2006—OS—103　(7)
2006／7／31

障害回復のための広域分散アーカイバルストレージ

吉野純平[†]　阿部洋丈[††]　加藤和彦[†,††]

本論文では, 障害回復のための広域分散アーカイバルストレージである Sustor の実装方式を提案する. アーカイブデータの保存において, その管理やコストが問題になることがある. その問題に対して, 広域ネットワークの多数の計算機のローカルストレージを連合させることで解決を目指す. 多くの計算機が協力し合うことで, 専用の計算機を使わずに各計算機の資源の一部を使用することでアーカイブストレージを構築する. さらに, Sustor は各計算機の負担の軽減やデータの転送速度が速度を考慮して設計されている. 本論文では, そのアルゴリズムを簡潔に実装する方式を設計し, 予備実験の結果を示す.

A wide area distributed archival storage for Failure recovery

Junpei Yoshino[†], Hirotake Abe[††] and Kazuhiko Kato[†,††]

This paper proposes an implementation method of Sustor which is the archival storage that autonomous failure recovery is possible. In a save of archive data, the management and a cost can become a problem. For the problem, we aim at solution by letting a local storage of a lot of computers of WAN combine together. We build an archive storage without using an exclusive computer because many computers cooperate by using a part of resources of each computer. Sustor is designed so that reduction of a burden of each computer and a transfer speed of data become high-speed. In this paper, we design a method to implement the algorithm simply and test it preliminarily.

1. はじめに

今日, ネットワーク障害や災害などによってデータへのアクセス可能性が低下することを防ぐために, 複製を遠隔地に配置することが行われる. データの更新の頻度が高い場合や, 程度が大きい場合, 帯域保障などがある高品位なネットワークで遠隔地に接続し, 複製を配置する方法をとることが多い. ネットワークで接続する場合, 十分にコストをかけても問題にならないケースならばよいが, 低コストでも同様なことを行いたいという要求がある.

アーカイバルストレージは, 蓄積されるデータを保存するためのストレージである. 扱うデータの例としては, データベースのトランザクションのログデータなどがある. また, 仮想計算機の実行状態を保存することで, 他の計算機でその実行状態を再現するという方法もある[1]. アーカイバルストレージを広域に分散して構築し, それらのデータを保存することにより, データベースの再構築や仮想計算機の再現が広域ネットワーク上で実現できる.

本論文では, 障害回復のための広域分散アーカイバルストレージ Sustor[2] のアルゴリズムを簡潔に実装する方式を提案するとともに, 予備実験の結果を示す. 障害回復のためには, 障害が発生したときに可能な限り全てのデータにアクセス可能な状態になることが必要である. Sustor は, 障害から回復するために多数の計算機にデータの複製を配置する. 広域ネットワークで分散して複製を作ることにより, ネットワークの障害から回復する. 障害発生時にすべてのデータにアクセスできない可能性があるが, その可能性が低くなるように設計されている.

本論文の構成は以下のとおりである. 2 章では関連研究について述べる. 3 章では, 想定環境と目標について述べる. 4 章では, システム構成について述べる. 5 章では, 通信プロトコルの設計について述べる. 6 章で評価を行う. 7 章でまとめと今後の課題について述べる.

† 筑波大学システム情報工学研究科 コンピュータサイエンス専攻
Department of Computer Science, University of Tsukuba
†† 科学技術振興機構
Japan Sience and Tecnology Agency

2. 関 連 研 究

障害回復に重点を置いた分散ファイルシステムの研究として, 井口らの方式[3], StarFish[4], Myriad[5] などがある. ファイルシステムは作成, 移動, 削除など多くの操作を行える点がある. それに対して, Sustor はデータの追加と読み込みを扱う. データの追加と読み込みに機能を限定するとこでより多数のノードでデータを共有することが容易になる.

P2P 広域分散ファイル共有技術として, Ivy[6], OceanStore[7], Farsite[8],PAST[9] などがある. 多くの P2P ファイル共有技術は特定のデータを検索しデータを転送するように設計され, データ全体を得るような処理をすることを想定していない. P2P ファイル共有ですべてのデータにアクセスする場合, 多くのノードと通信しなければならないことが多い.

アーカイバルファイルシステムとして, Venti[10] などがある. Venti はローカルでアーカイバルファイルシステムを構築し, ストレージを効率的に利用し, 任意の過去のディスクイメージを取り出すことができる. Sustor は遠隔地のストレージに冗長度を高く複製を保存し, 障害回復ができるシステムを構築する.

3. 想定環境と目標

Sustor は, 数十から数百程度の協力関係にあるノード群でデータを保持する. このノード群のリストが事前に与えら得れている状況, もしくは, 得ることができる状況を仮定する. 各ノードは, Sustor 専用の計算機ではなく, それぞれ独自の処理を有していることを想定する. Sustor は, 障害回復, 高速なデータ転送, 負荷の軽減, 状況把握を目標に向けて設計されている. 以下に, それぞれの目標について述べる.

(a) 障害発生時の対応
障害が発生すると, 他のノードとの接続が切断される. その際, 代用となる接続先を決定することができることが望ましい. さらに, 可能な限り接続先は負荷分散されるように選ばれることが望ましい.

(b) 高速なデータ転送
配布中に障害が発生した場合, ノードによってデータが異なる状態になり得る. それを防ぐため, データの配布の速さは高速であることが望ましい. また, データの収集の速さは可能な限り高速であることが望ましい.

(c) 負荷の軽減
ノードは専用の計算機ではないので, 可能な限りノード群への負荷は小さいことが望ましい. Sustor において, データは分割して配置されるので各ノードのネットワークやストレージに対する負荷は, すべてをコピーするのに比べ小さくなる.

(d) 状況把握
転送開始後に分割されたデータの一部が入手不能と判明すると, それまでの転送にかかったコストは無駄になってしまう. また, 他のノードの転送を遅くする可能性も発生する. そのため, 各ノードはデータをすべて得られるかを転送前に確認できることが望ましい.

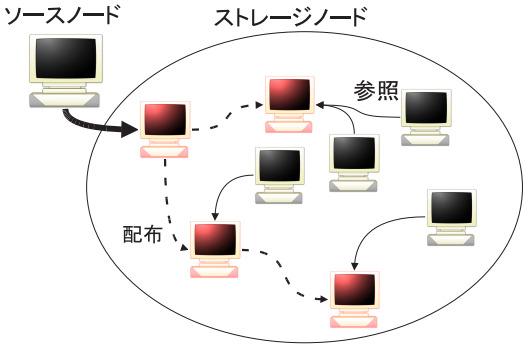

図 1 方式の概観

4. 基 本 方 式

4.1 概　　観

Sustor はデータの複製を作ることで障害発生時にもデータへのアクセス可能性を向上させた分散ストレージである. データとノードにはユニークな ID が付加される. データの配布と収集の際には, データ ID から計算によってデータを持つノードを決定できる. 具体的な計算手法は 4.2 章で述べる.

Sustor は, 2 種類のノードと 2 種類のネットワークによって構成されている. ノードは, データを保管する機能と外部からの要求でデータを読み込む機能を持つ. その機能に加えて, データ配布の指示を受ける機能と他のノードに対してノードリストを送信する機能を持つノードをソースノードと定義する. また, ソースノードではないノードをストレージノードと定義する. ネットワークには, データアクセスオーバレイネットワークとデータ配布オーバレイネットワークがある.

図 1 は, Sustor の方式の概観を示したものである. 図中のノードで大きなものがソースノード兼ストレージノードであり, それ以外のノードがストレージノードである. ソースノードがあるデータを配布している

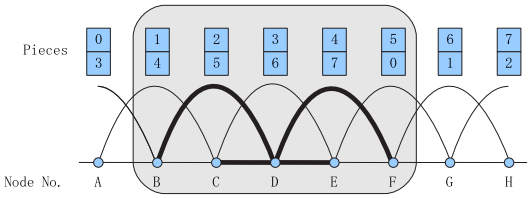

図 2 DAO ネットワークの概要

様子を示している．図中の太い矢印がソースノードからの配布を示す．また，破線の矢印がデータの配布を示すデータ配布オーバレイネットワークであり，細い矢印がデータの参照を示すデータアクセスオーバレイネットワークである．障害が発生した場合，図中の接続が切れる場合がある．その時，各ノードはノード ID から接続すべきノードを計算によって決定し，再接続を行う．

4.2 データアクセスオーバレイネットワーク

データアクセスオーバレイ (Data Access Overlay. 以下，DAO) ネットワークは，各ノードがデータにアクセスするためのネットワークである．図 2 は DAO ネットワークの概形を示している．このネットワークは，近隣のノードに対して接続している．各データを保持すべきノードは次の式が成立するノードである．次の式におけるパラメータ k は近隣ノードとの接続数，L はノード数を示す．このパラメータによって各ノードが保持すべきデータの ID が決まる．

$$\begin{aligned}\{i | i &= i_n + (L - (2k+1))j \bmod L \\ &= i_n + L - (2k+1)j \bmod L \\ &\left(0 \le j \le \left\lfloor \frac{L}{2k+1} \right\rfloor\right)\}\end{aligned}$$

図 2 は，$k = 2, L = 8$ の例である．どのノードからも左右の各 2 ノードと通信ができればすべての ID のデータにアクセスすることができる．図 2 は DAO ネットワークのトポロジを便宜上直線的に描いたものである．しかし，実際の DAO ネットワークは図 3 のような円形のトポロジを持つ．

DAO ネットワークは，ショートカットパスを持つ．ショートカットパスはランダムに選んだノード間を接続する．システムパラメータのショートカット作成率にしたがって，DAO ネットワークにショートカットパスを追加する．ショートカットパスは，ソースノードが作成する．

4.3 データ配布オーバレイネットワーク

データ配布オーバレイ (Data Dissemination Overlay. 以下，DDO) ネットワークは，データを配布するためのネットワークである．図 3 は DDO ネットワーク

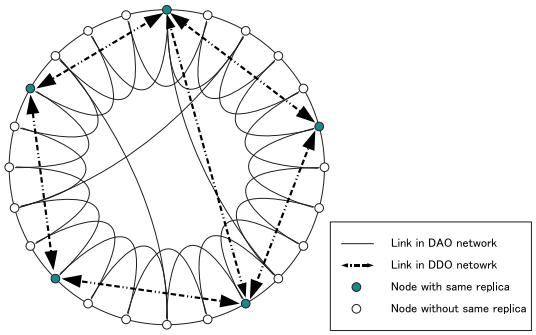

図 3 DDO ネットワークの概要

の概形を示している．図中のノードのうち，一つはソースノードであり，それ以外はストレージノードである．このネットワークは，同じデータを持つべきノード同士で接続している．持つべきデータの ID は，DAO を構築した際に決定したものである．DDO ネットワークは，ノード番号が近隣のノードで保持すべきデータ番号が同じノードと接続する．さらに DAO ネットワークのショートカットパスを考慮して，DDO ネットワークのショートカットパスを作成する．

システムパラメータの近隣ノードとの接続数を大きくすると，データのコピー数が減少する．これによって，各ノードの負荷を軽減することができる．しかし，障害発生時にすべてのデータにアクセスできるノードの数が減少する．

4.4 動　　作

ソースノードは，事前に与えられたノードリストを持つ．ソースノードは，パラメータで与えられたショートカットパス作成率でショートカットパスを作成する．ソースノードは，ノードリストにあるストレージノードに対してノードリストとショートカットパスの情報を配布する．

ストレージノードは，ソースノードからノードリストを受け取るとリストのサイズから全体のノード数がわかり，自身の ID がわかる．全体のノード数と自身の ID から DAO ネットワークと DDO ネットワークの接続先のリストを構築する．計算が終了すると，DAO ネットワークと DDO ネットワークの接続先にノードリストからアドレスを得て接続をする．

各ノードは定期的に接続を確認し，データを受け取ると DDO ネットワークに対してデータを送る．接続が切断されると，DAO ネットワーク接続先リストと DDO ネットワーク接続先リストから，それぞれ次に優先度の高いノードを決定し接続する．データ待ち状態のときに，データの取得要求がくると DAO ネットワークで接続されたストレージノードからデータを取

得する．

5. 通信プロトコルの設計

システム構成で述べた DAO ネットワーク, DDO ネットワークをソースノードとストレージノードで構成する．そのための通信としてノードリストを転送するもの, データの転送するもの, データの情報を転送するものの 3 つである．以下では, 各通信について述べている．

5.1 ノードリストの転送

ノードリストの転送処理は, ソースノードがストレージノードに対して行うものである．ソースノードは, ノードリストとショートカットリストをストレージノードに送信する．ストレージノードは, それらの情報を元に DAO ネットワークと DDO ネットワークの接続先を計算する．

まず, 計算において各データ番号のデータを持つべきノード番号の対応リストを作成する．以下の式が計算式である．

$(DataID + (2k+1) * i) \bmod L \left(0 \leq i \leq \left\lfloor \frac{L}{2k+1} \right\rfloor\right)$

データ番号とノード番号の対応表を表 1 に書く．その際のシステムパラメータは, $k = 2, L = 51$ とする．

表 1 各データ番号をもつノード番号表

データ番号	ノード番号					…	
0	0	5	10	15	20	…	50
1	1	6	11	16	21	…	46
2	2	7	12	17	22	…	47
30	30	35	40	45	50	…	29
50	50	4	9	14	19	…	49

このノード番号の対応リストができると各通信のためのアクセス先を計算する．DAO アクセス先リスト, 左 DDO アクセス先リスト, 右 DDO アクセス先リスト, ショートカット DDO リストが作成される．リストは各データ番号に対して作られる．リストは優先順位が高いアクセス先の順番でソートする．リストの前から順に接続可能かを定期的に確認することで, どのデータ番号のデータに対しても即座にデータを転送収集することができる．

まず, DAO でアクセスするノードの番号を計算する．DAO の接続先は各ノード番号に依存している．データ番号全てに対してデータを持つべきノードの番号が自身のノード番号に近い順にソートすることで, DAO ネットワークでアクセスする先が決定する．ソートは, 図 4 の評価式において x が小さいほうから順になるように行う．ノード番号 30 の処理結果を表 2 に示す．

```
A = abs(自身のノード番号-ソート対象のノード番号)
if(A  > L / 2){
   x = L - A
}else{
   x = A
}
```

図 4 ソートのための評価式

表 2 の使い方は, データ番号 2 を取得するときは, ノード番号 32 番のノードにアクセスする．もし接続ができない場合, ノード番号 37 番のノードにアクセスする．これを再帰的に繰り返す．

表 2 DAO ネットワークのアクセス先リスト (ノード番号 30 の場合)

データ番号	ノード番号					…
0	30	35	25	40	20	…
1	31	36	26	41	21	…
2	32	37	27	42	22	…
30	30	35	25	40	20	…
50	29	34	24	39	19	…

つぎに DDO ネットワークについて計算する．まず, リング型のトポロジを左にまわる方向のデータの転送について計算する．計算の方法は, 自分のノード番号より大きな番号を持つものが優先度が高くなるように対応リストをソートする．表 3 はノード番号 30 番のノードのリストを示したものである．

表 3 DDO ネットワークのアクセス先リスト (ノード番号 30 の場合)

データ番号	ノード番号					…
0	35	40	45	50	0	…
1	36	41	46	1	6	…
2	37	42	47	2	7	…
30	35	40	45	50	0	…
50	34	39	44	49	50	…

右にまわる方向のデータ転送については, ソートを逆順にしたものである．さらに, ショートカットのリストを作成する．ショートカットは, 転送先を元とし

た左回りの DDO ネットワークリストのリストを使う．これによって，ショートカット先のノードにアクセスができなくなったとしても，その周辺のノードにアクセスを試みることができる．

5.2 データの転送

データの転送には 2 つのパターンがある．一つは，各ノードが他のノードから要求を出してデータを転送する能動的転送である．もう一つは，各ノードがデータを他のノードから受け取ったときに他のノードに転送する受動的転送である．

データの能動的転送処理は，ソースノードとストレージノード間，または，ストレージノード間で行うものである．あるノードが特定のデータを必要とするときに DAO ネットワークまたは DDO ネットワークで接続されたノードに対して，ソースノードとデータ ID をデータを受け取るものである．

データの受動的受信処理は，ソースノードとストレージノード間，または，ストレージノード間で行うものである．DDO ネットワークで接続された他のノードからデータが転送されるときに呼び出される．通信内容はソースノード，データ，データ番号，転送元である．この処理は，呼び出されたノード自身が DDO ネットワークで接続された他のノードに対して再帰的に実行される．ただし，転送元に対しては転送しない．

5.3 データ情報の転送

このプロトコルは各オーバレイネットワークによって使われ方が違う．DAO ネットワークでは，各ノードが持っている最新のデータを通知するために使う．DDO ネットワークでは，データ転送の際のデータの所持確認に使う．

DAO ネットワークでは，定期的に接続されたノードに対して生存確認を行う．その際，DAO ネットワークで接続しているノードに対しては，自身の持つ最新のデータ情報を送信する．各ノードは受け取った最新のデータ情報を元に自身のデータが最新のものであるか計算する．ノード番号と全体ノード数と現在持つデータの番号から，次に受信するであろうデータの番号は計算できる．ここで，DAO ネットワークで接続されたノードから渡されたデータ番号が予測している番号よりも新しい場合，障害等の原因によって持つべきデータが受信できなかった可能性がある．その際は，他のノードからデータの能動的受信を行う．

DDO ネットワークにおけるデータ転送は図 3 で示した円形のトポロジで左右から行われる．よって，データの転送が完了しているノードに対して再度転送が行われる可能性がある．もしデータを持っている場合，再度データの転送を始めてしまうと，その転送が無駄になる．よって，ストレージノードは他のノードに対してデータを転送する際は相手にデータを持っているか確認しなければならない．

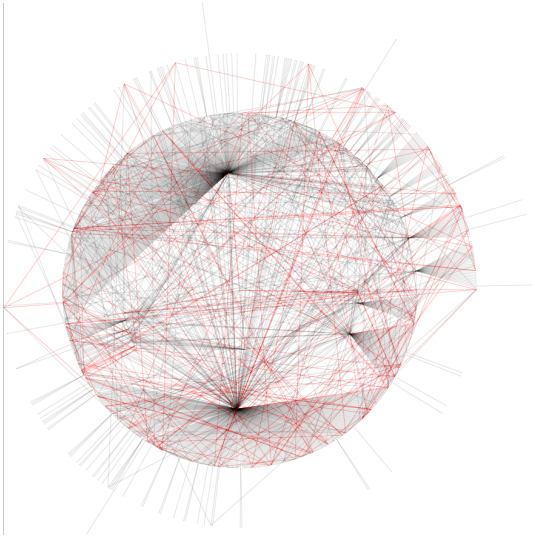

図 5 構築した仮想インターネット

6. 性能評価

SSF ネットワークシュミレーター[11] を使った結果を示す．インターネットのトポロジーをエミュレーションするために，scale-free ネットワークモデルを使う[12]．その中の一つである Generalized Linear Preference(GLP) モデル[13] を使うことでインターネットに似たネットワークができることが知られている．このネットワークを使用してアルゴリズムの有効性を確認する．図 5 は，構築した仮想インターネットを示す．システムのパラメータとして，$k = 2, 3, 4$, $L = 100$, ショートカット率 0.01 として評価した結果を以下で示す．

6.1 DAO ネットワークの生存率

Sustor では，DAO ネットワークで接続先となっているノードが接続可能ならば，即座にデータを転送し，必要なデータを集めることを目的としている．実験では，エミュレーションしたインターネットでルータを通信不可能にしたとしても，DAO ネットワークが切断されないノードがどの程度あるかを評価した．各ルータはいくつのネットワークを中継しているかが異なる．多くのネットワークを中継しているルータを通信不可能にしたときは，ネットワークが切断される可能性が高くなる．実験結果で得られた DAO ネットワークが切断されていないノードは，データを読み込むことが

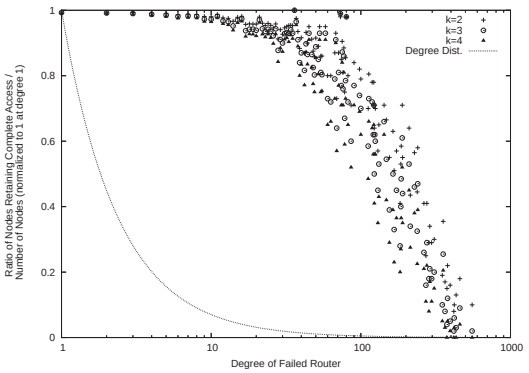

図 6 データへのアクセス可能性

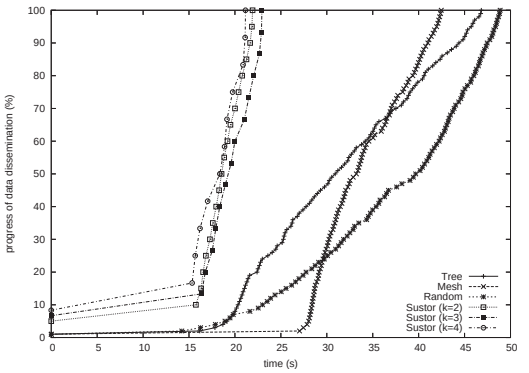

図 7 データの配布時間

できる.

図 6 は, DAO ネットワークが切断されていないノードの割合を示す. 横軸は, 通信不可能にしたルータの中継しているネットワークの数を表し, 縦軸は全体のノードの中で DAO ネットワークが切断されていないノードの割合を示す. この評価より, システムパラメータの近隣ノードとの接続数 k が小さいほうが障害に対して耐故障性を持っていることを確認できた. 全体で大部分を占める中継するネットワークの数が 100 以下であるルータで障害が発生したとしても, 6 から 8 割程度のノードはすべてのデータにアクセス可能だとわかった. 中継するネットワークの数が 500 以上であるルータで障害が発生するとほぼ全てのノードがデータにアクセス不可能になる. しかし, 今回の実験においては $k = 2, 3$ の場合, その確率は 0 にならなかった.

6.2 データ配布時間

DDO のデータ配布時間を評価した. 評価は, すべてのノードがデータにアクセスするための準備が整うまでの時間を計測する. この評価は障害が発生していない状況を想定している. 図 7 のデータは, Sustor でシステムパラメータである近隣ノードとの接続数を 2, 3, 4 の場合で確認したものと Tree, Mesh, Random のトポロジを持つオーバレイネットワークを構築した場合を示した. 実験の結果より, 全体に配布が完了するまでの他の手法に比べ速いことが確認できた.

7. まとめと今後の課題

本論文では, Sustor の実装の設計について述べた. また, インターネットをエミュレーションしてネットワークの障害に対するアクセスの可能性について評価した. 予備評価実験から大規模な障害に対しても少数のノードは DAO ネットワークを維持できるとわかった. データの転送速度に関しては, 配布する対象のノードが全体のノード群の一部であることから高速であると確認できた.

実装が完成した時には, クラスタを使用した仮想インターネット上で数千ノードレベルの大規模実験を行う. その上で, 最適なシステムパラメータを決定する. また, PlanetLab[14] を使用した実際の広域での実験も行いたい. 最終的には, サステーナブルサービス[1] のモジュールとして動作させ, その評価を行う.

参 考 文 献

1) 小磯知之, 阿部洋丈, 鈴木与範, Potter, R., 池嶋俊, 加藤和彦：サステーナブルサービスを実現する基盤ソフトウェアの設計., *SACSIS*, pp. 201–209 (2006).
2) Abe, H. and Kato, K.: Sustor:Distributed Storage for Disaster Recovery Using the Small-World Model., *IEEE International Conference on Computer and Information Technology* (2006). (to appear).
3) 井口寧, 渡辺浩二, 松澤照男：信頼性を考慮したグリッド向け自律分散ストレージシステム, 情報処理学会論文誌コンピューティングシステム, Vol.47, No.7, pp.219–230 (2006).
4) Gabber, E., Fellin, J., Flaster, M., Gu, F., Hillyer, B., Ng, W. T., czden, B. and Shrive, E.: StarFish: highly-available block storage., *USENIX Annual Technical Conference, FREENIX Track* (2003).
5) Chang, F., Ji, M., Leung, S.-T. A., MacCormick, J., Perl, S. E. and Zhang, L.: Myriad:Cost-effective Disaster Tolerance., *FAST* (2002).
6) Muthitacharoen, A., Morris, R., Gil, T. M. and Chen, B.: Ivy: a read/write peer-to-peer file system, *SIGOPS Oper. Syst. Rev.*, Vol.36, No.SI, pp.31–44 (2002).
7) Kubiatowicz, J., Bindel, D., Chen, Y., Cz-

erwinski, S., Eaton, P., Geels, D., Gummadi, R., Rhea, S., Weatherspoon, H., Wells, C. and Zhao, B.: OceanStore: an architecture for global-scale persistent storage, *SIGOPS Oper. Syst. Rev.*, Vol.34, No.5, pp.190–201 (2000).

8) Adya, A., Bolosky, W. J., Castro, M., Cermak, G., Chaiken, R., Douceur, J. R., Howell, J., Lorch, J.R., Theimer, M. and Wattenhofer, R. P.: Farsite: federated, available, and reliable storage for an incompletely trusted environment, *SIGOPS Oper. Syst. Rev.*, Vol.36, No.SI, pp.1–14 (2002).

9) Druschel, P. and Rowstron, A.: PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility, *hotos*, Vol.00, p.0075 (2001).

10) Quinlan, S. and Dorward, S.: Venti: A New Approach to Archival Storage, *FAST '02: Proceedings of the Conference on File and Storage Technologies*, Berkeley, CA, USA, USENIX Association, pp.89–101 (2002).

11) : Scalable Simulation Framework. `http://www.ssfnet.org/`.

12) Barabasi, A.-L. and Albert, R.: Emergence of scaling in random networks, *Science*, Vol.286, pp.509–512 (1999).

13) Bu, T. and Towsley, D.: On distinguishing between Internet power law topology generators, *INFOCOM'02* (2002).

14) Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M. and Bowman, M.: PlanetLab: an overlay testbed for broad-coverage services, *SIGCOMM Comput. Commun. Rev.*, Vol.33, No.3, pp.3–12 (2003).