Vol.2013-ARC-207 No.24
Vol.2013-HPC-142 No.24
2013/12/17

# 分散Key-Value Storeを用いたインメモリデータベースのベンチマークと大規模計算機基盤の構築

田中 康司[1,a)]　原口 弘志[1,b)]　岩瀬 高博[2,c)]　藤井 秀明[1,d)]　泥谷 誠[1,e)]　岩爪 道昭[1,f)]

**概要**：ソーシャルネットワークサービス，各種センサなどから膨大なデータ（ビッグデータ）が生成されるようになり，これらビッグデータの蓄積・解析および有効活用に関心と期待が高まっている．独立行政法人情報通信研究機構（NICT）では，数十億ページ規模の Web アーカイブ構築を行っており，この膨大な数のデータを収集・蓄積するにはストレージ容量だけでなく，I/O 性能が非常に重要となってくる．また，M2M（Machine to Machine）センサネットワークから生成される各種センサデータの蓄積についても同様と言える．本稿では，各研究開発においてカギとなるデータベースについて，分散 Key-Value Store の “okuyama” に着目し，インメモリデータベースの I/O 性能のベンチマークを行った結果を報告する．また，得られた結果からビッグデータの蓄積・解析および有効活用に必要なインフラ設備について検討し，現在構築・整備中の大規模計算機基盤を紹介する．

## 1. はじめに

近年，ソーシャルネットワークサービスや各種センサが広く普及したことによって，短時間に膨大なデータ（いわゆる “ビッグデータ”）が生成されるようになってきた．これらビッグデータを蓄積・解析および有効活用することによって，既存サービスの改善や新しいサービスの創出につながる可能性が高いとして多くの分野において広く関心と期待が高まっている．

独立行政法人情報通信研究機構（以下，NICT）では，言語，知識，距離などの障壁を乗り越え，いつでも，どこでも，誰とでも意思疎通を可能とするユニバーサルコミュニケーションの実現に向け，多言語翻訳技術，音声コミュニケーション技術，情報分析技術，超臨場感コミュニケーション技術など高度な情報通信技術の研究開発に取り組んでいる．

これらの各研究課題では，大規模な言語コーパスや各種の言語資源，ネットワーク上の Web コンテンツやセンサデータから得られる多様かつ膨大な情報などを取り扱うことを前提としているが，各技術課題において個別にデータを蓄積・管理し，要素技術の研究やそれに基づくアプリケーションサービスを開発することは，コストがかかり非効率であるため，共通基盤的な大規模計算機基盤の構築・整備が不可欠である．このような計算機基盤では，アプリケーションによってはリアルタイム性の高い処理やそのための大量データの高速かつスケーラブルな蓄積・管理機構とアプリケーション側にデータを提供するための高速な I/O 性能が求められる．

各研究課題において，カギとなるのが “データベース” であるが，従来の関係データベースマネジメントシステム（以下，RDBMS）は，このような用途には必ずしも適しておらず，仮に実現しようとするとハードウェア，ソフトウェアともに高いコストが伴う．そこで，近年 NoSQL と呼ばれる新しいデータベース技術が登場し，注目されている [1]．NoSQL は，RDBMS のように関係モデルに基づく固定的なデータ構造ではなく，データや計算機資源の増加に応じてスケールアウトしやすいシンプルなデータ構造とシステムアーキテクチャを採用しており，ビックデータを高速かつ効率よく処理することが可能である．

本稿では，NoSQL の 1 つである分散 Key-Value Store（以下，分散 KVS）の “okuyama” [2] に着目し，インメモリデータベースの I/O 性能のベンチマークを行った結果を報告する．また，得られた結果からビッグデータの蓄積・解析および有効活用に必要なインフラ設備について検討し，

[1] 独立行政法人情報通信研究機構
National Institute of Information and Communications Technology
[2] 株式会社神戸デジタル・ラボ
Kobe Digital Labo, Inc.
a) tanaka@nict.go.jp
b) harah@nict.go.jp
c) iwase@kdl.co.jp
d) h-fujii@nict.go.jp
e) hijiya@nict.go.jp
f) iwazume@nict.go.jp

© 2013 Information Processing Society of Japan

Vol.2013-ARC-207 No.24
Vol.2013-HPC-142 No.24
2013/12/17

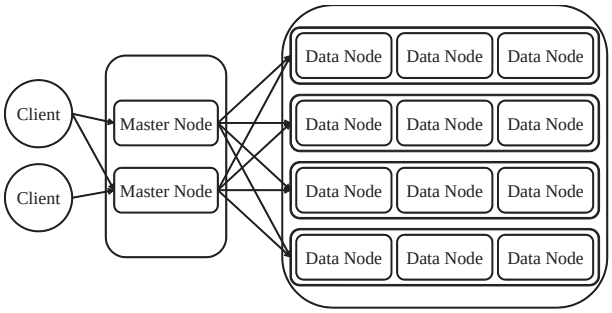


**図 1** okuyama の構成例

現在構築・整備中の大規模計算機基盤を紹介する．

## 2. okuyama 概要

okuyama はキー・バリュー型のデータ構造と分散型アーキテクチャを持つ NoSQL である．主に高速性，安定性，多様な用途での利用に優れた特性を発揮する．海外製の各種 NoSQL 製品が本格的に利用され始めた 2010 年にオープンソースの形式でリリースされ，その後，企業におけるビジネスユースでの利用などを経て現在も開発が継続されているプロダクトである．

okuyama は Java 言語で実装されており，マルチプラットフォームで稼働可能なデータベースである．以下，okuyama の機能概要を紹介する．

### 2.1 全体構成

okuyama は，データの保存・管理を担う「データノード」と，クライアントとデータノード間の接続中継を行う「マスターノード」から構成される．利用者はマスターノードに対してデータの検索・登録・削除を依頼し，マスターノードは依頼された操作をデータノードに指示する．各ノード間の通信は TCP/IP により行われる．また，データノードおよびマスターノードとも冗長構成を構築することが可能であり，障害時は自動的にフェイルオーバーが行われる．例えばデータノードを 3 冗長構成に設定した場合，同じデータが 3 つのノードに登録され，2 ノードまでの障害耐性を持つ．okuyama の構成例を**図 1** に示す．

### 2.2 データ構造

キー・バリュー型のデータ構造を採用しており，キーにより検索，更新などのデータ操作を行う．また，データ登録時にタグを付加することで，同じタグを付加されたデータを一括して取得することも可能である．

### 2.3 ストレージ

データノードは，データをメモリ上に保持するインメモリ型と，ディスク上に保持するディスク型のストレージを備えている．検索速度を担保する目的で，ディスク型においてキー・バリューのキーをメモリに保存する設定も可能である．また，データをメモリ上に保存する場合に圧縮することで，効率的にメモリを使用することも可能である．なお okuyama では，1 データ単位で圧縮せず，複数のデータを束ねた上で直列化し，バイナリ情報としてから圧縮を行うことで高圧縮効率が期待できるシリアライズマップ方式を採用する．

**表 1** 保存データのフォーマット

| | |
|---|---|
| キーデータ | 30 バイト固定長のユニーク値 |
| バリューデータ | 1,000 バイト固定長のユニーク値 |

### 2.4 スケーラビリティ

データノード／マスターノードともに無停止にて動的にノードを追加することができる．データの分散アルゴリズムとしては，Consistent Hashing[3] を採用している．また，ノードの追加処理中もすべての利用者からの操作に応答することが可能であり，利用規模の拡大に合わせてデータベース環境をスムーズに拡張することが可能である．

## 3. okuyama を用いた分散共有ストレージ／BigData in Memory

### 3.1 実験目的

現在，我々は大規模計算機基盤の実現を目指して，分散共有ストレージの構築を試みている．この分散共有ストレージに求められるものは，ビッグデータのようにサイズおよび数の点で大規模なデータを蓄積・管理し，複数のクライアントからの要求に応じてデータを高速に提供可能な性能である．そこで，Web アーカイブの URL を分散 KVS の okuyama で管理することを想定し，okuyama がこれらの性能を満たし得るかどうか実験を行い確認する．

### 3.2 実験内容

複数クライアントからの同時アクセス時の処理能力と大容量データの管理能力という 2 つの側面を検証するために，データノードのストレージ特性を変えて，40 億件余りのデータを用いた書き込みおよび読み込みテストを実施する．以下，実験で用いるデータのフォーマットとストレージ特性について説明する．

#### 3.2.1 データフォーマット

本実験で使用する保存データのフォーマットを**表 1** に示す．この条件下では，データ 1 件あたりのサイズはキーとバリューのペアで 1,030 バイトとなるため，40 億件分のデータサイズは 3.837 テラバイトとなる．さらにすべてのデータがレプリケーションデータを持つため，単純計算では 7.674 テラバイトのストレージ容量が必要となる．

#### 3.2.2 ストレージ特性

インメモリ型とディスク型の 2 パターンで実験を行う．

© 2013 Information Processing Society of Japan

**表 2** テストに利用した計算機のスペック

| | |
|---|---|
| CPU | Intel Xeon X5650 2.66GHz × 2（計 12 コア） |
| Memory | 72GB |
| HDD | 44TB (RAID6) |
| Network | 1GbE |

**表 3** okuyama の構成

| | |
|---|---|
| データノード | 80 台／ 200 okuyama データノード (※) |
| マスターノード | 5 台／ 5 okuyama マスターノード |
| クライアント | 2 台／ 200 クライアントスレッド |

※冗長化分を含めると 400 okuyama データノード

ディスク型では，キーをメモリ上に，バリューをディスク上に保存する構成とする．またメモリ上に保存されるデータは圧縮する設定とする．ディスク上のバリューに関しては，1 データを 4,096 バイトの固定長データとして扱い，ディスク上の 1 つのデータファイルにすべてのデータを格納し管理する．

### 3.3 実験方法

新規書き込みテストとランダムアクセステストの 2 パターン（計 4 ケース）を，インメモリ型とディスク型のストレージ特性ごとに実施する．

以下に各パターンの実験方法を説明する．

#### 3.3.1 新規書き込みテスト

okuyama データノードにデータが未登録（0 件）の状態から開始し，登録件数が 40 億件に到達するまでの処理時間を計測する．

#### 3.3.2 ランダムアクセステスト

- テストケース 1（ランダム書き込み）
  登録済みデータ数が 40 億件の状態（新規書き込みテストパターン終了時点）の okuyama に対して，200 の独立したクライアントスレッドよりランダムな値を登録し，5 分間で登録できた件数から性能を測定する．
- テストケース 2（ランダム読み込み）
  テストケース 1 終了後の状態に対して，200 の独立したクライアントスレッドよりランダムに登録済みの値を取得し，5 分間で取得できた件数から性能を測定する．
- テストケース 3（ランダム書き込み・読み込み同時）
  テストケース 1 終了後の状態に対して，200 の独立したクライアントスレッドよりランダムに読み書きを行い，5 分間に処理できた件数から性能を測定する．

### 3.4 実験環境

本実験の実施環境について以下に示す．

(1) 計算機環境
計算機は 87 台を利用した．計算機のスペックを**表 2** に示す．

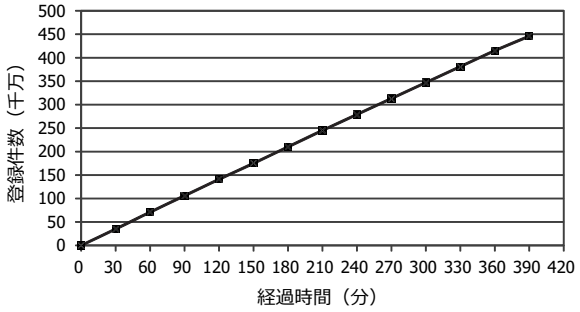


**図 2** メモリストレージでの新規値登録の結果グラフ

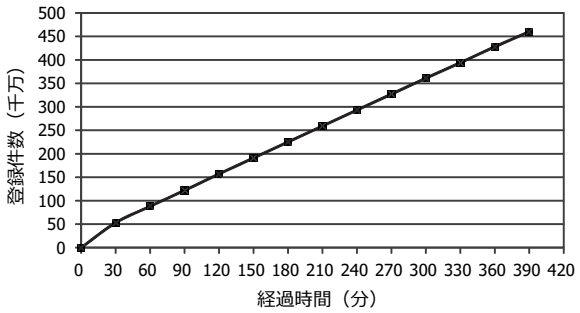


**図 3** ディスクストレージでの新規値登録の結果グラフ

(2) okuyama の構成
okuyama の構成を**表 3** に示す．

(3) okuyama の設定
データノードは 2 冗長化構成とする．したがって，レプリケーションデータを含めて okuyama 上に保存される実際の総データサイズは，登録されたデータの 2 倍となる．

(4) 実行環境の設定
実行環境はすべて Java7 を利用し，ガベージコレクションのアルゴリズムとして G1GC（Garbage-First Garbage Collector）[4] を指定する．メモリ割り当てに関しては，1 台のサーバ内に 5 つの okuyama データノードを起動し，それぞれ 12GB のメモリを割り当てる．

### 3.5 実験結果と考察

新規書き込みテストおよびランダムアクセステストの実験結果を以下に示し考察する．

#### 3.5.1 新規書き込みテスト

ストレージにインメモリ型を選択した場合のテスト結果を**図 2** に，ディスク型の場合を**図 3** に示す．

インメモリ型の場合，図 2 のテスト結果より 360 分で 40 億件のデータ登録が完了していることが分かる．また 30 分ごとに約 3.5 億件ずつデータ登録件数が増えており，その処理能力を最後まで維持している．

データ総量の計算から，本来，40 億件のデータではレプ

© 2013 Information Processing Society of Japan

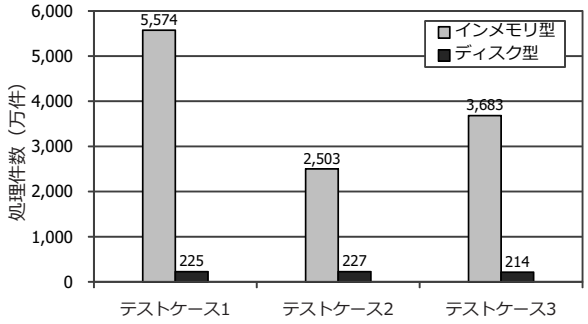


**図 4** ランダムアクセス時の結果グラフ

リケーションデータも含めると約 7.6 テラバイトのデータサイズとなる計算である．したがって，データノードに割り当てられているメモリ総量である 4.8 テラバイトを上回るサイズのデータを収容できていることが分かる．これはシリアライズマップによる圧縮処理が効率的に行われた結果であり，圧縮率はデータ内容に依存するため明確に断定はできないが，大容量データを管理する上で有益な技術であると言える．

一方ディスク型においても，図 3 のテスト結果より 360 分で 40 億件のデータ登録が完了している．また 30 分ごとに約 3.5 億件ずつデータ登録件数が増えており，その処理能力を最後まで維持している．

この結果より，インメモリ型とほぼ同等の書き込み性能であることが分かる．ディスクを利用しているにも関わらずインメモリ型と同等の処理能力が発揮された原因としては，ディスクアクセスのバッファリング機構や RAID コントローラによるキャッシングが奏功したと考えられるが，より詳細かつ正確な原因については今後の調査課題としたい．

#### 3.5.2 ランダムアクセステスト

ランダムアクセステストの結果を**図 4** に示す．

インメモリ型の場合，書き込みテスト（テストケース 1）での処理件数が読み込みテスト（テストケース 2）の 2 倍以上となっている．また，書き込み・読み込み同時テスト（テストケース 3）では，処理件数は両者（テストケース 1, 2）のほぼ中間の値となっている．このことから圧縮を行う書き込み処理に対して，復元を行う読み込み処理が低速なことが読み取れる．また 5 分間で約 5,500 万件の書き込み処理が行えていることから，30 分間換算にすると約 3.3 億件（毎秒約 18 万 3 千件）の登録ができることが分かる．これは新規書き込みテスト時の 30 分間あたり約 3.5 億件（毎秒約 19 万 4 千件）と比較して遜色のない処理性能であり，200 クライアントからの同時アクセスにもパフォーマンスを劣化させることなく対応可能な性能が確認できたと言える．

ディスク型の場合，新規登録時はインメモリ型と同等の処理能力が発揮されたが，ランダムアクセスを行った場合はインメモリ型の 10%以下の処理能力しか確認されなかった．これはディスクへのアクセスが発生した際に，ランダムアクセスであるが故にキャッシュヒット率が低く，常にディスクの参照を行う必要があるため性能が劣化しているものと思われる．

#### 3.5.3 考察

新規書き込みテストとランダムアクセステストの結果より，インメモリ型を使い圧縮と組み合わせることでテラバイト級のデータをメモリ上で管理できることが実証できた．またインメモリ型を選択することで，上書き処理やランダムアクセスが発生する状況下でも，ディスク環境下と比べて高い処理能力を発揮できることが判明した．

## 4. 大規模計算機基盤

前章では，Web アーカイブの URL を okuyama で管理することを想定したベンチマークを行った．今後，Web アーカイブが大規模になるにつれメモリ容量およびディスク容量の不足など，種々の問題が起こりうることは容易に想像できる．また，Web アーカイブだけでなく，M2M センサネットワークから生成される膨大かつ非構造化の多種多様なデータを蓄積・解析し，有効活用するためには，蓄積・解析するデータに適した大規模計算機基盤を構築・整備する必要がある．

本章では，まず構築・整備する大規模計算機基盤に要求されるスペックについて述べ，要求スペックを満たすべく構築・整備中の大規模計算機基盤について紹介する．

### 4.1 要求スペック

Web アーカイブのみならず，膨大かつ多種多様な M2M センサデータ（ビッグデータ）を蓄積・解析し，有効活用するために要求される計算機基盤のスペックは以下の通りである．

- 数 ~ 十 TB 規模のメモリ容量・Accelerator 付き共有メモリサーバ
  大規模グラフ解析などに代表される解析では，非常に大容量のメモリが必要とされる．また，分散メモリでの並列化が困難な計算や GPGPU などの Accelerator が有効な計算も存在する．
- 数百ノード規模かつ数十 TB 規模のローカルストレージの計算サーバ
  各種ビッグデータの解析には，多くのノード数かつ大容量のローカルストレージが必要な解析が数多く存在する．
- 高速なデータベースサーバ
  ビッグデータの蓄積・解析および有効活用には，高速なデータベースサーバが必要である．
- 数十 PB 規模の共有ストレージ
  ビッグデータの蓄積・解析に耐えうる大容量かつ高速

© 2013 Information Processing Society of Japan

表 4　計算システム機器概要

| System | Vendor | Nodes | CPU | Cores | Memory | Storage | Accelerator |
|---|---|---|---|---|---|---|---|
| 大規模共有メモリシステム | SGI | 1 | Xeon E5-4617 × 16 | 96 | 4TB | 50TB | |
| | SGI | 1 | Xeon E5-4617 × 48 | 288 | 12TB | 50TB | Phi 5110P × 32 |
| | SGI | 1 | Xeon E5-4617 × 56 | 336 | 14TB | 50TB | Tesla K20X × 8 |
| 大規模データシステム | DELL | 160 | Xeon E5-2667 × 2 | 12 | 256GB | 30TB | |
| | DELL | 160 | Xeon E5-2667 × 2 | 12 | 256GB | 45TB | |
| 高速データシステム | DELL | 10 | Xeon E5-2667 × 2 | 12 | 128GB | 3.2TB | |
| データベースシステム | DELL | 10 | Xeon E5-2667 × 2 | 12 | 256GB | 6.4TB | |
| 汎用計算システム | DELL | 70 | Xeon E5-2665 × 2 | 16 | 64GB | 6TB | |

表 5　ストレージ機器概要

| System | Vendor | Phisical | Detail |
|---|---|---|---|
| 共有ストレージ | DDN | 30PB | 4TB × 7,560 |
| バックアップ | DELL | 2PB | LTO6 Media × 409 |

表 6　ネットワーク機器概要

| Usage | Vendor | Ports |
|---|---|---|
| 内部接続用 | Mellanox | Infiniband FDR |
| LAN 接続用 | DELL | 40GbE (to Switches)<br>10GbE (to Nodes) |
| 監視用 | DELL | 10GbE (to Switches)<br>1GbE (to Nodes) |
| 基幹用 | Cisco | 40GbE/10GbE |
| ファイアウォール | PaloAlto | 10GbE/1GbE |

な共有ストレージが必要である．

- 高速なネットワーク
  大容量ファイルあるいは膨大な数のファイルへの高速なアクセスが必要である．

### 4.2　M2M データセンター設備

要求されるスペックを満たした大規模計算機基盤を「M2M データセンター設備」として現在構築・整備中で 2014 年 2 月末完成予定である．「M2M データセンター設備」は，コンテナ型データセンターとして構築・整備している．「M2M データセンター設備」における IT 機器とコンテナ型データセンターの概要を以下に紹介する．

#### 4.2.1　IT 機器概要

「M2M データセンター設備」における IT 機器のうち，計算システム機器，ストレージ機器，ネットワーク機器の概要をそれぞれ**表 4**，**表 5** および**表 6** に示す．

各 IT 機器の特色は以下の通りである．

大規模共有メモリシステムは，3 つのシステムから構成される SMP マシンで，4TB, 12TB, 14TB の大容量メモリを搭載している．また，12TB, 14TB のメモリが搭載されたシステムには，それぞれ Intel Xeon Phi 5110P と NVIDIA Tesla K20X が搭載されており，大規模なメモリ空間が必要とされるグラフ解析や共有メモリでの超並列計算および GPGPU などの Accelerator を利用した計算など各種計算に利用可能である．

図 5　M2M データセンター設備完成予想図

大規模データシステムは，合計 320 ノード構成であり，メンテナンス性の観点からブレード型かつディスクレスを採用した．ストレージは各ブレードに SAN 接続された外部ストレージを利用する．また，ブレードシャーシに仮想的な MAC アドレスを特定のブレードスロットに割り当て可能な機能があり，ブレード障害時にブレードを交換するだけでサービスが継続可能な特色がある．

高速データシステムおよびデータベースシステムは，ローカルストレージにおいて容量よりも I/O 性能を重視するような処理や高速なデータベースアクセスを実現するため，SSD を用いた RAID 構成によるローカルストレージを採用した．

共有ストレージは，Web アーカイブや各種 M2M センサデータの蓄積に耐えうるよう物理容量を 30PB とした．ファイルシステムには，GPFS（General Parallel File System）を採用し，各ノードには Infiniband FDR 経由でマウントする．これにより，高度なスケーラビリティとスループットおよびデータ共用性が実現できる．

また，解析済みデータの永続的なバックアップとして，2PB の容量を有した LTO6 テープロボットを用意した．

ネットワークは，表 6 にあるように，LAN 接続のスイッチ－スイッチ間は 40GbE，ノード－スイッチ間は 10GbE で接続し，高速アクセスが可能となっている．また，基幹用－ LAN 接続用間は，40GbE × 4（Bonding）で接続する．さらに，ネットワークはノード－監視用スイッチ間以外は二重化しており，障害時にも通信可能な構成とした．

#### 4.2.2　コンテナ型データセンター概要

「M2M データセンター設備」として必要とされる大規模計算機基盤を格納する方法として，コンテナ型データセン

© 2013 Information Processing Society of Japan

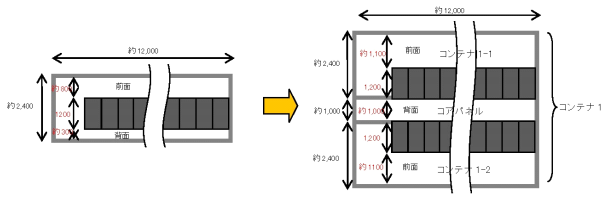


**図 6** コンテナの保守性比較

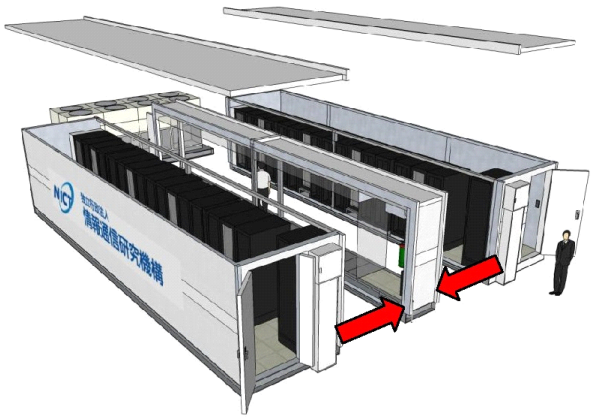

**図 7** コンテナ連結イメージ

ター設備を用いて構築・整備中である．**図 5** に M2M データセンター設備完成予想図を示す．

2011 年 3 月 25 日に国土交通省より提出された「コンテナ型データセンターに係る建築基準法の取り扱いについて」とされる書簡（国住指第 4933 号）[5] により，データサーバとしての機能を果たすために必要とされる最小限の設備は貯蔵槽に類するものとして，建築物対象より除外されたことを受け，日本国内では様々なコンテナ型データセンターが設置されてきており，NICT でも 2012 年 2 月にこの制度を利用したコンテナ型データセンターを設置し，運用を開始した．「M2M データセンター設備」の構築に際しても工期短縮の観点からコンテナ型データセンターの導入を決定した．

コンテナ型データセンターは建築物対象より除外されたため，2012 年 2 月に構築されたコンテナ型データセンターでは着工から完成まで約 2.5 ヶ月と短期間で竣工した．M2M データセンターのコンテナ型データセンターに関しても着工から竣工まで約 5 ヶ月の予定である．

今回導入するコンテナ型データセンターは，トレーラにより運搬可能な ISO 規格に準拠した 40 フィートコンテナ 2 機をコアパネルと呼ばれる接続パネルで接続し，その部分をホットアイルとし，IT 機器が搭載されるラック前面の空間を十分に確保することで ISO 規格コンテナ 1 機の場合よりも限られた空間を最大限利用でき，格段に保守性を高めている．コンテナの保守性を比較した図および連結イメージをそれぞれ**図 6** および**図 7** に示す．

また，冷却方法も IT ラック間に設置可能な局所的空調機を採用し，熱流体シミュレーションを行い最適な箇所に配置することで，ホットアイル側の熱を直接空調機が吸い込み，冷却効率を上げ PUE（Power Usage Effectiveness）1.3 以下を目指す．各コンテナへのアクセスは防犯面・利用面・管理面から非接触型 IC カードキーを採用した．コンテナの内外には監視カメラや赤外線センサを設置し，監視を行う．

また，IT 機器の電源として，IT 機器が収納されるコンテナ（IT コンテナ）とは別に無停電電源装置（UPS）専用のコンテナ設備（UPS コンテナ）も計画している．UPS コンテナには M2M データセンターに搭載されるすべての IT 機器を賄う容量の UPS が設置される．また UPS コンテナにはモジュール変換効率が 18%以上の高効率太陽電池モジュールと 15kWh のリチウムイオン蓄電池システムが導入され，各コンテナ内外の照明や防犯センサ，サービス電源などに利用される．

## 5. おわりに

本稿では，NoSQL の 1 つである分散 Key-Value Store の “okuyama” に着目し，インメモリデータベースの I/O 性能のベンチマークを行った結果を報告した．また，得られた結果から Web アーカイブのみならず，M2M センサネットワークから生成される膨大な各種センサデータ（ビッグデータ）の蓄積・解析および有効活用に必要なインフラ設備について検討し，現在構築・整備中の大規模計算機基盤（「M2M データセンター設備」）を紹介した．「M2M データセンター設備」完成後には，okuyama のより大規模なベンチマークを各種計算システム（大規模共有メモリ・大規模データ・高速データ・データベース）を用いて行い，多種多様なビッグデータの高速かつ効率的な処理を実現するための大規模計算機基盤はどうあるべきか検討する予定である．

### 参考文献

[1] Cattel, R.: Scalable SQL and NoSQL Data Stores, *ACM SIGMOD Record*, Vol. 39, Issue 4, pp. 12–27 (2010).
[2] okuyama: okuyama 公開サイト，`http://sourceforge.jp/projects/okuyama/` (2010).
[3] Karger, D., Lehman, E., Leighton, T., Levine, M., Lewin, D. and Panigrahy, R.: Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web, *Proc. of the Twenty-ninth Annual ACM Symposium on Theory of Computing (STOC '97)*, ACM, pp. 654–663 (1997).
[4] Detlefs, D.: A Hard Look at Hard Real-Time Garbage Collection, *Proc. of the 7th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2004)*, IEEE Computer Society, pp. 23–32 (2004).
[5] 国土交通省：「コンテナ型データセンターに係る建築基準法の取り扱いについて」，`http://www.mlit.go.jp/common/000138783.pdf` (2011).

© 2013 Information Processing Society of Japan