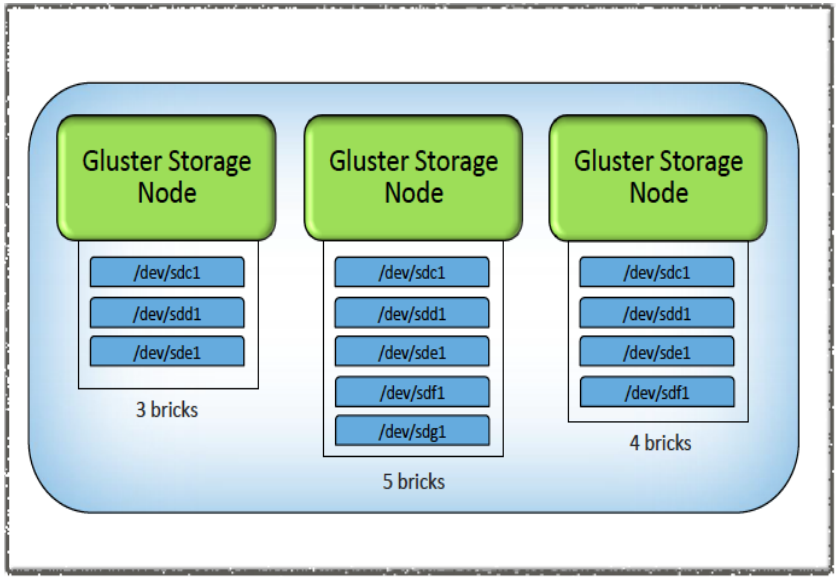
GlusterFSのご紹介

コアマイクロシステムズ株式会社

2011/08/31

1. 構成要素

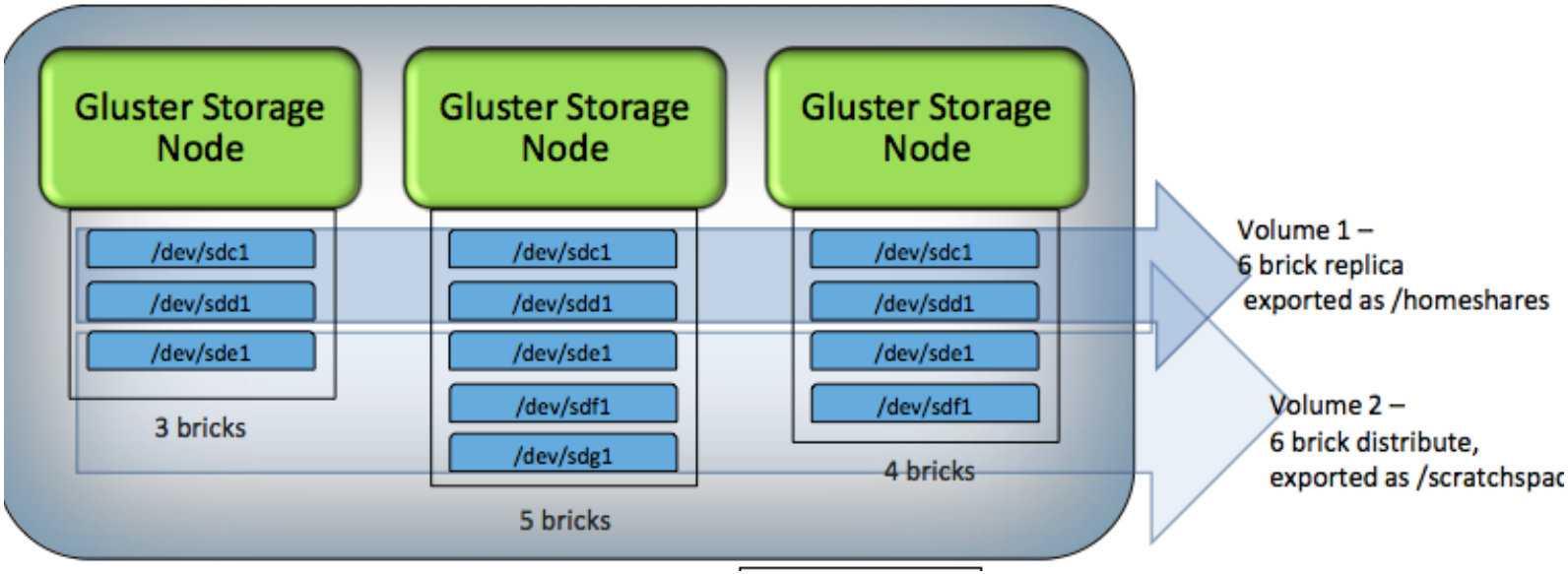
- ノード
  - CPU + HDD
  - Peerの単位
  - Peer同士は名前空間を共有する
- ブリック (Brick)
  - ノード上のファイルシステム
    - OSレベルのFile System
  - HDDないしRAIDボリュームが一般的
  - ファイルシステムの制約を受ける
    - 例えばext3なら最大容量が8TB



GLUSTER
The Fastest Path to
Public and Private Cloud Storage

2. Elastic Volumes

- 複数のBrickを論理的にまとめて、1つの名前空間を提供する
- Replica (ミラー)を組み合わせて、Distribution (ストライプ)の組み合わせ
- 単一のネームスペース → マウントする単位
- Volumeを構成するBrickは動的に追加可能



Replica

- Brick単位での冗長化
  - N-wayが可能
- 書き込みは瞬時に同期
- 読み出しは任意のノードから
  - 冗長度を上げると、読み出しパフォーマンスも向上

Distribution

- Brickの連結 (ストライピング)
- Replicaの単位で増設可能
  - N-wayならN個単位
- グループの縮小は今のところ不可
  - 開発計画あり

4. Elastic Hash AlgorithmとRebalanceの概念

Elastic Hash Algorithm

- ディレクトリ名とファイル名からハッシュ値を算出
- ハッシュ値の範囲でファイルが置かれるBrickが決まる
- Brickが増設された場合にはマッピングが自動的に変化する
  - 既存ファイルはrebalanceするまで移動されない
  - 新規作成ファイルは新しい位置に置かれる
- メタデータ用コントローラが存在しない
  - →単一障害点が無い

Rebalance

- 新しいHash Graphに基づいて、既存のファイルを移動する
- Brickの追加を行った場合に行う
- 移動する容量に応じて時間がかかる
- バックグラウンド処理

Gluster Native

- FUSEで実装 (Server/Client共)
  - クライアントのCPU負荷が高い
- ファイルを持っているノードと、ファイルを直接やりとりする
- Replicaが1つ以上生きていればアクセス可能 → デモ
- Linuxのみ

NFS

- Gluster Server内に実装されている
  - パフォーマンスはLinux Nativeとほぼ同等 (write時)
- アクセスノードがダウンしたら、そこを参照しているクライアントはアクセス不能
  - CTDBなどで回避可能 (後述)

CIFS

- Native ClientとSambaを組み合わせる
- アクセスノードがダウンしたら、そこを参照しているクライアントはアクセス不能
  - CTDBなどで回避可能 (後述)

6. 制御コマンド (CLI)

- Glusterコマンド1つのみ
- peerサブコマンド
  - ピアリングを制御 (追加・削除)
  - 1ノード = 1 IPの制約
    → アクセスセグメントとレプリケーションセグメントを分離できない
- volumeサブコマンド
  - ボリュームの作成・削除
  - サービスの制御
  - ボリュームを構成するBrickの追加・削除・置換
  - Rebalance、同期など

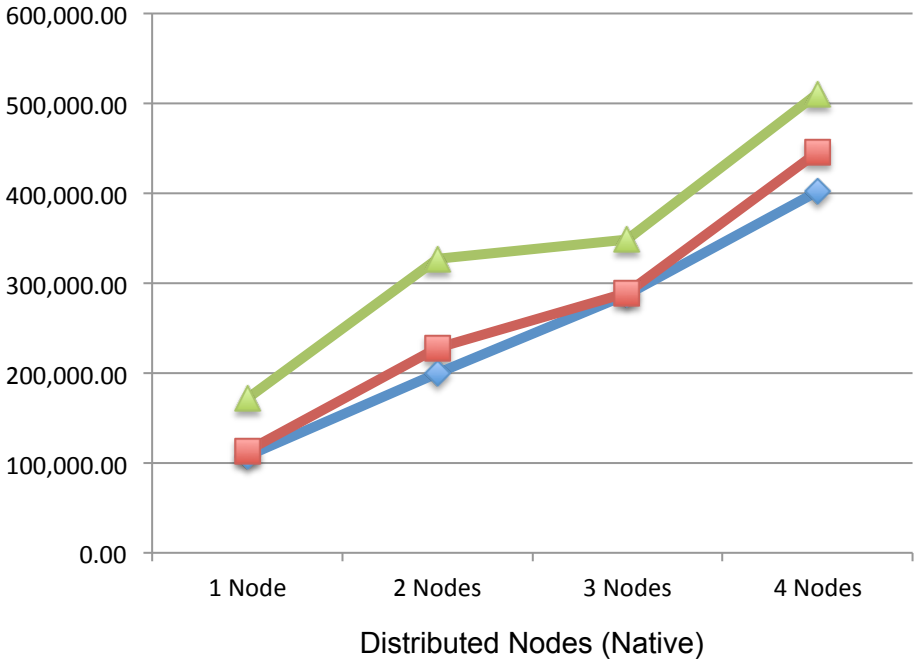
```
# gluster help
volume info [all|<VOLNAME>] - list information of all volumes
volume create <NEW-VOLNAME> [stripe <COUNT>] [replica <COUNT>] [transport <tcp|rdma|
tcp,rdma>] <NEW-BRICK> ... - create a new volume of specified type with mentioned bricks
volume delete <VOLNAME> - delete volume specified by <VOLNAME>
volume start <VOLNAME> [force] - start volume specified by <VOLNAME>
volume stop <VOLNAME> [force] - stop volume specified by <VOLNAME>
volume add-brick <VOLNAME> <NEW-BRICK> ... - add brick to volume <VOLNAME>
volume remove-brick <VOLNAME> <BRICK> ... - remove brick from volume <VOLNAME>
volume rebalance <VOLNAME> [fix-layout|migrate-data] {start|stop|status} - rebalance operations
volume replace-brick <VOLNAME> <BRICK> <NEW-BRICK> {start|pause|abort|status|commit} -
replace-brick operations
volume set <VOLNAME> <KEY> <VALUE> - set options for volume <VOLNAME>
volume help - display help for the volume command
volume log filename <VOLNAME> [BRICK] <PATH> - set the log file for corresponding volume/
brick
volume log locate <VOLNAME> [BRICK] - locate the log file for corresponding volume/brick
volume log rotate <VOLNAME> [BRICK] - rotate the log file for corresponding volume/brick
volume sync <HOSTNAME> [all|<VOLNAME>] - sync the volume information from a peer
volume reset <VOLNAME> [force] - reset all the reconfigured options
volume profile <VOLNAME> {start|info|stop} - volume profile operations
volume quota <VOLNAME> <enable|disable|limit-usage|list|remove> [path] [value] - quota
translator specific operations
volume top <VOLNAME> {[open|read|write|opendir|readdir] |[read-perf|write-perf bs <size> count
<count>]}  [brick <brick>] [list-cnt <count>] - volume top operations
peer probe <HOSTNAME> - probe peer specified by <HOSTNAME>
peer detach <HOSTNAME> - detach peer specified by <HOSTNAME>
peer status - list status of peers
peer help - Help command for peer
quit - quit
help - display command options
```

テスト条件

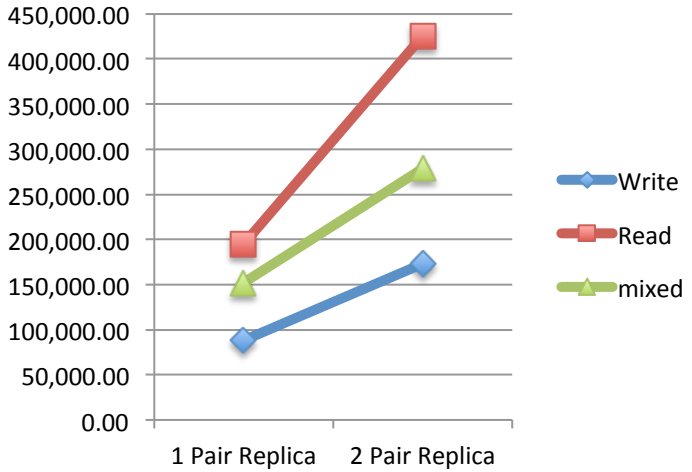
- 1GbEネットワーク
  - フルスイッチ
- クライアント6台 (CentOS)
  - 2GB メモリ以上
  - 102～103MB/sのNetパフォーマンス (各クライアント対サーバ)
- サーバ4台
  - 6GB メモリ以上
  - 2TB SATA × 8 (RAID 0)
  - 500～800MB/sのディスクパフォーマンス (ext3)

- ベンチマーク測定
  - IOzone (http://www.iozone.org/) を使用
  - 128Kブロックで、1GBのファイルを読み書き
- ネットワークの負荷要件
  - クライアントのネットワークIFが飽和する程度の負荷
  - サーバ1台に付き4台のクライアントが目安
  - クライアント台数が足りないので、ベンチマークを複数個実行している

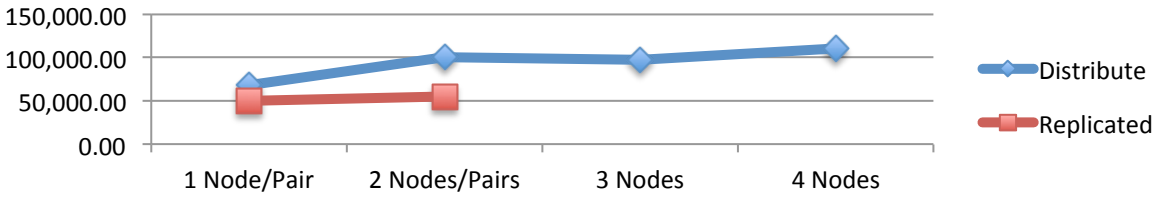
7-2. パフォーマンス測定結果



Distributed Nodes (Native)



Replicated Nodes (Native)



NFS Write throughput

単位はKB/sec

7-3. プロトコルごとのパフォーマンス傾向

Native Protocol

- FUSEによるクライアント負荷が高い
  - ベンチマーク中は、ロードアベレージが2～3以上
  - FileBenchによるベンチマークでは好成績が出ない
- サーバ側の負荷はそれほど高くない
  - ベンチマーク中は、ロードアベレージが0.7～1.8程度
- ノード数に対して、ほぼリニアに性能が向上
  - 実測したワイヤスピードに近い
- Replicaの特性
  - 書き込みパフォーマンスが低下
    - ノード数を増やすと論理値(半分)に近づくか
  - 読み出しパフォーマンスは向上

NFS

- クライアント負荷は低い
  - カーネルモードNFS
- サーバ側の負荷は高め
  - ベンチマーク中は、ロードアベレージが2.0～2.5程度
- 書き込み性能は悪い
  - NFSですから、、、、
  - パフォーマンスはLinux Nativeと同等
- 読み込み性能は高い(？)
  - クライアントのキャッシュ効果
    - 測定値が大きくばらつく
  - ノード数の効果があまり出ない

CIFS

- パフォーマンスはGluster Nativeと NFSの中間
- GlusterFS Native Clientでマウントしたディレクトリを、Sambaでエクスポートする

8-1. CTDBによる単一障害点の排除

NFS/CIFSは、プロトコル的に1台のサーバとしかやりとりできない！

→単一障害点となる

設定ファイル

- /etc/sysconfig/ctdb ファイル
  - 共有ディレクトリのパス
  - 仮想IPを与えるインターフェイス
  - サポートするサービスなどの指定
- /etc/ctdb/nodes
  - クラスタに属するノードのリスト
- /etc/ctdb/public_address
  - 仮想IPアドレスのリスト
- /etc/ctdb/event.d ディレクトリ
  - イベント発生時に実行するスクリプト

CTDBを使用してクラスタ化する

- Sambaプロジェクトの一部
  - Samba 4で本体に統合予定
  - Samba 3でも利用可能
- ハートビートによる相互監視
  - 仮想IPアドレスの切り替え
  - イベント発生時にスクリプト実行
- サポートアプリケーション
  - Samba
  - NFSサーバ、iSCSIサーバほか
- 共有ファイルシステムの利用が前提
  - 標準ではGFSなど
  - GlusterFSもOK
- 設定は簡単だがドキュメントがやや不足
  - 特に設定事例
- ダウンの検出から仮想IPの切り替えは高速
  - 数秒程度(設定次第)
  - ノード回復時は数分程度(Samba再起動まで)

8-2. CTDBとSamba/NFSを組み合わせた設定

Samba 3

- Samba 3.3以降
- オプション付きでコンパイル
  - --with-ctdb、--with-cluster-support など
  - RHELなどではバイナリ提供されている
- CTBDから自動的に起動される
- Takeover時には、クライアントが再接続して処理を続行

smb.confの例

```
[global]
        workgroup = GLUSTERFS
        security = user
        passdb backend = tdbsam
        guest account = nfsnobody
        clustering = yes
        idmap backend = tdb2
        private dir = /mnt/gluster/ctdb.var/lock
[cifsgl]
        comment = GlusterFS exported
        path = /mnt/gluster/CIFS
        posix locking = Yes
        writeable = Yes
        browsable = Yes
        printable = No
        guest ok = No
```

NFS

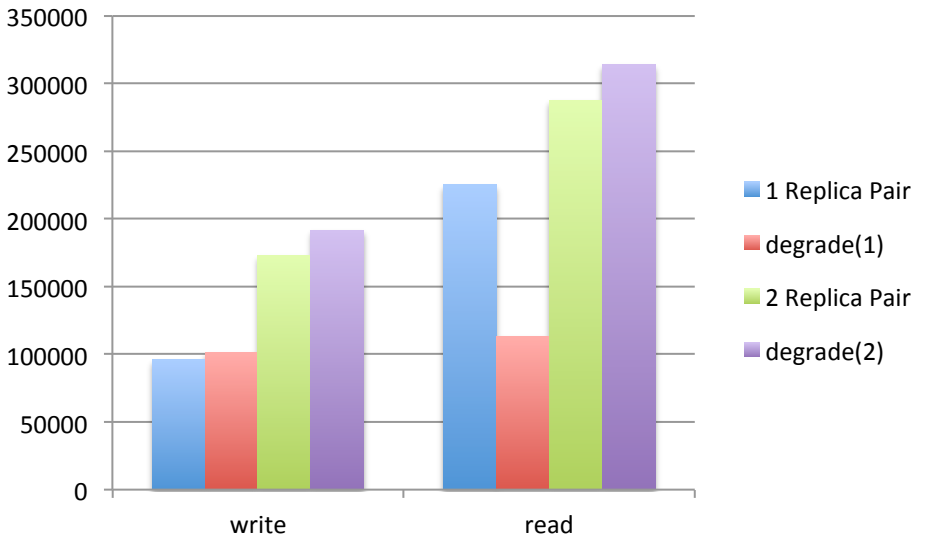
- GlusterFSのNFSを使用する場合は、仮想IPのみを利用する
  - 特別な設定は不要

2-wayのreplica (ミラー構成)

1. ベンチマークの実行
2. 1ノードの停止
3. ベンチマークの実行

ノードの追加 (2x2構成)

1. ノードの追加
2. rebalance
3. (ベンチマークの実行)
4. (1ノードの停止)
5. (ベンチマークの実行)



Q & A

コアマイクロシステムズ株式会社
Core Micro Systems, Inc.

URL：http://www.cmsinc.co.jp/　Mail：sales@cmsinc.co.jp
TEL： 03-5917-6451　IP Phone：050-5558-5410　FAX 03-5917-6452
本社 〒173-0026 東京都板橋区中丸町11-2 ワコーレ要町ビル9F