DEIM Forum 2011 C10-3

構成ノード電源停止によるシステム省電力化のためのインメモリ分散データストア設計

小林　大† 菅　真樹†,†† 大野　善之† 鳥居　隆史†

† NEC システムプラットフォーム研究所　〒 211–8666 神奈川県川崎市中原区下沼部 1753
†† 東京工業大学 大学院情報理工学研究科　〒 152–8552 東京都目黒区大岡山 2-12-1
E-mail: †{daik@ay,kan@bq,y-ohno@ji,t-torii@ce}.jp.nec.com

あらまし　ネットワーク接続された多数の計算機（ノード）の主記憶を利用した大容量かつ高速なデータストアが注目されている．一方，その大きな消費電力は問題であり，システム構成ノードの一部をシステム負荷に応じて動的に電源停止・復帰する省電力手法が有効である．本稿では，より多くのノードを停止しつつ，各ノードが電源停止・復旧時の要求性能を維持するため，柔軟なデータ配置機構，複製へのアクセス転送，他ノードへのライトオフローディング，タイムアウト抑制ネットワーク制御の導入を提案する．また，これらの機能を分散 KVS 実装に導入したプロトタイプシステムを利用し，省電力効果とアクセス性能を測定した実験の結果について紹介する．

キーワード　クラウドコンピューティング，分散システム，ストレージ，省電力化，効率化

Design of Power-Thrifty In-memory Distributed Data Store

Dai KOBAYASHI†, Masaki KAN†,††, Yoshiyuki OHNO†, and Takashi TORII†

† System Platforms Research Laboratories, NEC Corporation
1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211–8666 Japan
†† Department of Computer Science, Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-Ku, Tokyo 152–8552 Japan
E-mail: †{daik@ay,kan@bq,y-ohno@ji,t-torii@ce}.jp.nec.com

1. はじめに

計算機が搭載可能な主記憶量の増大に伴い，ネットワーク接続された多数の計算機（ノード）の主記憶を利用したインメモリ型の分散データ格納システム（データストア）がその高速性から注目されている．主記憶に用いられる DRAM は揮発性のためインメモリ型分散データストアでは，無停電電源装置の利用に加え，格納データの複製あるいは更新ログを複数の計算機に格納することでデータの喪失を防ぐ．このようなシステムの例として，H-Store [1] などの関係データベース, 楽天の ROMA 等の分散 Key-Value ストア などがある．また RamCloud [2] は Key-Value をベースにインデックスや構造化データなどの機能の一部を実現している．

一方，その大きな消費電力は問題である．システムを構成する 1 ノードあたり主記憶量は十から百 GByte 程度に過ぎず，さらに格納データの複製保持の数だけシステム全体として多くの主記憶量を必要とする．そのため，システム構成ノード数が増加し，容量あたりの消費電力が従来の HDD を用いたシステムよりずっと大きくなってしまう．データストアへのアクセスは恒常的に高負荷ではなく，データ需要の変化に応じ大小の波がある．少なくともシステム全体が低負荷の場合について，負荷量の減少に応じて消費電力を削減することを考える．

従来のディスクアレイによるストレージシステムでは一部の HDD を省電力モードに移行することで省電力化を行っている．従来のディスクアレイでは，CPU やメモリの搭載量に対し接続される HDD の数が多く，したがって電力がシステムで支配的である．よって，システム負荷に応じて一部の HDD を省電力モードに移行する電力削減の効果は大きい．例えば Narayanan らの研究 [3] では，HDD を停止し，停止中の HDD への書き込みアクセスは別の HDD に一時的に記録することで可用性とアクセス性能を維持する．合田らは，DBMS のクエリプランナの情報を用いて，性能を維持しつつ HDD を省電力モードに落としている [4]．また引田らは複数の計算機を用いたストレージシステムで，HDD を停止しキャッシュを使って性能を維持している [5].

一方，インメモリ分散データストアにおける省電力化では，

計算機ノード全体を省電力モードに移行するアプローチを取る必要がある．計算機のメモリ部分の消費電力量は全体の中でわずかであり，その他の多くの消費電力は負荷に拠らない．計算機ノードの消費電力は計算機の負荷によって増減する変動部分と，負荷に寄らず一定の量を消費する定常消費部分がある．例えば DB サーバでは定常消費が 54%を占め [6]，WEB サーバではより大きい割合の定常消費が報告されている [7] ．そのため，システム全体の性能あたり消費電力をより大きく削減したい場合，ノードごとに電力停止し，定常消費される電力を削減することが重要となる．データ格納ではなく計算を主体とするシステムでは，定常消費を削減するためシステム全体の負荷量をある特定のノード群に割り当て，その他のノードの電力消費を停止する方法がある [8].

本稿では，インメモリ型分散 Key-Value ストアの省電力化について論ずる．我々は，効率のよいデータインテンシブスケーラブルコンピューティング向けに提唱するデータセントリック分散制御 [9] に基づき，インメモリ型分散 Key-Value ストア DKVS を開発している．そこで，DKVS において導入したシステム構成ノードの一部をシステム負荷等に応じて動的に電源停止・復帰する機能について考える．分散データ格納システムではノード数を増加することで，容量増加，冗長性維持，スループット性能向上が同時に得られる．このうち容量と冗長性を確保しつつ需要量が細かく変動するスループットに応じてノードを電源停止し通電ノードを減らすことを考える．

インメモリ型分散 Key-Value ストアにおいて検討すべき要件は，省電力効果向上，アクセス機能とデータ冗長性維持，レイテンシ維持であった．省電力効果向上とは，より多くのノードを停止出来ることである．このためには格納データのうち複製がひとつはアクセス可能なノードに残るような柔軟なデータ配置が必要となる．アクセス機能とデータ冗長性維持とは，データを保持するノードを停止させた際にも，データへのリード/ライトアクセス機能を提供し，かつ耐故障性のためのデータ冗長性を保持し続けることである．このために，停止ノードに対するリードアクセスは通電ノードの複製により提供し，ライトアクセスは本来の複製格納先とは別の通電ノードに更新部分のみ保持させる．レイテンシ維持は，ノードの電源停止などのシステム構成変更時でもクライアントアクセス性能を大きく劣化させないことである．

本稿における我々の貢献は次の通りである：

- スケーラビリティと省電力を両立するための，階層データ配置制御を導入したインメモリ型の分散 KeyValue ストアの設計について紹介. (3.)
- ノード電源停止の際，リードアクセス，ライトアクセスを他のノードに任せることにより，アクセス可用性とデータ冗長性を維持するアクセス制御手法を提案．(4. 2)
- ノード電源停止の際，クライアントからのアクセスをタイムアウトなく他ノードに転送しアクセスレイテンシを維持するネットワーク制御手法を提案．(4. 3)
- インメモリ型分散データストア DKVS プロトタイプシステムを用いた初期実験の結果を紹介し，低電力化とアクセスレイテンシ維持の両立を提示．(5.)

2. 関 連 研 究

データセンターにおけるシステム負荷と消費電力の関係について，現在非線形であるものをより線形に近づけること (energy proportionality) の重要性については，Barroso らによって言及されている [10].

分散データストアにおいて，その構成要素である HDD や計算機ノードを停止することによって，消費電力を抑制させる研究が幾つか行われている．これらはその構成要素の利用率が低いことを前提にしている．MAID [11] はそのデータアクセスの局所性を考慮し、アクセスの無いディスクを停止させることにより電力を削減させている．WriteOffloading [3] は，停止 HDD に格納されているデータに対する Write により，停止 HDD の再起動待ちによる性能劣化を防ぐために他の HDD に一旦 Write して置くことでデータの冗長性とアクセス性能維持を両立させている．

大規模かつクラスタベースの分散データストアに対して，ノードを停止させることにより電力効率性を向上させる研究が数多く行われている [12–15]．[16] は，Hot/Cold Zone という HDFS クラスタを用意し，データの利用頻度に応じて格納する Zone を分け省電力化を達成している．また，例えば [12] は，データブロックの少なくとも 1 つのレプリカを Covering Subset と呼ばれる停止させないノード群に格納させ，その他のノードを停止させることで，アクセス可用性を保ちつつ省電力化を図る．これらの研究は，主としてデータのアクセス性能を維持させるためのデータ配置方式に着眼している．我々は，データストアとしての外部へのサービス維持を念頭に，特にノード停止時のクライアントからのアクセスタイムアウトの発生の抑制に着眼している．

3. DKVS の概要と省電力向けデータ配置機構

ここではまず，我々の開発中の例からインメモリ型分散データストアの仕組みの概要を示し，その後省電力化のためのデータ配置機構について説明する．まず，我々が開発中の分散 Key-Value ストア DKVS の概要を示す．その後，より多くのノードを停止しつつ，システムのスケーラビリティを保つため DKVS に導入した階層管理によるデータ配置機構について説明する．

3.1 DKVS

DKVS とは，分散 Key-Value ストアの一種である．DKVS ではデータは複数のプロパティを合わせたオブジェクトと呼ぶデータ単位に対しシステム内で一意の Key 値を付与し格納する．オブジェクトの複数の複製が複数の計算機の主記憶に格納される．データは必要に応じて HDD 等の不揮発性媒体にも同期・非同期に記録できる．また，オブジェクトレベルの一貫性があり，同じオブジェクトへのアクセスはトランザクショナルに行うことが出来る．これは複数あるオブジェクトの複製のうちひとつをプライマリとし，プライマリオブジェクトを持つノードが主となりその他のバックアップオブジェクト間の一貫性を管理すること，及び各ノード内ではオブジェクトは複数バー

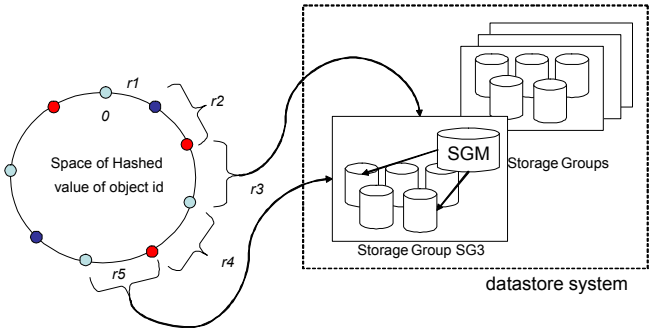

図 1　DKVS の 2 段階データ配置方式

ジョンを保持・管理することで実現している．

クライアントプログラムからは，クライアント計算機内のライブラリを通し直接オブジェクトを持つノードにアクセスされる．クライアントライブラリはシステム構成ノードリストの非同期キャッシュを保持しており，このノードリストキャッシュを元に後述するデータ配置機構を用いてオブジェクトを持つノードを特定する．

3.2　省電力のための自由度の高いデータ配置

システムのスケーラビリティを担保するため，分散 Key-Value ストアではコンシステントハッシュ等を用いてデータ配置を行うことが多いが，システム電力制御のためには柔軟性に欠ける．データ配置とは，データを作成・格納するノードの決定や，アクセス対象データの格納ノードの特定である．メタサーバ等の集中管理では，集中管理部分の性能がボトルネックや単一障害点になる．そのため，大規模なシステムでは各クライアントそれぞれ独立して算術的にデータ格納ノードを特定できるコンシステントハッシュが好まれる．

システム省電力化のため一部のノードの電源を停止しつつシステムを運用するためには，各オブジェクトごとに最低ひとつの複製が，通電中のノードに配置される必要がある．すべての複製格納ノードが停止されると，システムはリードアクセスするために，ノードを電源復帰させねばならず，これは非常に時間がかかる．しかし単純なコンシステントハッシュ方式では，データ配置箇所はキーのハッシュ値により定まってしまう．そのため，複製のひとつを維持する制約下では，電源停止できるノードが減ってしまう．単純な解決方法は，各ノードごとにプライマリのみ格納，バックアップのみ格納，と役割を決めることである．これは通常運用時やノード障害復旧時の負荷分散の観点で好ましくない．よって，分散管理によるスケーラビリティと，電源停止のためのデータ配置自由度を持つデータ配置方式が必要となる．

3.3　階層データ配置管理による自由度とスケーラビリティ

そこで DKVS では，スケーラビリティと電源制御のための柔軟性を両立するため， 2 段階の階層管理を行う．図 1 に概要を示す．DKVS では，システム構成ノード群を複数のストレージグループ (SG) に分割する．各 SG には 1 台の SG マスターノード (SGM) を任命する．システムはコンシステントハッシュ方式によりオブジェクトキーのハッシュ値空間を分割し，各領域にひとつの SG を割り当てる．SG 内のデータ配置は SGM を集中管理部とし，自由に配置する．ひとつの SG は複数のハッシュ値空間を受け持ってもよい．

この階層管理方式でのクライアントアクセスは次のとおり行われる．クライアントはまずオブジェクトキーのハッシュ値から SG を特定する．その SG へのアクセスが初めてならノードリストサービスから SGM アドレスを取得し，SGM にアクセスする．SGM はオブジェクトを持つノードのアドレスを返却する．クライアントはノードアドレスを取得し，当該ノードへアクセスする．

このように 2 段階管理し，SG 内の自由度を増すことで，オブジェクトの複製を適切に配置し，システム低負荷時にはより多くのノードの電源を停止することが出来る．このような柔軟なデータ配置は将来的には，低頻度アクセスオブジェクトを寄せて電源停止や，類似オブジェクトの重複排除，分散処理時のデータ転送ボトルネック回避などにも利用できる．データ配置柔軟性の応用例については [9,17] を参考とされたい．

一方，2 段階アクセスによりレイテンシは低下するため，クライアントに対し 2 回目以降のアクセスでは直接格納ノードへアクセスするための機構を入れる．2 回目以降のアクセスの場合，取得したノードアドレスをクライアントライブラリ内にキャッシュし，ノードに直接アクセスすることで，平均的にはレイテンシを削減できる．ノードアドレスをキャッシュするため，SG 内のオブジェクト再配置など，クライアントが間違ったノードにアクセスすることがある．よってクライアントは，ノードアドレスキャッシュを利用しオブジェクトの存在が確認できない場合，再度 SGM へ問い合わせをすることで，正しいノードを取得する．

SGM は SG 内では単一障害点になるため，SGM 間で障害監視を行う．DKVS ではすべての SGM がお互いに通信し，ひとつの代表ノード（RootMaster) を選出する．この RootMaster はすべての SGM の死活監視を行い，SGM 障害の際は，RootMaster が SGM 再選出等の処理を行う．また，RootMaster 障害に対応するため，各 SGM が RootMaster の死活を監視し，RootMaster 障害時には残りの SGM から再選出する．

4.　電源制御とアクセス分配

前節で述べたデータ配置の柔軟性により，各オブジェクトの複製のうち最低 1 つを生かしたまま，その他のオブジェクトを格納するノードを低電力モードへ移行することで，システム全体の消費電力を削減することができる．

4.1　省電力モード

一時的にノードの使用を停止する場合，ACPI(Advanced Configuration and Power Interface) で規定された省電力モードが利用できる [18]．ACPI では低消費電力モードとして，スリープ，サスペンドやハイバネーションがある．スリープでは CPU クロックをオフに，サスペンドではメモリ以外の給電を停止する．ハイバネーションではメモリの内容を不揮発性の 2 次記憶装置に退避しすべての給電を停止する．

このサスペンドやハイバネーションを利用し，インメモリ中の格納オブジェクトを不揮発記憶に格納し電源を停止すること

表 1　省電力状態における消費電力

	スリープ (S1)	ハイバネーション (S4)
消費電力 [W]	122	1.65
電力削減率	10.9 %	98.8 %
停止処理必要時間 [sec]	2	18

でデータ喪失なくノードの消費電力を大きく削減できる．表 1 に著者らの以前の調査 [17] における省電力モードの効果を示す．これより，システム負荷の減退が十分長いと見込める状況では，S4 により消費電力を大きく減らすことが出来る．しかし電源を停止すると，格納データへのリードやライトなどのアクセスが出来なくなる．

4.2　電源停止ノードのもつデータへのアクセス

電源停止中データへのアクセスを処理可能にするため，ここではディスクアレイにおける Write Offloading(WO) [3] に似た機能を導入する．電源停止中で当該ノードへライトできない場合，あらかじめノードごとに定めた別のノードに対しログを書き込むことで，オブジェクトの冗長性を維持する．このログはノードを電源復帰する際に，そのノードのメモリ内データを最新の状態に更新するために用いる．また，電源停止中で当該ノードからリードできない場合，あらかじめ複製データのうちひとつを電源を停止しないノードにマイグレートしておくことで，複製へアクセスを転送しアクセス能力を維持する．このような機能を導入し，各オブジェクトの複製のうち最低 1 つを電源停止しないノードに配置することで，システムとして格納するすべてのデータへのアクセスを処理することが出来る．

4.3　電源停止時のタイムアウト抑制

このようなアクセス能力維持機能を分散データストアに導入するにあたり障害となるのが，電源停止ノードへのクライアント計算機からのアクセスがタイムアウトすることである．ディスクアレイでは，クライアントからのアクセスを受け付けるコントローラ部分の電源は維持されるため，コントローラが適切にログディスクを選びアクセスを処理することが出来る．一方，分散データストアではクライアント計算機が直接ノードにアクセスしてしまう．特に，大規模分散データストアではスケーラビリティのためクライアント計算機がシステム構成ノードのアドレスリストの非同期キャッシュを保持していることがある．例えば我々の DKVS ではクライアントがノードリストのキャッシュを保持する．この場合，クライアントは古いアドレスリストを利用してすでに電源停止したノードにアクセスしタイムアウトを待ってしまう．

この性能低下を回避するため，電源停止シーケンスの一部として，ネットワークの経路情報を変更し，タイムアウトを抑制する方法を導入する．提案手法では，ネットワーク経路情報を変更し，停止ノード宛パケットを即座にリジェクトするノードを作成する．

制御の概略は次のとおりである．電源停止するノード A と同じネットワーク内に複製や WO 対象など転送先となるノード B がある場合，ノード B がノード A のアドレスを詐称する．そして同じネットワーク内の他の計算機やルータのアドレス表を書き換え，ノード A 宛のパケットがノード B に届くように処理をする．この後ノード A の電源を停止する処理を行う．

また，ノード A とノード B が同じネットワーク内にない場合，中継ノード C を作成する．ノード A と同じネットワーク内の任意のノード C をノード A のアドレスに詐称する．ノード C はノード A 宛アクセスは処理できないため，パケットを拒否する返答をクライアントに返す．クライアントはアドレスリストを更新し，正しいターゲットであるノード B にアクセスをする．このような制御によりクライアントはタイムアウト無くデータにアクセスできる．

タイムアウト抑制手法の実装方法として，従来の TCP/IP を使ったネットワークでは Gratuitous ARP(GARP) パケットを使った制御により実現できる．GARP パケットはもともと同一ネットワーク内の IP アドレス重複確認に用いられる機能で RFC 5227 に記述がある．GARP パケットはまた，同一ネットワーク内機器の ARP テーブル強制更新にも用いられる．GARP によるアドレス付け替えは VM のマイグレーションとよく用いられる [19].

また，OpenFlow 技術 [20] などサーバ側から経路制御可能なネットワークではより簡単に実現することが出来る．前述の ARP による付け替えはネットワークをまたぐことが出来ず，中継ノードが必要であった．OpenFlow によるルーティングでは，ネットワークをまたいでパケットを転送することが出来るため中継ノードは必要ない．

4.4　ノード電源停止/復帰時のシーケンス

前節で示した，WO とタイムアウト抑制手法を併せた，ノード電源停止・復旧のシーケンスを図 2，3 に示す．

4.4.1　ノード停止時

図 2 では，まず電源を停止するノード（今回はノード A) が決定したら，停止ノードに対する WO ノード B を選定し，設定する．ノード B では，ノード A 格納オブジェクトへの更新内容を保持するログオブジェクトを作成する．つづいて，システム構成情報を変更し，ノード A に格納されたオブジェクトへの Read アクセスは複製保持ノードに，Write アクセスは WO ノード B へ転送するよう変更する．ここではまた，ノード A で処理中のアクセスの安全な終了や，データ一貫性等に注意しながら，アクセス先をノード B に振り分ける必要がある．同時にノード A では，以降のノード A 宛のアクセスをノード B に転送するよう設定する．

続いて，ノード A に対する中継ノード C を設定する．まずノード C では，ノード A のデータ転送用アドレスを詐称する設定を行う．つづいて，ノード A 宛のパケットに対しては即座にリジェクトするよう設定する．リジェクトは例えば ICMP の type3(Destination Unreachable) などで返すことが出来る．そして，ノード C は同一ネットワーク内の計算機に対し，以降のノード A 宛パケットをノード C に転送するよう，経路情報の変更依頼を送信する．図では GARP パケットのブロードキャストにより同一ネットワーク内の ARP テーブルを更新している．

ここで，ノード A がネットワーク経路情報変更後には切り離され，ノード停止命令を送付することが出来ない場合，図 2 に

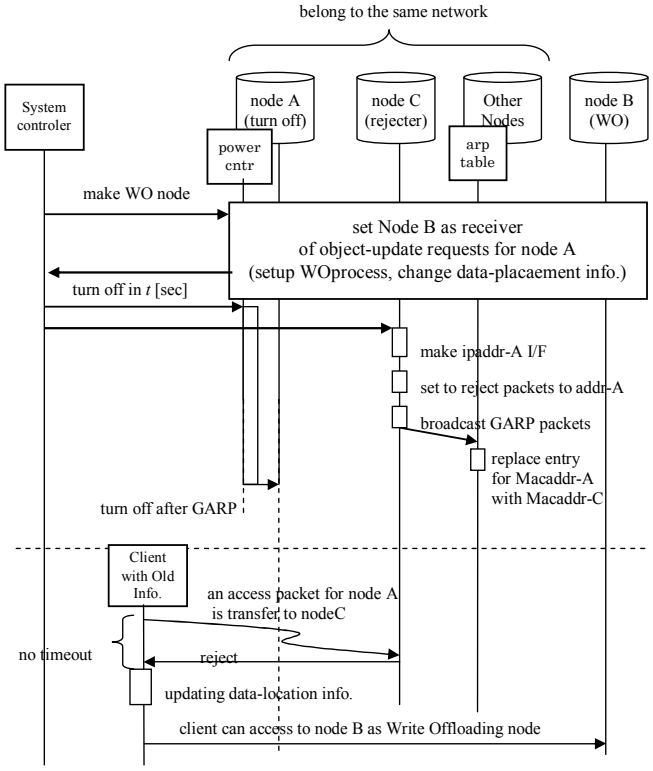

図 2　ノード電源停止のシーケンス

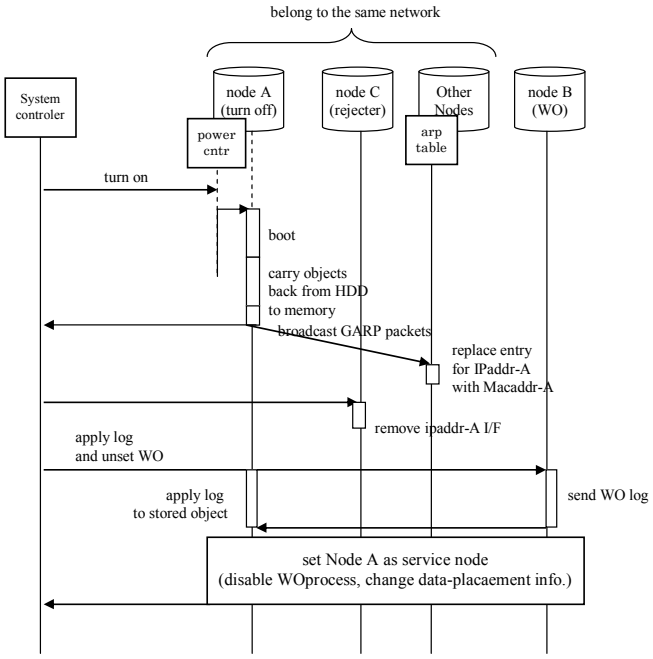

図 3　ノード電源復帰のシーケンス

示すように中継ノード C 設定前にあらかじめノード A に適切な時間 (図中 t) 後に発動するノード停止命令を送付する．ノード A が管理用のネットワークなど，経路情報変更後もアクセス可能な場合はノード停止命令はノード C 設定後でもよい．ノード停止命令を受けたノード A は ACPI などの実装を利用して，省電力モードへ移行する．

4.4.2　ノード復帰時

図 3 では，まず電源停止中ノードのうち復帰対象ノード（今回はノード A）を決定し，ノード A の電源をオンにする．ノード A では機材の起動処理が行われ，その後ハイバネーションであれば HDD に退避されたデータを主記憶中に復元する．ノード A は停止処理時にノード A 宛アクセスはノード B に転送するよう設定されているため，この設定を継続する．

ノード A はタイムアウト無くパケットを受けられる状態に復帰したため，中継ノード C の設定を解除する．このためにまず GARP パケットにより同じネットワーク内の ARP テーブルを更新し，ノード C のアドレス詐称設定やパケットリジェクト設定を解除する．

続いて，ノード A の WO ノードであるノード B から，ノード A に Write Offloading されたオブジェクト更新情報を転送し，ノード A 上のオブジェクトを最新にアップデートする．まだシステム構成情報としてはノード B はノード A の WO ノードであるため，このタイミングではオブジェクトの更新はノード B 上に到達する．性能を考慮する場合，これはノード B 上に書き込み，リクエスト処理終了後再びノード A 上に転送する必要がある．ノード B 上のログ量が十分小さい場合，ノード B 上のログを排他制御し，ログ適用後のノード A に要求を転送することが出来る．

最後に，ノード A に対するノード B の WO 機能を解除，システム構成情報を更新しノード A を通常モードに復帰する．実際には，ノード A 上の更新とシステム構成情報更新は一部並列に行うことが出来る．

表 2　評価システム諸元

ノード諸元

CPU	Intel Xeon L3110(3GHz, 2 コア)
Memory	8.0 GByte, DDR2-800, ECC
省電力モード	ACPI S4 (ハイバネーション), HDD に格納

システム諸元

ノード間ネットワーク	1000Base-T + TCP/IP
ノード数	16
オブジェクト複製数	3

5. 実　　験

ここでは，電源制御を行ったときのアクセス要求に対するレイテンシについて，実機上に構成したプロトタイプ・システムにより実験した結果を示す．プロトタイプは 3. で述べた DKVS の要件を元に構成した．本実験に用いたプロトタイプは開発中のものであり，本稿における提案事項の評価以外の機能の完全性を保証するものではない．

表 2 にシステム構成の諸元を示す．

3. で述べたストレージグループ導入によるアクセスレイテンシへの影響を調べるため，予備実験を行った結果を表 3 に示す．ここでは全ノードを 1SG, 2SG, 16SG の 3 構成を比較する．なお，オブジェクト複製数は 1 としている．この 64KB のオブジェクトを 12000 個格納し，100 ミリ秒ごとにひとつのオブジェクトの更新を行う負荷を各 1000 回ずつ，4 クライアン

表 3　ストレージグループ階層構造のアクセスレイテンシへの影響

構成 / 16nodes	1 SG	2 SG	16 SG
平均レイテンシ [$msec$]	5.01	4.97	4.99

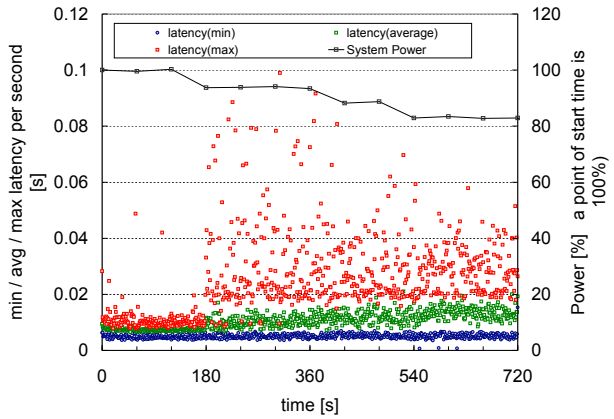

図 4　実験におけるシステム全体消費電力とアクセスレイテンシの推移

トプロセスから発生させた．この 1 アクセスは，ノード探索，オブジェクト取得，オブジェクト更新とコミットのアクセスを含む．図より，本実装では構成間のレイテンシ差は 1%以内であった．これは，各クライアントでのデータ配置情報キャッシュが有効に作用しているからと考えられる．以降の実験では 1SG 構成を基本とする．なお，スループットを考慮する場合，SG 数が少ないほど SGM が高負荷となるため，システムスケーラビリティは 1SG 構成が 16SG 構成より劣ると考えられる．

以降の実験では，あらかじめストレージグループ内のデータ配置を調整してある．実験で停止されないノードに最低１つのオブジェクト複製が存在するよう配置され，実験中オンラインデータマイグレーションは発生していない．

aa

5.1　実験 1: WriteOffloading によるアクセス性能維持

ここでは，WriteOffloading により各オブジェクトへのアクセスが維持される様子を観測する．ここでは予備実験と同様の負荷の下，1 台ずつ 3 分おきに，計 3 台のノードの電源停止処理を行った．

図 4 は実験におけるシステム全体の 1 分ごとの平均消費電力と，同じ実験におけるオブジェクトへの更新アクセスのレイテンシの推移をプロットしたものである．ここで電力は Raritan 社製のインテリジェント電源タップ Dominion PX を用い各サーバごとに測定・収集した値を合算したものである．測定誤差は電流・電圧それぞれに 5%であるため，電力値の誤差はおよそ 7%である．図から，ノードの電源を 1 台停止するごとに 1 ノードあたりの負荷が上がり，平均・最大レイテンシが上昇しているが一方システムとしてデータアクセスを継続できていることがわかる．3 台停止後の消費電力は 82.9%と，(16-3)/16=81.5%に近く，停止した台数に比例して電力が削減できることがわかる．電源停止中ノードの消費電力は主に管理用の BMC(ベースボード管理コントローラ) に拠る．

図中平均レイテンシより，電源停止の後も各クライアントは要求オブジェクトへのアクセスが行えていることがわかる．この更新負荷はノード探索，オブジェクト取得，オブジェクト更新とコミットのアクセスを含み，read 時の複製へのアクセス要求転送と write 時の Write Offloading のいずれも使用している．両機能を使わない場合，電源停止ノードに格納されたオブジェクトへのアクセスを冗長性低下無く行うためには，当該ノードを電源復帰させサービスに再参加させる必要があり，大きなレイテンシ悪化を伴う．両機能によりシステムはアクセス能力と電源制御を両立できる．

ただし，最大レイテンシは最初の電源停止後悪化している．この最大レイテンシ上昇の主な要因はログ書き込みの排他制御による．Write Offloading 時の書き込みを現在各ノードごとにひとつのログ用オブジェクトに記録しており，このオブジェクトの排他制御で待たされレイテンシが長大化する．この性能改善は今後の課題としたい．

5.2　実験 2: 電源停止時のタイムアウト抑制

前実験では最大レイテンシはどのタイミングでも 100msec 以下であり，タイムアウトを含む大きなレイテンシのアクセスは見られなかった．これは電源停止時のタイムアウト抑制が効果的に機能しているためである．ただし，懸念するタイムアウトは古いノード構成情報に従ったクライアントが存在する時のみ発生する．

次の実験ではタイムアウト抑制効果がより顕著な設定においてレイテンシを観測する．ここでは，各クライアントのノード構成情報について，ノード電源停止時に更新する機能をオフにした．そのため，各クライアントは古い情報を元に電源停止後のノードへアクセスし続ける．この状況の下，4.3 にて提案したタイムアウト抑制機能を有効/無効化し更新負荷を 4 クライアントから 200ms 間隔で合計 1600 回発生させた実験をそれぞれ行った．各クライアントは 1 リクエストごとにその終了を待ち，次のリクエストを送信する．そのため各リクエストでタイムアウトなどレイテンシが長大化すると実験の終了時間が延びる．

図 5 に結果を示す．実験では開始 30 秒後に 1 台ノード電源オフを行っている．提案機能無効（no control) では，ノード停止後，TCP のタイムアウトに起因する数秒から数十秒の大きなレイテンシが観測され続ける．そのため，わずか 1200 回のアクセスに 300 秒以上要している．一方，提案機能有効 (cntl for no-timeout) では，ノード停止後も実験 1 と同様わずかな性能低下のみで，タイムアウトは観測されなかった．実験も約 80 秒で終了している．この実験から，古いノード構成情報に従ったクライアントが存在しても，提案手法により大きな性能低下なくノード電源制御が行えることが確認できた．

6.　ま　と　め

高性能と省電力性を両立したインメモリ型分散データストアは，人と地球にやさしい情報社会の実現において重要な要素の１つである．本稿では，システム構成ノードの一部をシステム負荷に応じて動的に電源停止・復帰する省電力手法に着目し，インメモリ型分散データストアに対し，柔軟なデータ配置機構，

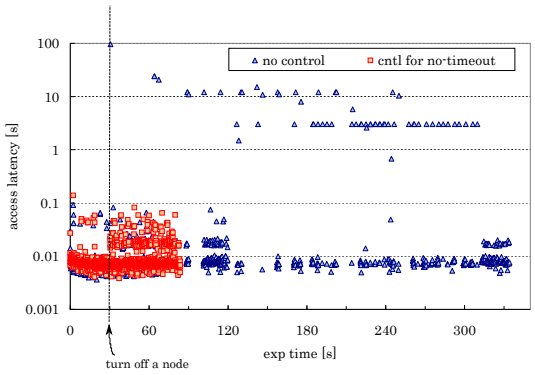

図 5　タイムアウト抑制制御有無でのアクセスレイテンシの違い

複製へのアクセス転送，他ノードへのライトオフローディング，タイムアウト抑制ネットワーク制御を導入した．これらの機能をインメモリ型分散データストアに実装することで，システムへのアクセス負荷の大小に応じて多くのノードを停止しつつ，各ノードが電源停止・復旧時にもアクセス性能を維持することができる．実験では我々の開発する DKVS を用いたプロトタイプシステムを用いて，電源停止ノード台数に比例した省電力効果と，アクセス性能の維持効果を確認した．

今後の課題としては，負荷の変動に合わせたデータ配置・ノード電源停止手法が挙げられる．4. で述べたように，大きな省電力効果のあるモードはモード移行時間が長いため，スループット性能について要求を満たしつつ最大限に電力削減するための制御アルゴリズムが必要である．

また，インメモリ型分散データストア本来の機能であるノード障害時の復旧機能と本稿で検討した省電力機能の融合や，実験で見られた省電力機能による性能低下に対する改善等が挙げられる．例えば，5. では，Write Offloading 用ログへのアクセスに起因する性能低下が見られた．オブジェクトアクセスの並行制御と電力制御を考慮した高速化は今後の課題である．

謝辞　本研究の一部は，独立行政法人新エネルギー・産業技術総合開発機構 (NEDO) の委託事業による成果である．プロトタイプ構築にあたり協力頂いた NEC ソフトウェア東北（株）肥田野氏らに感謝する．

文　　献

[1] R. Kallman, H. Kimura, J. Natkins, A. Pavlo, A. Rasin, S. Zdonik, E. P. C. Jones, S. Madden, M. Stonebraker, Y. Zhang, J. Hugg and D. J. Abadi: “H-store: a high-performance, distributed main memory transaction processing system”, Proc. VLDB Endow., **1**, pp. 1496–1499 (2008).

[2] J. K. Ousterhout, P. Agrawal, D. Erickson, C. Kozyrakis, J. Leverich, D. Mazières, S. Mitra, A. Narayanan, M. Rosenblum, S. M. Rumble, E. Stratmann and R. Stutsman: “The case for RAMClouds: Scalable high-performance storage entirely in DRAM”, SIGOPS Operating Systems Review, Vol. 43, pp. 92–105 (2009).

[3] D. Narayanan, A. Donnelly and A. Rowstron: “Write off-loading: practical power management for enterprise storage”, Proceedings of the 6th USENIX Conference on File and Storage Technologies, USENIX Association, pp. 17:1–17:15 (2008).

[4] 合田, Q. Wenyu, 喜連川：“”複数問合せ処理を意識したディスクストレージ省電力化に関する一考察””, DEWS2008: 第 19 回データ工学ワークショップ (2008).

[5] 引田, 横田：“プライマリ・バックアップ構成を有効利用したストレージシステムの省電力化手法の提案”, 第 2 回データ工学と情報マネジメントに関するフォーラム (DEIM 2010), pp. E6–4 (2010).

[6] D. Tsirogiannis, S. Harizopoulos and M. A. Shah: “Analyzing the energy efficiency of a database server”, Proc.of the 2010 international conference on Management of data (SIGMOD2010), pp. 231–242 (2010).

[7] 今田, 佐藤, 堀田, 木村：“分散型 web サーバにおけるノード状態制御による省電力化の検討”, 情報処理学会研究報告 HPC., **2007**, 88, pp. 55–60 (2007-09-09).

[8] 広渕, 小川, 中田, 伊藤, 関口：“仮想クラスタ遠隔ライブマイグレーションにおけるストレージアクセス最適化機構”, 情報処理学会研究報告 HPC, **2008**, 74, pp. 19–24 (2008-07-29).

[9] 菅, 小林, 鳥居, 小川, 板 橋, 宮田, 山川, 長谷部：“スケーラビリティと高効率性を備えたクラウド基盤を実現するデータセントリック分散制御”, DEIMForum 2010:第 2 回データ工学と情報マネジメントに関 するフォーラム, pp. C2–2 (2010).

[10] L. A. Barroso and U. Hölzle: “The case for energy-proportional computing”, Computer, **40**, pp. 33–37 (2007).

[11] D. Colarelli and D. Grunwald: “Massive arrays of idle disks for storage archives”, Proceedings of the 2002 ACM/IEEE conference on Supercomputing, Supercomputing ’02, Los Alamitos, CA, USA, IEEE Computer Society Press, pp. 1–11 (2002).

[12] J. Leverich and C. Kozyrakis: “On the Energy (In) efficiency of Hadoop Clusters”, HotPower ’09: Workshop on Power Aware Computing and Systems, Big Sky, MT (2009).

[13] W. Lang and J. M. Patel: “Energy Management for MapReduce Clusters”, Proceedings of the VLDB Endowment, **3**, 1 (2010).

[14] H. Amur, J. Cipar, V. Gupta, G. R. Ganger, M. A. Kozuch and K. Schwan: “Robust and flexible power-proportional storage”, International Conference on Management of Data, pp. 217–228 (2010).

[15] W. Lang, J. M. Patel and J. F. Naughton: “On energy management, load balancing and replication”, ACM SIGMOD Record, **38**, 4, p. 35 (2009).

[16] R. Kaushik and M. Bhandarkar: “GreenHDFS: Towards An Energy-Conserving, Storage-Efficient, Hybrid Hadoop Compute Cluster”, usenix.org (2010).

[17] 大野, 小林, 菅：“データインテンシブコンピューティングの省電力化に向けた gpu ノードの活用”, 電子情報通信学会技術研究報告 (CPSY), **110**, 167, pp. 1–6 (2010).

[18] Hewlett-Packard, Intel, Microsoft, Phoenix Technologies and Toshiba: “Advanced configuration and power interface specification, revision 4.0a” (2010).

[19] C. Clark, K. Fraser, S. Hand, J. G. Hansenyand, E. July, C. Limpach, I. Pratt and A. Warfield: “Live migration of virtual machines”, Proc. of NSDI’05 (2005).

[20] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner: “Openflow: Enabling innovation in campus networks”, SIGCOMM Comput. Commun., Vol. 38, pp. 69–74 (2008).