WIDE Technical-Report in 2012

大規模ストレージソフトウェアの評価

wide-tr-cloud-distributed-storage-measurement-00.pdf

WIDE Project : http://www.wide.ad.jp/

If you have any comments on WIDE documents, please contact to board@wide.ad.jp

Title: 大規模ストレージソフトウェアの評価
Author(s): 島 慶一, 関谷 勇司, 宮本 大輔
Date: 2012-2-16

大規模ストレージソフトウェアの評価

島 慶一* 関谷 勇司† 宮本 大輔†

2012 年 2 月 16 日

概 要

IaaS サービスなどを大規模広域ネットワーク環境で実現するために必要となる、大規模広域分散ストレージとその仮想環境での運用実現性を検証するために、sheepdog と GlusterFS を対象とした性能計測を実施した。実験の結果、sheepdog はディスクアクセス性能が GlusterFS より安定しているが、全体としての性能は低い事や遅延に弱いことなどが確認された。今後より詳細な計測を進め、広域環境に的した分散ストレージの研究に繋げていきたい。

1 評価の目的

IaaS サービスなどを大規模広域ネットワーク環境で実現するために必要となる、大規模広域分散ストレージとその仮想環境での運用実現性を検証する。現在広く利用されている仮想環境 KVM を基礎環境として、その上で分散ストレージシステム sheepdog と、GlusterFS を、最大 128 台のノードを用いて運用性を検証し、さらに広域ネットワークのエミュレーションを用いて広域環境での運用性を検証する。

2 機器の仕様

検証には StarBED で提供されている Cisco UCS C200 M2 を用いた。表 1 に UCS の仕様を示す。

表 1: 検証に利用した計算機の仕様

部品	仕様
CPU	Intel(R) Xeon(R) X5670@2.93GHz × 2
Cores/CPU	6
Memory	48G
Disk	SATA 500GB × 2
NIC (0-3)	Broadcom NetXtreme II BCM5709 1000Base-T
NIC (4-5)	Intel(R) Gigabit Ethernet Network

NIC0 から NIC3 は Brocade MLXe-32 スイッチによって相互に接続されている。また NIC4 は管理用セグメントに接続されている。NIC5 は本来未使用であるが、今回 sheepdog 検証のためのストレージエリアネットワークとして利用した (3.1 章参照)。

*株式会社 IIJ イノベーションインスティテュート
†東京大学

3 sheepdog 検証

3.1 ネットワークトポロジ

図 1 に検証に用いたトポロジを示す。

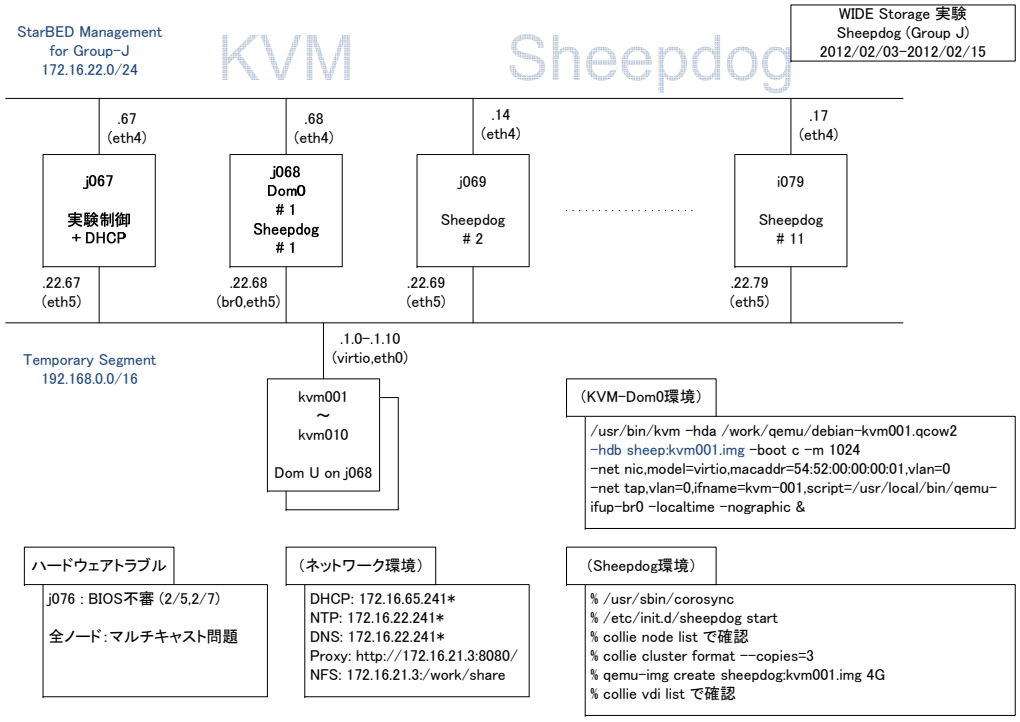

図 1: sheepdog 検証ネットワークトポロジ

sheepdog の検証には、グループ J と呼ばれる Cisco UCS から構成された物理計算機 11 台を用いた。j067 を制御用として利用し、j068 から j079(BIOS 不良が発生していた j076 を除く) を sheepdog ストレージを構成するクラスタノード群として利用した。また、j068 は KVM ハイパーバイザーとしても動作し、10 台の仮想計算機をホストしている。仮想計算機は j068 が提供する sheepdog ストレージを仮想ローカルディスクとして利用した。制御用、およびクラスタ構成機材では Ubuntu 11.10 (64bit)、仮想計算機では Debian6.0 (32bit) オペレーティングシステムを用いた。

本来ならば、NIC0 から NIC3(eth0 から eth3 に対応) を実験用として sheepdog クラスタ間の通信に利用するところであるが、Broadcom NetXtreme II BCM5709 のネットワークアダプタドライバのマルチキャストパケット処理の不具合と思われる現象が発生したため、Intel(R) Gigabit Ethernet で構成されている NIC5(eth5) をストレージネットワーク用インターフェースとして利用した。NIC5 の相互接続には D-Link DGS-3450 を用いた。

3.2 ソフトウェア構成

検証には Ubuntu 11.10 の APT パッケージシステムとして提供されているバージョンを用いた。表 2 に関連ソフトウェアの情報を示す。また、付録 A に sheepdog で利用されるクラスタドライバ corosync の設定ファイル例を示す。

sheepdog 用として、Cisco UCS に搭載されている 2 台の 500GB ハードディスクの一方を用いた。sheepdog はストレージクラスタ構成時に冗長化のための複製数を指定できるようになっ

表 2: sheepdog 検証に利用したソフトウェアバージョン

パッケージ名	APT でのバージョン
sheepdog	0.2.3-0ubuntu1
corosync	1.3.0-3ubuntu1
libcorosync4	1.3.0-3ubuntu1

ているため、今回は複製数を 3 として構成している。結果、500GB×11 台を 3 で割ったおよそ 1.7TB がストレージの容量となる。

3.3 検証手順

sheepdog の検証には bonnie++を用いた。j068 上で稼働している仮想計算機で bonnie++を動作させ、表 3 に示す場合のディスク性能を計測した。

表 3: sheepdog 検証ケース

ケース	仮想計算機台数	遅延
1	10	なし
2	10	50ms

計測ケースは 2 ケースで、ケース 1 は sheepdog クラスタノード間 (j068 から j079、j076 を除く) の遅延なしで 10 台の仮想計算機による同時ディスクアクセス、ケース 2 は、クラスタノードのネットワークインターフェース出力方向に 50ms の遅延を入れた状態で、同様に 10 台の仮想計算機による同時ディスクアクセスとなっている。

3.4 結果の考察

表 4、5 にケース 1、2 の結果を示す。各仮想計算機の結果の値は、bonnie++を 5 回実行した平均値を使用している。50ms の遅延が入った場合、書き込み性能、特にブロック書き込み性能が著しく低下している (ブロック書き込みは 3.8%まで性能が下がっている) のがわかる。それに対して、読み込み性能は、キャラクタ単位で 79%、ブロック単位では 98%の性能が維持できていることがわかる

表 4: bonne++の結果 (sheepdog、ケース 1 (10VM, 遅延なし))

仮想計算機	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
kvm1	571.6	3175.4	2809.4	2228.4	14669.6
kvm2	570.0	3174.8	2823.0	2166.8	13758.8
kvm3	584.4	3153.8	2802.8	1956.2	13875.8
kvm4	588.0	3159.0	2818.8	2203.0	13912.0
kvm5	554.6	3153.8	2820.0	2206.2	13854.0
kvm6	565.8	3164.8	2826.6	2168.8	14225.8
kvm7	562.6	3134.4	2807.4	1665.4	14355.8
kvm8	595.8	3155.8	2816.8	2053.8	13640.0
kvm9	585.6	3157.2	2807.2	2085.0	14336.6
kvm10	590.4	3173.4	2814.2	2162.2	14051.2
MIN	554.6	3134.4	2802.8	1665.4	13640.0
MAX	595.8	3175.4	2826.6	2228.4	14669.6
AVERAGE	576.88	3160.24	2814.62	2089.58	14067.96
STDEV	13.79	12.547	7.7404	170.70	321.099

表 5: bonne++の結果 (sheepdog、ケース 2 (10VM, 50ms 遅延))

仮想計算機	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
kvm1	175.0	333.4	154.0	201.6	384.2
kvm2	180.8	335.2	153.0	201.6	382.8
kvm3	185.8	335.6	150.4	201.6	366.8
kvm4	169.4	334.0	156.0	197.4	388.8
kvm5	183.8	335.0	152.4	296.0	384.8
kvm6	182.6	335.4	150.0	201.0	360.4
kvm7	185.0	333.8	153.4	686.6	389.2
kvm8	177.6	335.4	151.8	205.0	375.8
kvm9	182.6	334.6	154.0	294.6	394.0
kvm10	169.0	332.4	156.2	201.0	403.4
MIN	169.0	332.4	150.0	197.4	360.4
MAX	185.8	335.6	156.2	686.6	403.4
AVERAGE	179.16	334.48	153.12	268.64	383.02
STDEV	6.185	1.050	2.076	151.9	12.64

表 6: 遅延が入った場合の性能低下 (遅延/無遅延)

	put chr	put blk	rewrite blk	get chr	get blk
MIN	0.433	0.122	0.109	0.827	1.031
MAX	0.427	0.019	0.022	0.766	1.004
AVERAGE	0.421	0.038	0.044	0.798	0.983

4 GlusterFS 検証

4.1 ネットワークトポロジ

図 1 に検証に用いたトポロジを示す。

GlusterFS の検証には、グループ I と呼ばれる Cisco UCS から構成された物理計算機 128 台を用いた。i003 を制御用として利用し、i004 から i174(連番ではないため、詳細な構成は付録 B を参照) の 128 台を GlusterFS ストレージを構成するクラスタノード群として利用した。また、i004 から i014(ただし、動作不良を起こしていた i012 を除く) の 10 台は KVM ハイパーバイザーとしても動作し、それぞれが 10 台の仮想計算機をホストしている。仮想計算機はハイパーバイザーが提供する GlusterFS ストレージを仮想ローカルディスクとして利用した。制御用、およびクラスタ構成機材では Ubuntu 11.10 (64bit)、仮想計算機では Debian6.0 (32bit) オペレーティングシステムを用いた。

本来ならば、NIC0 から NIC3(eth0 から eth3 に対応) を実験用として GlusterFS クラスタ間の通信に利用するところであるが、NIC0 から NIC3 の上流に配置してある Brocade MLXe-32 で通信障害が発生しており、多数のパケットロス、およびラインカード間の通信不良が観測されたため、マネージメントセグメント NIC4(eth4) をストレージネットワーク用インターフェースとして利用した。NIC4 の相互接続には D-Link DGS-3450 を用いた。

4.2 ソフトウェア構成

検証には Ubuntu 11.10 の APT パッケージシステムとして提供されているバージョンを用いた。表 7 に関連ソフトウェアの情報を示す。

GlusterFS 用として、Cisco UCS に搭載されている 2 台の 500GB ハードディスクの一方を用いた。GlusterFS は、冗長構成を持たない「Distributed」型、冗長構成を持つ「Replica」型を利用することができるため、比較のためにこの 2 種類を構成した。なお、性能向上のための「Stripe」型も提供されているが、検証時点で動作が安定しないなかったため、今回検証対象には入れていない。

Distributed 型では、500GB×127 台でおよそ 58TB、Replica 型は、今回複製数を 3 に指定したため、その 3 分の 1 のおよそ 15TB を利用できる。

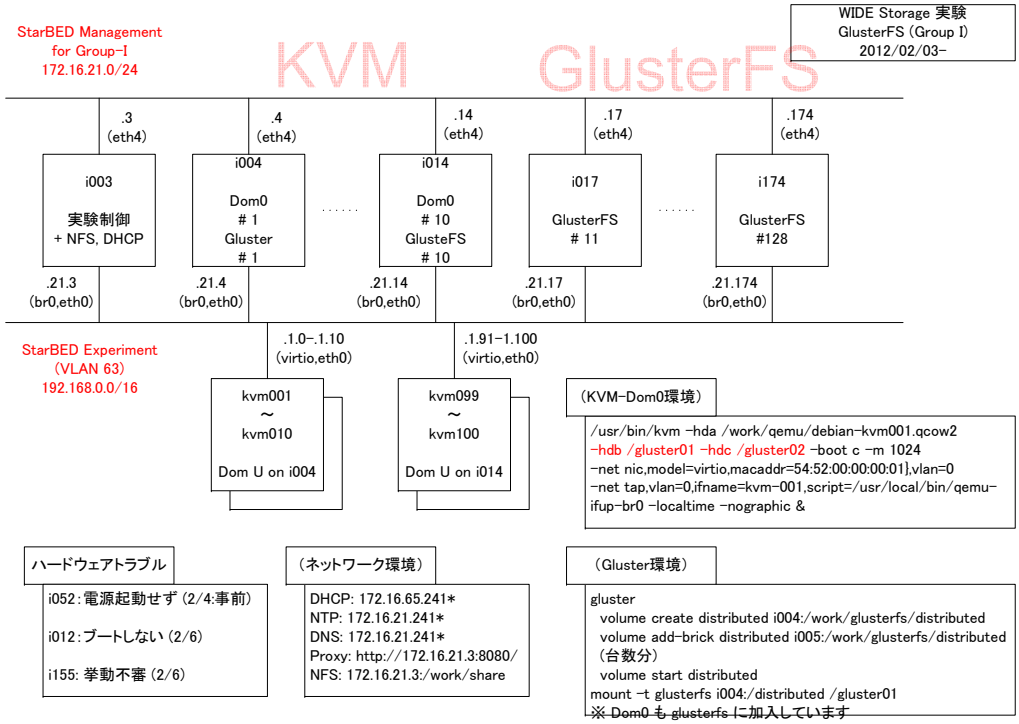

図 2: GlusterFS 検証ネットワークトポロジ

表 7: GlusterFS 検証に利用したソフトウェアバージョン

パッケージ名	APT でのバージョン
glusterfs-client	3.2.1-1
glusterfs-common	3.2.1-1
glusterfs-examples	3.2.1-1
glusterfs-server	3.2.1-1

4.3 検証手順

GlusterFS の検証には bonnie++を用いた。i004 から i014 上で稼働している仮想計算機で bonnie++を動作させ、表 8 に示す場合のディスク性能を計測した。

計測ケースは 6 ケースで、ケース 1、3、5 は GlusterFS クラスタノード間の遅延なし、ケース 2、4、6 は出力方向に 50ms の遅延が入っている。ケース 1 と 2、3 と 4、5 と 6 はそれぞれ、1 台、10 台、50 台、100 台の仮想計算機からの同時ディスクアクセスによる計測となっている。このケース 1 から 6 を Distributed 型と Replica 型それぞれで計測した。

4.4 結果の考察

表 9、10 に Distributed 型、ケース 1 と 2 の結果を示す。各仮想計算機の結果の値は、bonnie++を 5 回実行した平均値を使用している。また、表 11、12、13、14 にケース 3 から 6 の結果を示す。データ量が多いため、最小値、最大値、平均値、標準偏差のみを掲載する。

遅延を入れた場合、書き込み性能が落ちている事がわかる。また、遅延を入れた状態で 100 台の仮想計算機から同時アクセスした場合、読み込み性能のばらつきが大きくなっている。仮想

表 8: GlusterFS 検証ケース

ケース	仮想計算機台数	遅延
1	10	なし
2	10	50ms
3	50	なし
4	50	50ms
5	100	なし
6	100	50ms

計算機の数が増える事で、同じクラスタノードを利用している仮想計算機からの同時アクセスによる輻輳が原因と思われるが、正確な理由は今後の調査が必要である。

表 9: bonne++の結果 (GlusterFS、Distributed 型、ケース 1 (10VM、遅延なし))

仮想計算機	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
kvm1	772.4	72777.0	62407.4	3098.0	699825
kvm11	708.8	9733.6	12365.8	2888.0	661311
kvm21	753.2	55373.4	53991.0	2901.4	660159
kvm31	685.0	9713.8	12361.2	2751.2	619964
kvm41	669.2	9568.2	12364.0	2909.6	576281
kvm51	776.2	52278.2	54154.8	2845.2	612073
kvm61	668.0	8629.0	11317.4	2812.8	685477
kvm71	664.2	9712.6	12345.6	2716.4	549896
kvm81	693.2	9750.6	12440.8	2804.2	631052
kvm91	761.6	53111.0	52793.2	2840.2	643199
MIN	664.2	8629.0	11317.4	2716.4	549896
MAX	776.2	72777.0	62407.4	3098.0	699825
AVERAGE	715.18	29064.74	29654.12	2856.7	633923.7
STDEV	45.91	25848.7	22680.7	105.35	46653.5

表 15、16 に Replica 型、ケース 1 と 2 の結果を示す。各仮想計算機の結果の値は、bonnie++ を 5 回実行した平均値を使用している。Distributed 型の同ケースと比較すると、複製が必要な分、書き込みの性能が落ちる傾向が見られる。他のケースも、おおむね Distributed 型と同じ傾向がみられる。遅延を入れた場合の 100 台の仮想計算機からの同時アクセスで読み込み性能のばらつきが出ている現象は、Replica 型でも再現していた。

5 考察

sheepdog と GlusterFS を比較すると、同じ環境では GlusterFS の方が高い性能を示す。例えば、表 4 と表 15 において、sheepdog のブロック書き込み性能は最大値でも 3175.4K/sec なのに対して、Replica 型の GlusterFS のブロック書き込み性能は最大値で 26742.2K/sec と 8.4 倍近い差がある。読み込みに関しても同様の傾向があり、sheepdog の 14669.6K/sec と GlusterFS の 633579K/sec と、43 倍もの差がある。

これは、sheepdog はディスクイメージをブロック単位で分割、複製するのに対し、GlusterFS はひとつのディスクイメージが 1 ファイルになっているためと思われる。sheepdog の場合、ひとつの仮想計算機のディスクにアクセスするために、多数の sheepdog クラスタノードとの通信が必要なのに対し、GlusterFS の場合、Distributed 型ではひとつ、Replica 型では複製数と同じ数のクラスタノードにアクセスするだけですむ。

一方、仮想計算機ごとのディスクアクセス性能に関しては、sheepdog の方がゆらぎが少ない。GlusterFS の場合、仮想計算機が稼働しているハイパーバイザーと、対応する仮想ディスクが格納されている GlusterFS のノードの組み合わせによって、高い性能がでる場合と性能がでない場合が明確に分かれる。特にブロック読み書き時の性能のばらつきが大きい。

表 10: bonne++の結果 (GlusterFS、Distributed 型、ケース 2 (10VM、50ms 遅延))

仮想計算機	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
kvm1	331.8	1397.2	1391.8	2291.6	567084
kvm11	296.8	1108.2	1337.0	2252.0	610029
kvm21	311.0	1083.2	1239.6	2332.8	685430
kvm31	301.0	1106.6	1329.4	2246.6	617564
kvm41	289.2	1093.4	1309.8	2268.8	585167
kvm51	292.8	1053.4	1255.4	2373.2	598867
kvm61	305.6	1096.2	1326.0	2256.6	702741
kvm71	288.0	1103.2	1332.6	2281.4	597652
kvm81	297.4	1098.6	1329.4	2246.6	620693
kvm91	303.4	1058.4	1257.2	2267.2	650562
MIN	288.0	1053.4	1239.6	2246.6	567084
MAX	331.8	1397.2	1391.8	2373.2	702741
AVERAGE	301.70	1119.84	1310.82	2281.68	623578.9
STDEV	12.80	99.307	46.735	41.411	43459.3

表 11: bonne++の結果 (GlusterFS、Distributed 型、ケース 3 (50VM、遅延なし))

	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
MIN	669.6	6547.0	6912.4	2604.2	565291
MAX	781.2	51083.8	47192.0	3013.2	700060
AVERAGE	724.06	21224.36	20320.90	2783.66	646068.8
STDEV	21.83	13721.5	12542.9	97.762	34265.2

謝辞

本検証作業にあたり、北陸リサーチセンターの三輪信介所長、宮地利幸博士、および中井浩氏には、実験機材の手配および構成に関して有用な助言をいただき、また作業環境の構築に尽力していただきました。ありがとうございました。

A corosync 設定ファイル

```
# Please read the openais.conf.5 manual page

totem {
  version: 2

  # How long before declaring a token lost (ms)
  token: 3000

  # How many token retransmits before forming a new configuration
  token_retransmits_before_loss_const: 10

  # How long to wait for join messages in the membership protocol (ms)
  join: 60

  # How long to wait for consensus to be achieved before starting a new round of membership configurati
  consensus: 3600

  # Turn off the virtual synchrony filter
  vsftype: none
```

表 12: bonne++の結果 (GlusterFS、Distributed 型、ケース 4 (50VM、50ms 遅延))

	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
MIN	289.6	1001.4	1129.6	2214.0	528388
MAX	382.4	1505.2	1495.4	2369.4	706130
AVERAGE	309.95	1144.49	1282.82	2278.57	619416.8
STDEV	20.12	142.69	88.901	35.484	35595.4

表 13: bonne++の結果 (GlusterFS、Distributed 型、ケース 5 (100VM、遅延なし)

	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
MIN	506.6	4398.4	4472.0	2248.6	229637
MAX	751.8	58965.2	56140.6	2980.8	700719
AVERAGE	682.77	15235.99	14820.16	2707.59	539161.8
STDEV	42.95	8812.6	8124.35	115.45	113093

```
  # Number of messages that may be sent by one processor on receipt of the token
  max_messages: 20

  # Limit generated nodeids to 31-bits (positive signed integers)
  clear_node_high_bit: yes

  # Disable encryption
  secauth: off

  # How many threads to use for encryption/decryption
  threads: 0

  # Optionally assign a fixed node id (integer)
  # nodeid: 1234

  # This specifies the mode of redundant ring, which may be none, active, or passive.
  rrp_mode: none

  interface {
    # The following values need to be set based on your environment
    ringnumber: 0
    bindnetaddr: 192.168.22.68
    mcastaddr: 226.94.1.1
    mcastport: 5405
  }
}

amf {
  mode: disabled
}

service {
  # Load the Pacemaker Cluster Resource Manager
  ver: 0
  name: pacemaker
}

aisexec {
  user:   root
```

表 14: bonne++の結果 (GlusterFS、Distributed 型、ケース 6 (100VM、50ms 遅延))

	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
MIN	252.8	772.4	403.2	405.2	1032.8
MAX	766.2	57513.4	60969.4	2910.8	609411
AVERAGE	320.40	1740.47	1218.01	1113.04	64095.28
STDEV	54.17	5637.1	6037.7	538.55	107610.3

表 15: bonne++の結果 (GlusterFS、Replica 型、ケース 1 (10VM、遅延なし))

仮想計算機	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
kvm1	606.2	5574.0	5455.4	2799.8	623009
kvm11	616.0	5058.8	6456.6	2746.4	572266
kvm21	591.0	4698.2	5974.2	2725.6	588609
kvm31	607.2	5139.0	6615.6	2739.4	558716
kvm41	729.0	26624.4	29706.2	3040.8	581283
kvm51	734.0	26742.2	29859.8	2828.2	557647
kvm61	609.4	5627.2	7146.2	2781.4	633579
kvm71	601.2	5347.0	6684.8	2803.6	567401
kvm81	640.2	5531.2	6990.6	2716.0	611544
kvm91	634.6	5244.2	6670.6	2883.4	585852
MIN	591.0	4698.2	5455.4	2716.0	557647
MAX	734.0	26742.2	29859.8	3040.8	633579
AVERAGE	636.88	9558.62	11156.0	2806.46	587990.6
STDEV	51.98	9029.7	9829.25	97.076	26579.5

```
  group:  root
}

logging {
  fileline: off
  to_stderr: yes
  to_logfile: no
  to_syslog: yes
  syslog_facility: daemon
  debug: off
  timestamp: on
  logger_subsys {
    subsys: AMF
    debug: off
    tags: enter|leave|trace1|trace2|trace3|trace4|trace6
  }
}
```

表 16: bonne++の結果 (GlusterFS、Replica 型、ケース 2 (10VM、50ms 遅延))

仮想計算機	put chr (K/sec)	put blk (K/sec)	rewrite blk (K/sec)	get chr (K/sec)	get blk (K/sec)
kvm1	301.4	1104	1087.2	2279.4	569349
kvm11	258.8	797.2	990.8	2178.2	585465
kvm21	262.4	794.6	988.2	2212.2	571915
kvm31	261.2	805.8	989.2	2240.0	531693
kvm41	256.2	805.6	997.4	2325.8	587171
kvm51	252.0	805.8	1001.4	2314.6	567483
kvm61	265.6	815.0	999.2	2278.6	541098
kvm71	251.0	791.6	993.2	2195.8	540323
kvm81	250.8	791.8	983.4	2306.0	621533
kvm91	258.4	808.8	989.8	2320.0	543093
MIN	250.8	791.6	983.4	2178.2	531693
MAX	301.4	1104.0	1087.2	2325.8	621533
AVERAGE	261.78	832.02	1001.98	2265.06	565912.3
STDEV	14.79	95.88	30.44	54.762	27769.7

B クラスタノードリスト

B.1 sheepdog クラスタノードリスト

j068, j069, j070, j071, j072, j073, j074, j075, j077, j078, j079

B.2 GlusterFS クラスタノードリスト

i004, i005, i006, i007, i008, i009, i010, i011, i013, i014, i019, i020, i021, i022, i023, i024, i025, i026, i027, i028, i029, i030, i035, i036, i037, i038, i039, i040, i041, i042, i043, i044, i045, i046, i051, i053, i054, i055, i056, i057, i058, i059, i060, i061, i062, i067, i068, i069, i070, i071, i072, i073, i074, i075, i076, i077, i078, i083, i084, i085, i086, i087, i088, i089, i090, i091, i092, i093, i094, i099, i100, i101, i102, i103, i104, i105, i106, i107, i108, i109, i110, i115, i116, i117, i118, i119, i120, i121, i122, i123, i124, i125, i126, i131, i132, i133, i134, i135, i136, i137, i138, i139, i140, i141, i142, i147, i148, i149, i150, i151, i152, i153, i154, i156, i157, i158, i163, i164, i165, i166, i167, i168, i169, i170, i171, i172, i173, i174