

# クラウド基盤評価報告書

穉山 空道<sup>†‡</sup>

平塚 信明<sup>†‡</sup>

横山 重俊<sup>†§</sup>

<sup>†</sup> 東京大学大学院 情報理工学系研究科 創造情報学専攻 <sup>‡</sup> 国立情報学研究所 <sup>§</sup> (株)NTT データ

## 1. はじめに

国立情報学研究所 (以下 NII と呼ぶ) の持つ edubase-Cloud の改善を目的として、複数のクラウド基盤ソフトウェアの性能評価を行った。本報告書ではその詳細と結果について報告する。尚、今回は Eucalyptus[1] のバージョン 1.6.2 および 2.0 を評価した。

## 2. 評価環境

本章では評価対象のハードウェアおよびソフトウェア環境について述べる。

### 2.1 ハードウェア

NII の 17 階に設置されている 11 台のラックマウント型サーバークラスタのうち 10 台を使用した。一台のスペックは表 1 の通りである。

表 1: 物理マシンのスペック

CPU	Intel Xeon X5460 (4 cores)
Memory	7800 KiB
Disk	250GB HDD
Network	1000Base-T Ethernet NIC × 3

上記スペックのマシン 10 台を LAN で繋ぎ、10.1.2.1 から 10.1.2.5 および 10.1.2.11 から 10.2.1.15 のプライベート IP アドレスを割り当てた。10.1.2.2 の IP を持つマシンのみ 136.187.36.230 のグローバル IP アドレスも同時に持ち、インターネットに接続されている。その他のマシンがインターネットにアクセスする際は 10.1.2.2 を経由してアクセスする。以上を図に表したものが図 1 である。

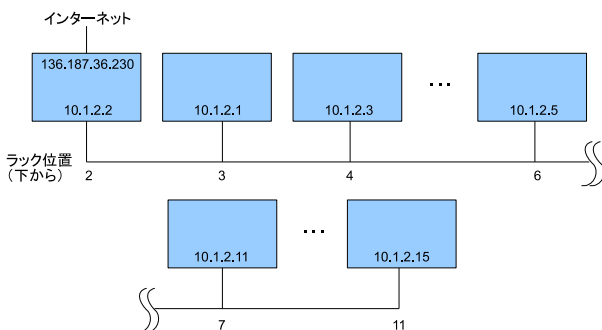


図 1: 物理マシンのネットワーク構成

10 台のマシンのうち 5 台ずつを Eucalyptus 1.6.2 用と 2.0 用にそれぞれ割り当ててインストールを行った。詳細な割り当ては表 2 の通り。

表 2: マシンの割り当て

10.1.2.11	1.6.2 の CLC, CC
10.1.2.12-10.1.2.15	1.6.2 の NC
10.1.2.2	2.0 の CLC, CC
10.1.2.1, 10.1.2.3-10.1.2.5	2.0 の NC

尚、インストールに際しては [2] および [3] を参考にした。

### 2.2 ソフトウェア

本章では評価対象のソフトウェア環境について述べる。まず表 3 に、Eucalyptus よりも下層のソフトウェア環境 (1.6.2 用と 2.0 用のマシンで共通) を示す。

表 3: 共通ソフトウェア環境

OS	CentOS release 5.5 (Final) 64bit
Xen	3.0.3-105.el5_5.5
libvirt	0.6.3
dhcpd	isc-dhcpd-V3.0.5-RedHat
SELinux	disabled

次に表 4 に Eucalyptus の設定のうちデフォルトから変更を行ったものを示す。表中の VNET\_PUBLICIPS は上位 3 バイト (136.187.36) を省略して示している。

表 4: Eucalyptus の設定

Eucalyptus のバージョン	1.6.2	2.0
フロントエンドの設定		
VNET_MODE	MANAGED	
VNET_SUBNET	192.168.0.0	10.2.0.0
VNET_NETMASK	255.255.0.0	
VNET_DNS	136.187.17.3	
VNET_ADDRSPERNET	32	
VNET_PUBLICIPS	***.236 ***.237 ***.238 ***.239 ***.240 ***.241	***.229 ***.231 ***.232 ***.233 ***.234 ***.235
VNET_PUBINTERFACE	eth1	eth0
VNET_PRIVINTERFACE	eth1	
ノードの設定		
VNET_BRIDGE	xenbr1	

VNET\_SUBNET は同一 LAN 内で使用していないものにする必要があるため、1.6.2 と 2.0 で揃えることができない。ただし違う物であれば何でもよいため、今回は 192.168.0.0/16 とした 1.6.2 の設定を 10.3.0.0/16 に

することは可能である。また VNET\_PUBINTERFACE が 1.6.2 と 2.0 で異なるのはネットワーク構成の非対称性 (図 1, 表 2) による。1.6.2 のフロントエンドでは eth1(10.1.2.11) のみ有効になっており LAN 内のマシンおよびインターネット共に eth1 側にあるが、2.0 のフロントエンドでは eth0(136.187.36.230) と eth1(10.1.2.2) が有効になっておりインターネットは eth0 側に、LAN 内のマシンは eth1 側にある。

Eucalyptus 上で動作させるマシンイメージは [5] にある Cent OS 5.3 の 64bit 版を使用した。Eucalyptus では起動する仮想マシンの Type に m1.small, c1.medium, m1.large, m1.xlarge, c1.xlarge を選べ、各 Type のスペックを自由に設定することができる。本報告書では表 5 の通り設定した。これはデフォルトの設定のままである。

表 5: 仮想マシンスペック

Type	CPUs	Memory (MB)	Disk (GB)
m1.small	1	128	2
c1.medium	1	256	5
m1.large	2	512	10
m1.xlarge	2	1024	20
c1.xlarge	4	2048	20

### 3. 評価結果

本章では Eucalyptus 1.6.2 と 2.0 の評価結果を示す。評価は大きく 3 つの分類で行った。以下では Eucalyptus 上で実行される仮想マシン (Virtual Machine) のことを VM と呼ぶ。

1. Eucalyptus の機能評価
  - (a) euca2ools の動作チェック
2. 単体 VM レベルでのベンチマーク
  - (a) VM の起動速度の計測
  - (b) unixbench による評価
3. VM 連携レベルでのベンチマーク
  - (a) NAS Parallel Benchmark による評価

#### 3.1 Eucalyptus の機能評価

Eucalyptus の機能評価として、Euca2ools[4] の動作確認を行った。Euca2ools は Eucalyptus を操作するためのコマンドラインツール群であり、仮想マシンの起動/終了や仮想マシンイメージのアップロードなど、Eucalyptus を使用するのに必要な機能はすべて Euca2ools から利用できる。

本章では Euca2ools を Eucalyptus 1.6.2 と 2.0 に対して使用したとき、問題のあったコマンドについて述べる。使用した Euca2ools のバージョンは 1.3.1 である。

**euca-attach-volume** ボリューム (ディスクイメージ) を仮想マシンに接続する。接続されたボリュームが仮想マシンから何という名前で見えるかを指定でき、manpage と -help の出力には名前について制限は書かれていない。しかし実際は仮想マシン上で動く linux がデバイスファイルの名前で使用するドライブを決定するため、sda, sdb, hda などのブロックデバイスに割り当てられた名前しか使えない。これは linux の事情であってバグではない。

ボリュームを接続するプロトコルとして Eucalyptus 2.0 では iSCSI と ATA over Ethernet (以下 AoE) が、Eucalyptus 1.6.2 では AoE のみが選べる。しかし、評価環境のネットワーク設定 (第 2 章を参照のこと) では AoE を用いたボリュームの接続ができなかった。AoE ではデバイス (ブロックデバイスやディスクイメージ) をネットワークに向けて公開する際、どのネットワークデバイス (eth0, eth1 など) に向けて公開するかを指定する。評価環境では CLC, CC, NC はすべて eth1 で繋がっているため eth1 に向けてデバイスを公開すべきである。しかし euca-attach-volume を用いると自動的に eth0 に公開されボリュームの接続ができなかった。AoE を手動で設定して (すなわち euca-attach-volume を使わずに) eth1 にデバイスを公開した場合にはボリュームの接続ができることを確認した。なお Eucalyptus ではボリュームはまず EBS から AoE で NC に接続され、さらに NC から Xen のリソース割り当て機能 (xm block-attach) で仮想マシンに接続される。

**euca-reboot-instances** 仮想マシンを再起動する。仮想マシンを複数指定できるが、Eucalyptus 2.0 では 1 番目に指定したもののしか再起動されなかった。Euca2ools および下層のライブラリのソースコードにはバグが発見できなかったこと、Eucalyptus 1.6.2 では指定した仮想マシンが全て再起動することから、Eucalyptus 2.0 のバグであると考えられる。

**euca-run-instances** 仮想マシンを起動する。今回の評価では合計 16 コアあり仮想マシンの 1 コアに物理マシンの 1 コアを割り当てる設定にしているため、例えば 2 コアの m1.large なら 8 台同時起動できる。しかし、Eucalyptus 1.6.2 では m1.large を 8 台同時起動すると高い確率で 1 から 3 台程度が起動できなかった。本バグの所在は Euca2ools ではなく Eucalyptus だと考えられ、Eucalyptus 2.0 および edubaseCloud に使われている改変済みの Eucalyptus 1.6.2 でも修正済みであると考えられる。

euca-bundle-vol 起動中の仮想マシンのファイルシステムを保存し、同じ状態で起動できる仮想マシンイメージにまとめる。Eucalyptus 2.0 と Euca2ools 1.3.1 の組み合わせでは本コマンドで作成した仮想マシンイメージが起動できなかった。本問題は Euca2ools のバグであると考えられ、公式のフォーラムでも報告されている [11]。これによれば Euca2ools のマイナーバージョンレベルでバグが発現したりしなかったりしている。実際に Euca2ools を 1.2 に変更して試したところ、起動直後のマシンから作成したイメージが起動できたが、多くのソフトウェアをインストールした後のマシンから作成したマシンは起動できなかった。

### 3.2 単体 VM レベルでのベンチマーク

単体 VM レベルでの評価では、次の二種類を行った。

**VM の起動速度の計測** VM の起動にかかる時間を VM 同時起動数やイメージサイズを変えて計測した。

**unixbench による評価** VM の単体での性能をベンチマークソフトによって計測した。

**bonnie++による評価** I/O の性能を VM のローカルディスクと EBS に対して行った。

#### 3.2.1 VM の起動速度の計測

VM の起動にかかる時間は、ユーザビリティやサービス運用時のスケールにかかる時間の観点から重要である。そこで Eucalyptus1.6.2 と 2.0 で VM の起動にかかる時間を計測、比較した。ただしここで VM の起動にかかる時間とは、Euca2ools の euca-run-instances で VM 起動命令を発行してから、euca-describe-instances で確認できる status が running になるまでの時間である。実際には running になった後に sshd や httpd などの各種 daemon が起動するため、この時間は本当に使えるようになるまでの時間ではない。

計測は同時に起動する VM の数とマシンイメージのサイズを変化させ、それぞれ 5 回の平均を取った。結果を表 6、表 7 に示す。

表 6: VM 起動の所要時間 (秒), Eucalyptus 1.6.2

イメージサイズ \ 同時起動数	1	4	8
2GB	106	112	216
5GB	246	262	509
10GB	508	552	*
20GB	990	*	*

表 6 中の\*は 3.1 節で述べたバグにより計測ができなかったことを表す。また本計測はマシンイメージのキャッシュが NC の HDD に存在する状態で行った。すなわち

表 7: VM 起動の所要時間 (秒), Eucalyptus 2.0

イメージサイズ \ 同時起動数	1	4	8
2GB	131	126	242
5GB	264	301	581
10GB	518	600	1150
20GB	1008	1178	2160

起動時間にはマシンイメージをリポジトリから NC までネットワーク越しにコピーする時間は含まれない。

#### 3.2.2 unixbench による評価

unixbench[6] は、「Unix 全体の性能を測定する合成ベンチマーク [7]」であり、浮動小数点計算、ファイル IO、プロセス生成、グラフィック性能などを計測する。ただしクラウド上の VM でゲームをする等の使い方は通常行われないため、グラフィック性能を計測するベンチマークはオフにした。

Eucalyptus1.6.2 と 2.0 上で 1 コア、2 コア、4 コアの VM を起動し、unixbench を実行した。実行時には各クラウド上に VM が一台のみ存在する状態で行った。また物理レイヤーでの結果との比較のため 2.0 の NC 上でも測定を行った。unixbench のバージョンは 5.1.2、コンパイルに使用した gcc のバージョンは 4.1.2 である。

図 3、図 5、図 7 が Eucalyptus1.6.2 での結果、図 2、図 4、図 6 が Eucalyptus2.0 での結果、図 8 が NC での結果である。unixbench はマルチコア環境を認識し、コア数分のマルチスレッドでの測定とシングルスレッドでの測定を自動的に行う。各図は色の薄い棒グラフがシングルスレッドでの結果、色の濃い棒グラフがマルチスレッドでの結果である。

結果の横軸はテストの種類、縦軸はベースラインに対する倍率を表す。なお結果の最大が約 6000 となっているが、これは unixbench が古くから使われておりベースラインが 1995 年時点のデータだからである。テストの種類とその説明、ベースラインを表 8 に示す。説明の英語原文は [6] にある。

表 8: テストの種類, 説明, ベースライン

種類	説明	ベースライン
1	Dhrystone: マシンの総合性能を測る合成ベンチマーク。主に文字列処理を行い, 浮動小数点演算は行わない。	116700 lps (loops per second)
2	Whetstone: 科学技術計算の性能を測る合成ベンチマーク。整数演算, 浮動小数点演算, 配列アクセスを統合的に行う。	55 MWIPS (Mega Whetstone Instructions Per Seond)
3	Execl Throughput: execl 関数が 1 秒に何回実行できるか。	43 lps
4	File Copy 1: ファイルのコピー速度。バッファサイズ 1024 バイト。	3960 KBps (Kilo Bytes per second)
5	File Copy 1: ファイルのコピー速度。バッファサイズ 256 バイト。	1655 KBps (Kilo Bytes per second)
6	File Copy 1: ファイルのコピー速度。バッファサイズ 4096 バイト。	5800 KBps (Kilo Bytes per second)
7	Pipe Throughput: 512 バイトのデータをパイプに書き読み戻す操作が 1 秒間に何回できるか。	12440 lps
8	Pipe-based Context Switching: 2 つのプロセスがパイプを通じて整数を 1 秒間に何回やりとりできるか。	4000 lps
9	Process Creation: プロセス生成 (子プロセスは fork の直後に exit する) が 1 秒間に何回できるか。	126 lps
10	Shell Scripts (1 concurrent): あるシェルスクリプトを 1 分間に何回実行できるか。	42 lpm (loops per minute)
11	Shell Scripts (8 concurrent): あるシェルスクリプトを 1 分間に何回実行できるか。ただしシェルスクリプトを 8 並列で実行する。	42 lpm (loops per minute)
12	System Call Overhead: システムコールのオーバーヘッド。システムコール (getpid) が 1 秒間に何回実行できるか。	15000 lps
13	System Benchmarks Index Score: 全スコアの平均	-

### 3.2.3 bonnie++による評価

bonnie++[12] は I/O の性能を測るために広く使われるベンチマークであり, 読み込み, 書き込み, ファイル作成, シークの速さを計測する。

Eucalyptus 2.0 上の仮想マシンから仮想マシンのローカルディスクおよび EBS 上のボリュームに対して bonnie++を実行した。仮想マシンのスペックは m1.small, ボリュームのサイズは 5GB で, ボリュームの接続プロトコルには iSCSI を利用した。AoE を利用した場合についてはボリュームの接続自体ができなかった (3.1 節参照) ため計測していない。また EBS は CC が兼任しており, EBS への書き込みとはネットワークを経由した CC のディスクへの書き込みである (図 9)。

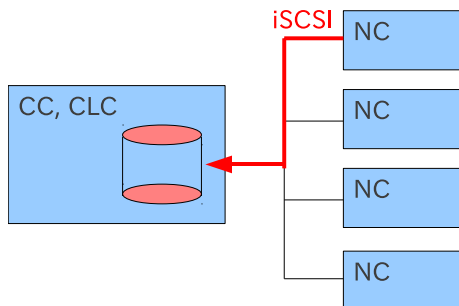


図 9: EBS へのアクセス

結果を表 10 に示す。上から順に Eucalyptus 1.6.2 でのローカルディスクのベンチマーク, Eucalyptus 2.0 でのローカルディスクのベンチマーク, Eucalyptus 2.0 での EBS のベンチマークである。各結果の読み方を表 9 に示す [14]。各項目の %CP は当該ベンチマーク実行時の平均 CPU 使用率である。また結果はそれぞれ 5 回分を示してある。

表 9: bonnie++の結果の読み方

Sequential Write	
Per Chr	1 バイト毎のファイル書き込み速度
Block	ブロック毎のファイル書き込み速度
Rewrite	読み込み後内容を変更して書き戻す速度
Sequential Input	
Per Chr	1 バイト毎のファイル読み込み速度
Block	ブロック毎のファイル読み込み速度
Random Seeks	
1 秒間に実行できた lseek(2)[13] の回数	
Sequential Create	
Create	ファイル名順のファイル作成速度
Read	ファイル名順のファイル状態取得速度
Delete	ファイル名順のファイル削除速度
Random Create	
Create	ランダム順のファイル作成速度
Read	ランダム順のファイル状態取得速度
Delete	ランダム順のファイル削除速度

```

Eucalyptus 1.6, ローカルディスク
-----Sequential Output----- --Sequential Input-- --Random--
-Per Chr- --Block-- --Rewrite-- -Per Chr- --Block-- --Seeks--
K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
4804 6 76907 15 29317 4 75809 93 816828 26 36338.1 53
77152 95 124855 25 18056 2 81121 92 789720 31 1793.1 3
71416 94 104076 21 17966 2 80743 93 842212 17 39933.5 33
66761 92 56223 11 18293 2 73721 94 805868 29 44031.4 94
71891 91 29086 6 25174 3 80422 93 780817 48 35164.2 68
-----Sequential Create----- --Random Create-----
--Create-- --Read-- --Delete-- --Create-- --Read-- --Delete--
/sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
10553 15 323858 94 98110 100 4731 6 418760 102 4709 6
4624 6 319588 93 70798 96 3185 4 423720 103 4819 6
8447 16 320502 93 96207 101 77581 98 416703 101 75174 100
4511 6 319850 93 84720 99 60835 99 579759 99 85316 97
52174 99 319650 93 74038 101 3180 4 405695 99 8600 10

```

```

Eucalyptus 2.0, ローカルディスク
-----Sequential Output----- --Sequential Input-- --Random--
-Per Chr- --Block-- --Rewrite-- -Per Chr- --Block-- --Seeks--
K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
15412 23 15739 3 13231 0 43190 49 51952 0 77.2 0
14537 19 16308 3 12945 0 46180 54 52047 0 81.6 0
15563 20 15450 3 13256 0 28747 34 51992 0 81.7 0
15960 21 15422 3 12624 0 32149 37 52757 0 82.0 0
16467 21 16228 3 13517 0 45008 50 51567 0 80.7 0
-----Sequential Create----- --Random Create-----
--Create-- --Read-- --Delete-- --Create-- --Read-- --Delete--
/sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
68628 97 468595 102 108034 100 20617 25 426598 93 104917 99
54846 85 327699 104 76887 99 65726 91 596347 101 105184 100
58762 97 469844 103 86541 90 70628 89 595154 101 89058 100
57590 82 322932 102 82762 99 52674 96 495015 96 107036 99
67597 95 465043 90 106959 101 71076 85 473881 92 90312 99

```

```

Eucalyptus 2.0, EBS
-----Sequential Output----- --Sequential Input-- --Random--
-Per Chr- --Block-- --Rewrite-- -Per Chr- --Block-- --Seeks--
K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP /sec %CP
22814 28 24127 5 13314 0 33032 35 47974 0 345.1 0
22393 29 31366 6 9442 0 35492 38 51385 0 385.0 0
20469 25 20678 4 9986 0 37074 41 50482 0 405.7 0
23406 28 20892 4 11082 0 35904 40 51024 0 405.1 0
25078 30 20506 4 11836 0 34729 38 49144 0 463.2 0
-----Sequential Create----- --Random Create-----
--Create-- --Read-- --Delete-- --Create-- --Read-- --Delete--
/sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP /sec %CP
63508 99 466062 102 106668 98 80111 99 596347 101 105085 100
56091 94 327555 95 80383 100 71877 98 428137 94 87490 98
62590 100 469215 103 108410 97 70609 99 595695 101 93072 99
46770 68 324472 95 60862 87 59289 99 598153 102 93198 100
73324 98 460185 101 92009 101 64876 99 597020 102 78251 99

```

表 10: bonnie++の結果

### 3.3 VM 連携レベルでのベンチマーク

VM 連携レベルでのベンチマークでは, NAS Parallel Benchmarks[8] (以下 NPB と呼ぶ) による評価を行った. NPB は MPI を用いた並列計算の性能を測るベンチマークであり, NASA によって開発され広く使われている. 並列計算に用いるタスクは以下の 8 つから任意に選ぶことができる. 各タスクの説明は [9] より引用した.

- EP 乗算合同法による一様乱数, 正規乱数の生成
- MG 簡略化されたマルチグリッド法のカーネル
- CG 正値対称な大規模疎行列の最小固有値を求めるための共役勾配法
- FT FFT を用いた 3 次元偏微分方程式の解法
- IS 大規模整数ソート
- LU Symmetric SOR iteration による CFD アプリケーション
- SP Scalar ADI iteration による CFD アプリケーション
- BT 5x5 block size ADI iteration による CFD アプリケーション

#### 3.3.1 コア数が一定の場合

m1.small(1 コア) を 16 台, m1.large(2 コア) を 8 台, c1.xlarge(4 コア) を 4 台それぞれ起動して 16 コアのク

ラストとして NPB を実行し, Eucalyptus 1.6.2 と 2.0 で結果を比較した. ただし表 5 に示した通り合計コア数は同じでもディスクサイズとメモリ容量は異なる.

Eucalyptus 1.6.2 での結果を表 11 に, 2.0 での結果を表 12 に示す. 評価環境では FT のコンパイルができなかったため FT 以外の 7 タスクを実行した. また実行に用いた MPI の実装は Open MPI バージョン 1.4.3[10] である.

表 11: コア数一定での NPB の結果 (1.6)

	m1.small	m1.large	c1.xlarge
EP	20 sec	19 sec	20 sec
MG	9657 sec	42 sec	48 sec
CG	230 sec	221 sec	244 sec
IS	24 sec	24 sec	19 sec
LU	216 sec	216 sec	178 sec
SP	614 sec	624 sec	600 sec
BT	206 sec	216 sec	221 sec

表 12: コア数一定での NPB の結果 (2.0)

	m1.small	m1.large	c1.xlarge
EP	19 sec	19 sec	19 sec
MG	14661 sec	49 sec	47 sec
CG	183 sec	209 sec	242 sec
IS	26 sec	19 sec	28 sec
LU	219 sec	222 sec	180 sec
SP	615 sec	627 sec	603 sec
BT	212 sec	218 sec	220 sec

なお m1.small の MG が非常に遅いのはメモリ不足でスワップアウトが発生したためである.

#### 3.3.2 コア数が異なる場合

次に m1.small(1 コア) の台数を 2 台, 4 台, 8 台, 16 台と変えて NPB を実行した. ただし前小節で発生したようなメモリ不足を防ぐため, メモリは 1VM あたり 1800MB に設定した. また物理マシンに対する VM の割り当ては, 起動している VM が最小の物理マシンを選ぶようになっている (すなわち 2, 4 台では最大 1VM/物理マシン, 8 台では 2VM/物理マシン, 16 台では 4VM/物理マシンである).

結果を表 13, 表 14 に示す. BT, SP ベンチマークは NPB の仕様上コア数が平方数でないと実行できないため, 2 コアと 8 コアの結果には含まれていない.

## 4. 結果の考察

### 4.1 Eucalyptus の機能評価に関する考察

Eucalyptus の機能評価では, 4 つのコマンドに問題が発見された. このうち euca-attach-volume と euca-reboot-instances は些末な問題であり, 使用上少し注意すれば問題ない. 具体的には, euca-attach-volume では

表 13: コア数を变化させた場合の NPB の結果 (1.6)

	2 コア	4 コア	8 コア	16 コア
EP	157 sec	79 sec	39 sec	19 sec
MG	- sec	53 sec	- sec	45 sec
CG	212 sec	149 sec	190 sec	231 sec
IS	29 sec	18 sec	16 sec	22 sec
LU	1083 sec	414 sec	277 sec	216 sec
SP	-	692 sec	-	601 sec
BT	-	254 sec	-	204 sec

表 14: コア数を变化させた場合の NPB の結果 (2.0)

	2 コア	4 コア	8 コア	16 コア
EP	173 sec	78 sec	38 sec	19 sec
MG	134 sec	53 sec	46 sec	45 sec
CG	220 sec	150 sec	197 sec	227 sec
IS	29 sec	17 sec	16 sec	24 sec
LU	1103 sec	423 sec	286 sec	216 sec
SP	-	706 sec	-	601 sec
BT	-	257 sec	-	210 sec

linux で通常ハードディスクに使われる名前を使えばよく, euca-reboot-instances では再起動する VM を一台ずつ指定すればよい。

euca-run-instances と euca-bundle-vol の問題は運用上影響がある。特に euca-bundle-vol については Euca2ools のマイナーバージョンのレベルで現象が異なること, またイメージ化する仮想マシンの状態によってもバグの発現の有無が違ふことから大きな問題であると言える。euca-run-instances の問題は Eucalyptus 2.0 および edubase-Cloud で独自に修正した Eucalyptus 1.6.2 では起こっていないことから, 最新のバージョンでは解決されていると言える。

#### 4.2 VM の起動速度に関する考察

VM の起動速度評価で特筆すべきは, マシンイメージの全サイズに亘って Eucalyptus 1.6.2 のほうが性能がよいことである。なお, 今回の評価ではマシンイメージを NC の管理するリポジトリからネットワーク越しに CC へ転送するのではなく, CC の HDD にイメージのキャッシュが存在する状態で行った。従って起動速度にはネットワーク遅延や物理的なマシン配置の影響はない。また測定にはもちろん同じマシンイメージを使っている。

起動速度に寄与する要因の一つはマシンイメージのキャッシュからのコピーの速さ, すなわちディスク性能である。本結果の原因が Eucalyptus のバージョン差がディスク性能の差かを判断するため, 1.6.2 と 2.0 の NC (それぞれ 10.1.2.12 と 10.1.2.1) で bonnie++[12] を用いてディスク性能を評価したところ, 両者のディスク性能は同等との結論を得た (具体的な数字は割愛する)。これはクラスタ内の物理マシンが全て同じものであると

いう事実とも一致する。

従って, 今回得られた起動速度の差は Eucalyptus のバージョン差によるものだと考えられる。バージョンが高い方が起動速度が遅いため, 新たに追加されたエラーチェック等に時間がかかっている可能性と, 退行バグの可能性の二つがある。

#### 4.3 unixbench の結果に関する考察

unixbench の結果では, 全パターンに亘って Eucalyptus 1.6.2 と Eucalyptus 2.0 でほぼ同じ結果が出た。これは Eucalyptus が VM を起動した後はすべて Xen に任せ VM の実行には関与しないため, unixbench の結果は Xen の性能にのみ依存するからであると考えられる。すなわち単体 VM レベルでは Eucalyptus 1.6.2 と Eucalyptus 2.0 に性能差はない。

4 コアの VM を起動すると物理マシン一台のコアをすべて使い切るため, 仮想化のオーバーヘッドがなければ c1.xlarge での結果と物理マシンでの結果は同じになるはずである。しかし, 図 2 と図 8 を見ると VM での実行のほうが結果がよい。これは Xen による仮想化の存在, OS やカーネルのバージョンが異なることなどが関係していると考えられる。しかし Xen の中での事象であり, Eucalyptus には関係がないと考えられるので, 本稿ではこれ以上の追求はしない。

#### 4.4 bonnie++ の結果に関する考察

bonnie++ の結果を総括すると以下ようになる。

1. 全体的に Eucalyptus 1.6.2 のほうが 2.0 よりも結果がよい。
2. 2.0 のローカルディスクと EBS を比べると, バイト毎の書き込みのみ前者のほうがよい。残りはほぼ同じ。

まず 1 については不可解である。Eucalyptus のバージョンは違っても Xen のバージョンは同じであるためほぼ同じ結果を期待していた。CPU 使用率で正規化すれば同様の値になるが, ディスク IO は CPU を多く使えばそれだけ速くなる性質のものではないため, 正規化して比べることが妥当なのかどうか不明である。

2 については, iSCSI のレイヤーでバッファリングと複数書き込み命令のマージがされていると考えられる。オーバーヘッドはハードディスクの seek 及び wait 時間だと考えられるため, iSCSI 以下 (TCP や IP) のレベルのバッファリングには意味がない。

#### 4.5 NPB の結果に関する考察

NPB の結果で Eucalyptus 1.6.2 と 2.0 で差異があったのは MG と CG である。

MG の m1.small での結果についてはメモリ不足によるスワップアウトが発生しているため, 正常な結果ではない。従って MG の差異を議論することは無意味である。

CG では、Eucalyptus 2.0 では VM の数を増やす（合計コア数は同じであるので 1VM あたりのコア数を減らす）につれて明らかに結果がよくなっている。一方 Eucalyptus 1.6.2 ではそのような傾向は見られない。4.3 節で述べたように単体 VM レベルでは性能差は見られないため、VM の並列にかかわる性能が Eucalyptus 2.0 で向上したことが原因である可能性がある。例えば、VLAN に関する処理性能の向上によりネットワーク遅延が減った可能性などが考えられる。

まとめると、NPB による評価において、CG 以外のベンチマークでは Eucalyptus 1.6.2 と 2.0 は同等の性能を示した。前述のネットワーク遅延の低減については調査する価値があるが、本稿では CG ベンチマークにおいて Eucalyptus 2.0 の方がよい結果を示したと述べるに留める。

## 5. 結論

本稿では Eucalyptus 1.6.2 と Eucalyptus 2.0 にそれぞれ 5 台ずつのマシクラスタを割り当て、その機能評価と非機能評価を行った。

機能評価では、Eucalyptus を操作する Euca2ools で提供されている機能のいくつかに運用上問題となりうる大きな問題が見つかった。特に euca-bundle-vol については、起動中の仮想マシンのスナップショットを取るといった用途で多用されるため重大な問題であると言える。非機能評価では、以下の 2 点に差が出た。

1. 仮想マシンの起動速度
2. 仮想マシンのローカルディスクへの IO 性能
3. NPB のうちの CG ベンチマーク

仮想マシンの起動速度は Eucalyptus 1.6.2 の方が結果がよかった。NC のディスク性能は同等であること、今回の評価方法ではネットワーク性能は関係ないことから原因はハードウェア性能ではなくソフトウェア性能である。Eucalyptus 2.0 に何らかの Eucalyptus 1.6.2 にはないエラーチェックや Verify が追加されているか、もしくは Eucalyptus 2.0 の退行バグである可能性がある。

IO 性能については、Eucalyptus 1.6.2 の方が結果がよかった。しかしこれについては非常に不可解でありさらなる調査が必要である。

CG ベンチマークについては Eucalyptus 2.0 の方が結果がよかった。単体の仮想マシンの性能は Eucalyptus 1.6.2 と 2.0 で同等であったため、VLAN によるネットワーク遅延などが改善されて連携性能が上がったと考えられる。

上記以外の点では二つのバージョン間に性能差は見られなかった。結論としては、

Eucalyptus 1.6.2 と Eucalyptus 2.0 には細かな性能の差異はあるものの、全体としてはほぼ同等の性能を示す。ただし 2.0 では運用上問題となる仮想マシンが起動しないことがあるというバグが修正されている。

と言える。

## 付録 A 成果物

本評価における成果物は以下の通り。

1. Eucalyptus 1.6.2 と Eucalyptus 2.0 の比較結果 (本報告書)
2. インスタンス起動時間計測スクリプト
3. NAS Parallel Benchmarks の手順書
4. NAS Parallel Benchmarks をインストールして実行可能にするスクリプト

## 参考文献

- [1] Eucalyptus, <http://open.eucalyptus.com/>
- [2] Installing Eucalyptus, eucalyptus documentation, [http://open.eucalyptus.com/wiki/EucalyptusInstallation\\_v2.0](http://open.eucalyptus.com/wiki/EucalyptusInstallation_v2.0)
- [3] Eucalyptus 構築セミナー, [http://www.creationonline.com/image/Eucalyptus\\_seminar090806.pdf](http://www.creationonline.com/image/Eucalyptus_seminar090806.pdf), 2009
- [4] Euca2ools User Guide, eucalyptus documentation, [http://open.eucalyptus.com/wiki/Euca2oolsGuide\\_v1.3](http://open.eucalyptus.com/wiki/Euca2oolsGuide_v1.3)
- [5] Image Creator's Guide, eucalyptus documentation, [http://open.eucalyptus.com/wiki/EucalyptusUserImageCreatorGuide\\_v2.0](http://open.eucalyptus.com/wiki/EucalyptusUserImageCreatorGuide_v2.0)
- [6] byte-unixbench A Unix benchmark suite, <http://code.google.com/p/byte-unixbench/>
- [7] Andri D. Balsa 著, 岡本 一幸 訳, Linux ベンチマーク HOWTO, <http://archive.linux.or.jp/JF/JFdocs/Benchmarking-HOWTO.html>, 1997, 3.4 節
- [8] NAS PARALLEL BENCHMARKS, <http://www.nas.nasa.gov/Resources/Software/npb.html>
- [9] [http://mikilab.doshisha.ac.jp/dia/smpp/01\\_bench/naspara.html](http://mikilab.doshisha.ac.jp/dia/smpp/01_bench/naspara.html)
- [10] Open MPI: Open Source High Performance Computing, <http://www.open-mpi.org/>
- [11] <http://open.eucalyptus.com/forum/eucalyptus-20-centos-any-created-image-will-not-boot>
- [12] <http://www.coker.com.au/bonnie++/>
- [13] Manpage of LSEEK, [http://archive.linux.or.jp/JM/html/LDP\\_man-pages/man2/lseek.2.html](http://archive.linux.or.jp/JM/html/LDP_man-pages/man2/lseek.2.html)
- [14] Bonnie++ Documentation, <http://www.coker.com.au/bonnie++/readme.html>

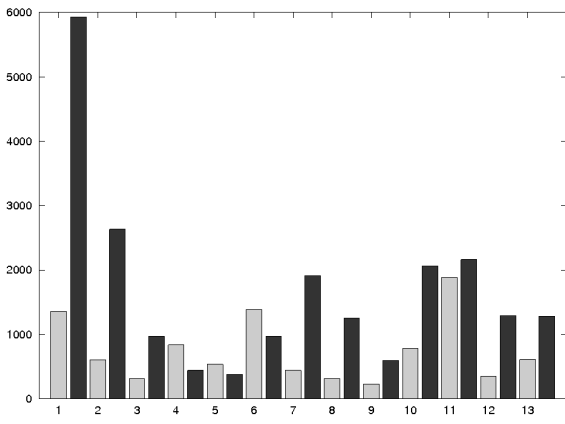


图 2: 4 cores, Eucalyptus 2.0

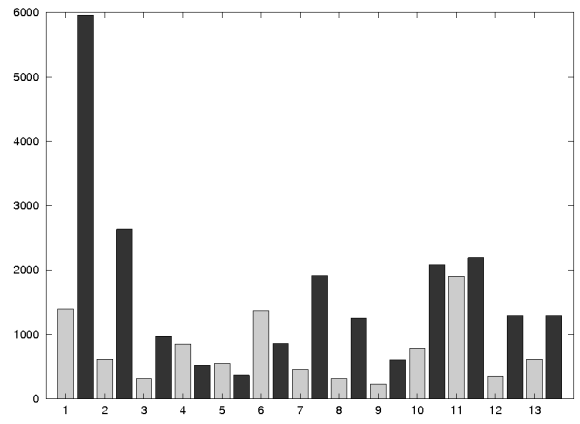


图 3: 4 cores, Eucalyptus 1.6

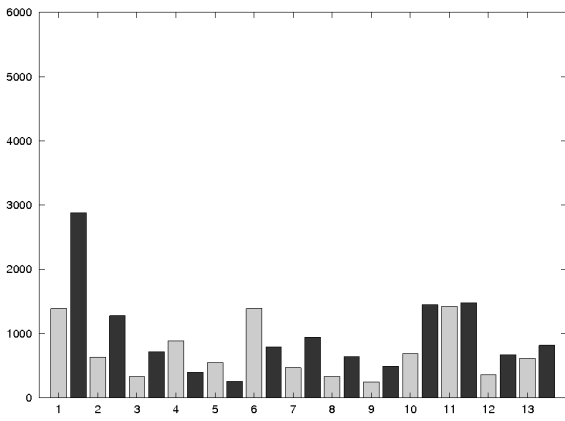


图 4: 2 cores, Eucalyptus 2.0

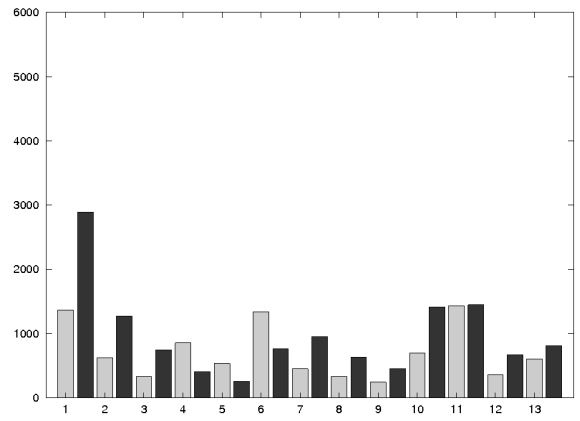


图 5: 2 cores, Eucalyptus 1.6

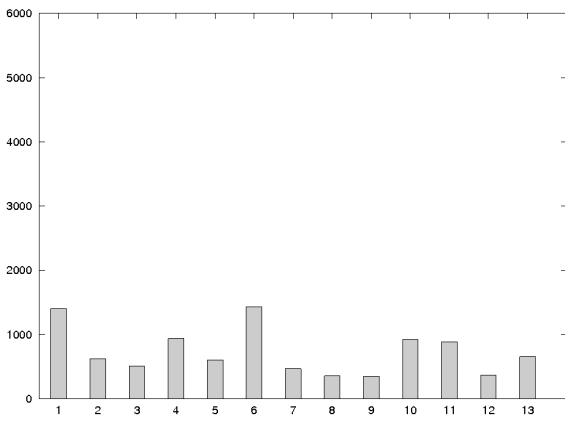


图 6: 1 core, Eucalyptus 2.0

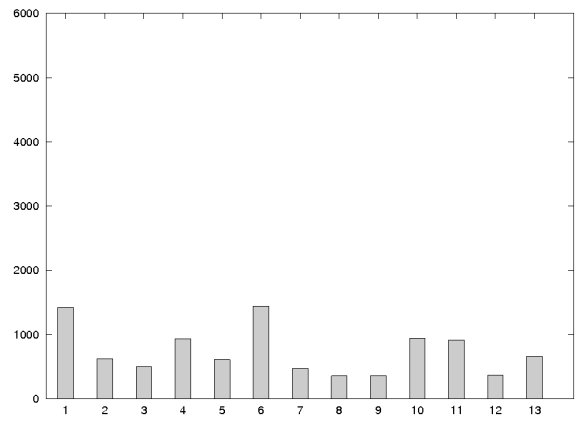


图 7: 1 core, Eucalyptus 1.6

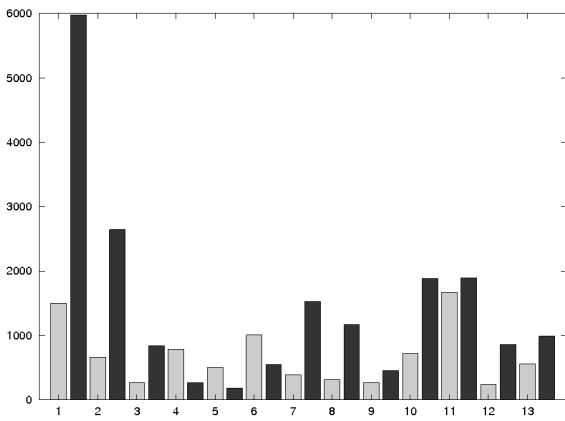


图 8: 4 cores, Node Controller