\alpha_5(PkgSize) + \beta_5(CPUfreq)^{-1} + \gamma_5(DiskRead)^{-1} + \delta_5$

$PkgSize$: 差分インストールするパッケージサイズ (MB)
 $TransferOrder$: パッケージ転送順序
 $CPUfreq$: CPU周波数 (GHz)
 $DiskRead$: ディスク読み込み速度 (MB/s)
 $DiskWrite$: ディスク書き込み速度 (MB/s)

各係数を重回帰分析により実験的に決定

係数特定のための実験手法

■ 実験データについて

- 松岡研究室のPresto IIIクラスタ上で、既存システム[Nishimura et al., '07]による仮想クラスタ構築時のデータ
 - 約200VM
- 差分インストールするパッケージサイズパターン
 - 0, 5, ..., 150 MBの31通り

■ 実験環境 (Presto III)

	CPU	RAM	HDD	Network
構成0	Athlon2000+	1GB	IDE	Gigabit Ethernet
構成1	Opteron242	2GB	IDE	
構成2	Opteron280	4GB	SATA	
構成3	Opteron250	2GB	SCSI	

	OS	VMM	インストーラ	データ転送
	Linux-2.6.16	Xen 3.0.2-2	Lucie 0.0.5	Dolly+ 0.93-1

モデル

ステップ名		
パッケージダウンロード	$0.12 \times PkgSize + 0.72$	最大12秒の差
パッケージ転送	$0.022 \times PkgSize + 0.04 \times TransferOrder + 0.14$	
パッケージインストール	$0.784 \times PkgSize + 56 \times (CPUfreq)^{-1} + 808 \times (DiskWrite)^{-1} - 47$	
カスタマイズスクリプト	$0.016 \times PkgSize + 1.1 \times (CPUfreq)^{-1} + 13 \times (DiskWrite)^{-1} - 0.8$	最大118秒の差
リポート	$0.045 \times PkgSize + 9.9 \times (CPUfreq)^{-1} + 62 \times (DiskRead)^{-1} + 6.7$	最大25秒の差

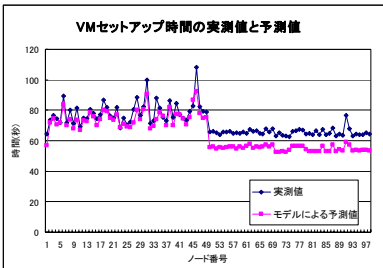
PrestoIIIのマシンスペックの最大値・最小値

	MAX	MIN
CPU周波数(GHz)	2.4	1.6
Disk read speed(MB/s)	69.3	18.4
Disk write speed(MB/s)	76.4	10.8

モデルの精度評価

0.9を上回る高精度なモデル

ステップ名	決定係数
パッケージダウンロード	0.9959
パッケージ転送	0.9978
パッケージインストール	0.9236
カスタマイズスクリプト実行	0.5593
リポート	0.468



差分インストールするパッケージサイズが45MBのときの各ノードのVMセットアップ時間

モデルを用いた資源選択の有効性の評価

■ シミュレーションによる資源選択方法の比較

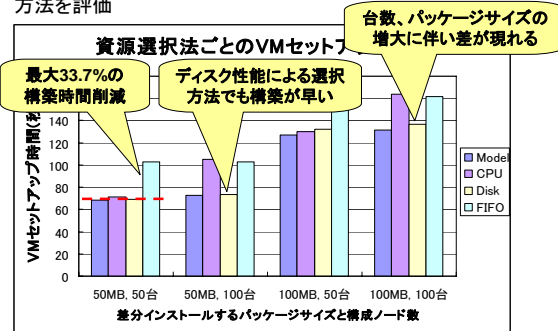
- モデル式
- CPU周波数のみ
- ディスク性能のみ
- FIFO

仮想クラスタ構築シミュレータの実装

- モデル式をもとにグリッド環境シミュレータ SimGrid[Casanova et al. '01]を用いて実装
 - 各ホストに属性値(CPU周波数、ディスク性能)を付加する機構を追加
- 複数サイトを含む任意の環境を設定可能
- 実行時に指定する引数
 - 資源選択方法
 - 構成ノード数
 - 各サイトで差分インストールするパッケージサイズ

シミュレーション実行結果

- 松岡研究室のPrestollIIクラスタ環境を再現し、各資源選択方法を評価



議論

- PrestollII環境ではディスクを評価すればおおよそのヘテロ性は考慮が可能
 - PrestollIIの環境はディスクのヘテロ性が高く、一方でCPUは2種類
- グリッド環境ではディスク性能のみでは不十分
 - CPU、ネットワーク性能差が極めて大きい
 - 例 Intrigger, Grid'5000
- モデル式は、各パラメータのヘテロ性を考慮
 - 自動的に最適な資源選択が可能

まとめと今後の課題

- まとめ
 - 仮想クラスタ構築時間のモデリング手法
 - モデルベースの複数サイトにまたがる仮想クラスタ構築シミュレータ
 - モデルに基づいた資源選択手法の有効性をシミュレーションにより確認
 - 構築時間が最大33.7%減少
- 今後の課題
 - モデルをベースにした選択資源を行うシステムの実装および検証
 - 構築時間だけでなく、構築時および使用時のフォールトを考慮した資源選択方法についての検討