単一スクリプトによる分散ファイルシステム

荒川　淳平†　　笹田　耕一†

1. はじめに

クラウドの登場などにより，分散ファイルシステムに再び注目が集まっている．

しかし，既存の分散ファイルシステムは，複数のコンポーネントから構成されており，導入や設定が困難である場合が多い．また，コードセットも巨大であるため，既存の機能やアルゴリズムを修正したり，新しい機能を追加したりすることが難しい．この二つの問題は，分散ファイルシステムの利用と今後の発展を滞らせる要因となり得る．

そこで我々は，性能のスケーラビリティや単一故障点が存在しないことを満たしつつ，簡単に導入が可能で，簡単に中身を弄れる分散ファイルシステム yass（yet another simple storage）を開発した．

2. 設計および実装

2.1 サーバサイド P2P とクライアント

yass はサーバ/クライアントモデルの分散ファイルシステムである．サーバサイドは対等な複数のノードによる P2P 型のアーキテクチャを採用している．このことは，特別な役割を持つノードが存在せず，単一故障点の排除につながる．また，サーバサイドのプログラムを 1 種類にすることで，導入が必要なコンポーネントの種類を減らすこともできることも P2P アーキテクチャを採用した理由である．各ノードは URL で同定され，HTTP を用いてお互いに通信を行う．クライアントとの通信も同様に HTTP を用いて行う（図 1）．

そして yass はウェブサーバ上で動作する単一の PHP スクリプト（yass.php）として実装した．PHP スクリプトをウェブサーバ上で動作させることを選択した理由は，通信処理の基礎的な部分を PHP 処理系及びウェブサーバが受け持つため，コード量を大幅に削減できるからである．

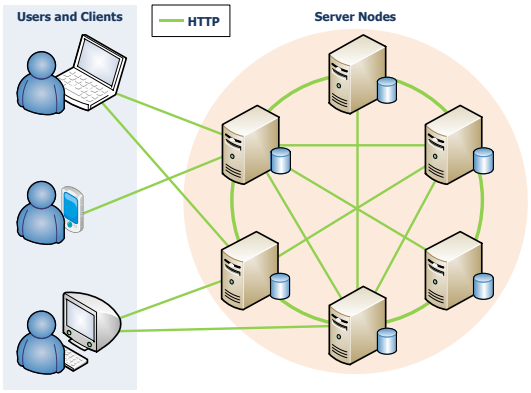

図 1: サーバサイド P2P アーキテクチャ

2.2 分散化と結果整合性

yass ではファイルの分散に Consistent Hashing を用いる．各ノードには ID が割り当てられ，リング状のハッシュ空間にマッピングして管理される．ファイルはブロック（ファイルのデータを一定のルールに従って分割したもの）とエントリ（ファイルのメタデータおよび構成するブロックのインデックス情報）に分けて分散させる．分散のキーには，それぞれブロックのハッシュ値，及び親ディレクトリの ID を用いる．

また，yass ではファイルシステムの操作（例：ファイルの保存や削除，ディレクトリ作成や名前変更）にタイムスタンプをつけたものをエントリとして追記でのみ保存する．そして，特定のリソース（ファイルやディレクトリ）について，もっとも新しいエントリの情報をそのリソースの状態とする．このため，同じリソースに対してほぼ同時刻に削除と保存といった操作が行われた場合，ノード間で一時的に異なる状態を返す可能性がある．しかし，ノード間の同期が行われた後，最終的にタイムスタンプがもっとも新しい操作の状態に落ち着く．また，結果的に任意の過去の状態を取り出せるため，バージョン管理が実現されている．

† 東京大学大学院 情報理工学系研究科
Graduate School of Information Science and Technology, The University of Tokyo

3. 評価

3.1 スケーラビリティ

我々は性能のスケーラビリティを評価するために，ベンチマークプログラムを作成し，評価実験を行った．評価実験には Pentimu 4 2.8GHz の CPU と 1GB のメモリを搭載した計算機を 15 台用いた．PHP は 5.3.1 を CentOS 5.4 上の Apache 2.2.14 で動作させた．ノード間はギガビットのイーサーネットワークで接続されている．

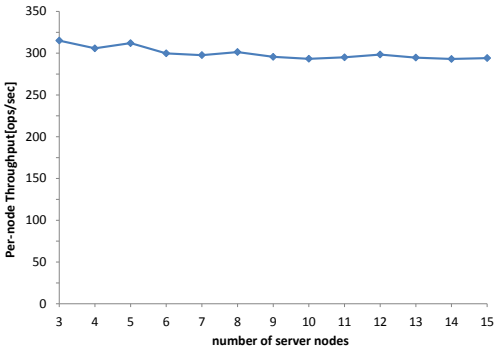

図 2: スケーラビリティ評価結果

ベンチマークプログラムとしては，ディレクトリ作成を入れ子的にひたすら繰り返すプログラムを PHP で記述して利用した．結果として，ノード数の増加に対しても 1 台当たりの性能がほぼ一定で，スケールアウトすることが確認できた（図 2）．

3.2 コード行数

機能の追加や修正の容易さを評価するために，我々は既存の分散ファイルシステムが持つ機能とそのコード行数を比較した（表 1）．

表 1: コード行数の比較評価

対象	言語	単一故障点	行数
Gfarm-2.4.0	C	MDS	66,435
HDFS-0.20.2	Java	NameNode	106,470
MogileFS-2.37	Perl	No	13,272
Ceph-0.21.2	C++	No	136,788
GlusterFS-3.0.5	C	No	161,182
Lustre-1.8.4	C	No	185,256
yass	PHP	No	2,962

結果として，yass は他の分散ファイルシステムと比べて，一桁か二桁以上コード行数が少ない．また，Gfarm や HDFS が単一故障点を持つのに比べて，yass は単一故障点を持たない．また，分散ファイルシステムに統合された機能として，yass と HDFS のみがバージョン管理機能を持っている．以上のことから，yass は高機能であるにもかかわらず，非常にコンパクトであると言える．

3.3 導入コスト

導入および設定の容易さを評価するために，我々は既存の分散ファイルシステムで個別に導入・設定する必要のあるコンポーネントと依存している OS について比較した（表 2）．

表 2: 導入コストの比較評価

対象	OS 依存	コンポーネント
Gfarm-2.4.0	Linux	MDS/FSNode
HDFS-0.20.2	No(Java)	NameNode/DataNode
MogileFS-2.37	Linux	Tracker/StrageNode/DB
Ceph-0.21.2	Linux	MDS/ODS
GlusterFS-3.0.5	Linux	GlusterFS
Lustre-1.8.4	Linux	MSD/OST
yass	No(PHP)	yass.php

結果, HDFS を除く他の分散ファイルシステムがすべて Linux に依存しているのに対して，yass は PHP が動作するすべての OS で動作が可能である．また, GlusterFS を除く他のシステムが 2 つ以上のコンポーネントの設定が必要なのに対して，yass は単一のコンポーネントの設定のみで動作が可能である．以上のことから，yass は依存性が低く，かつ容易に導入・設定が可能であると言える．

4. おわりに

我々はウェブディレクトリ上に設置するだけで導入ができる単一の PHP スクリプトで動作する分散ファイルシステム yass を開発した．開発したスクリプトファイルは 3000 行に満たず，機能追加や修正が容易である．また，この開発経験を元に，セキュリティモデルの研究や強い一貫性を提供する分散ファイルシステムの開発にも着手している．

参 考 文 献

[1] 荒川淳平, 笹田耕一, 竹内郁雄, ”yass: yet another simple storage”, 情報処理学会研究報告システムソフトウェアとオペレーティング・システム,Vol.2010-OS-113, No.11, 2010.
[2] Gfarm, http://datafarm.apgrid.org/.
[3] HDFS, http://hadoop.apache.org/hdfs/.
[4] MogileFS, http://danga.com/mogilefs/.
[5] Ceph, http://ceph.newdream.net/.
[6] GlusterFS, http://www.gluster.org/.
[7] Lustre, http://wiki.lustre.org/.