Vol.2010-IOT-8 No.16
2010/3/1

ramfs，iSCSI，LVMを用いた仮想サーバクラスタにおけるメモリ共有手法

永井　洋太郎[†1]　岡本　慶大[†1]　奥田　剛[†1]
河合　栄治[†2]　砂原　秀樹[†3,†1]

サーバの利用形態として，クラウドコンピューティングに代表されるデータセンタ集中型が注目されるにつれ，仮想計算機 (VM) 技術 を用いた仮想サーバクラスタが普及している．仮想サーバクラスタにおいては，VM が種々の計算機リソースを共有することでリソース利用効率が向上し，コスト削減のつながる．しかしながら，実計算機間でのリソース共有は十分に行われておらず，余剰リソースが存在する場合に効果的に利用できない状況が存在する．

本研究では，仮想サーバクラスタ環境で実計算機間のメモリリソース共有を可能にするシステムを提案する．VM 内でメモリ使用状況を監視し，余剰が発生すればエクスポートを行う．エクスポートされたデバイスは他の実計算機にホストされる VM からリモートスワップデバイスとして利用できる．本システムにより仮想サーバクラスタ全体でのメモリリソース利用率が向上する．

A method to share memories in virtual server cluster using ramfs, iSCSI and LVM

Yotaro Nagai,[†1] Yoshihiro Okamoto,[†1] Takeshi Okuda,[†1]
Eiji Kawai[†2] and Hideki Sunahara[†3,†1]

In this paper, we propose a method to share memories in virtual sever clusters that realize efficient memory usage across real machines. This method utilizes widely used technologies such as ramfs, LVM and iSCSI. We show this inexpensive method can improve performance in the case a guest OS have to use swap devices.

1. はじめに

昨今，仮想計算機（VM）技術を用いたサーバ多重化が盛んに行われるようになってきている．サーバ多重化を行うことで，CPU，メモリ，ストレージ，ネットワークなどのリソースが複数の VM で共有され，各リソースの利用効率が向上し，総合的なコスト削減が期待されるためである．特にデータセンタのクラウドコンピューティングのような利用方法が多重化されるサーバの数はますます増加し，コスト削減の効果も大きくなっている．複数の実計算機を協調的に利用する仮想サーバクラスタを用いて実現される中で，このような環境では，各実計算機に仮想計算機モニタ（VMM）が導入され，その上に VM（仮想サーバとして利用される）が展開される．ところで，VM として動作させる仮想サーバには様々な種類と用途があり，それに応じて必要とするリソースの種類や量が異なる．それぞれの仮想サーバが利用するリソースの種類と量について，事前にプロビジョニングを行うことで，仮想サーバクラスタにおける VM の配置が決定される．

しかし，プロビジョニングの想定を越える負荷が与えられた場合割り当てていた以上のリソースが必要となる場合がある．例えば，ある VM が想定されていた以上のメモリを要求した場合を考える．この場合，当該 VM がホストされている実計算機内に余剰のメモリが存在すればその領域を与えることがもっとも効率がよい．そのような余剰メモリがない場合，ライブマイグレーションを用いて，メモリリソースが潤沢に余っている実計算機に VM を移動することで，必要なリソースが確保でき，サービスを継続できる．このようなライブマイグレーションを用いたリソースマネジメントが現在の運用の主流である．

ライブマイグレーションに基づいたリソースマネジメントは VM の移動によって行われる．移動先の実計算機上において使用可能な CPU，メモリ等のリソースが揃って潤沢に存在しなければ VM の移動は不可能であるか，あるいは効果的でない．たとえばメモリが 1GB 余っているような実計算機が 2 台存在するとき，VM をどちらの実計算機に移動しても，使用できるメモリは 1GB が上限となる．既存のリソースマネジメント手法では，このように散在するリソースを利用することはできない．

†1 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
†2 情報通信研究機構
National Institute of Information and Communications Technology
†3 慶應義塾大学
Keio University

© 2010 Information Processing Society of Japan

現在，仮想化が行われるサーバによっては，時期・時刻によって，突発的に大量のリソースを求めるサーバなどが存在する．また，将来的には，科学計算などの用途で仮想計算機が利用されることも想定され，リソース要求量が巨大になることも考えられる．そのような場合に，仮想サーバクラスタ全体に存在している未使用リソースをなるべく有効に利用するために，CPU やメモリに関しても当該実計算機にとどまらない，実計算機間でのリソース共有を可能とし，より柔軟なリソース割り当てが行えるシステムが求められる．

本研究では，実計算機間でメモリリソースを共有可能な仮想サーバシステムを提案する．

本論文では，2 章で関連研究について述べ，3，4 章でシステムの提案，設計，実装を示す．さらに 5 章で評価実験の結果を示し，6 章で本研究の成果をまとめる．

2. 関 連 研 究

本章では，ライブマイグレーション関する関連技術について述べる．

2.1 ライブマイグレーション

本節では，Xen で実装されているライブマイグレーションの機能について概要を述べる．

Xen では，VM（Domain-U）を動作させたまま，別の実計算機に移動することができる．これをライブマイグレーションという．ライブマイグレーションを行うためには，前提として，移動元の実計算機と移動先の実計算機においてストレージが共有されていることが必要となる。

このような環境を準備すると，VM のメモリ内容を順次コピーすることで，VM の移動が実施できる．メモリー内容のコピー中も VM は稼働しており，クライアントへのサービスが継続される．

ライブマイグレーションの利用により，実計算機のメンテナンス時などにおいても VM の稼働を止めることなく，サービスが継続できるというメリットがある．

2.2 ライブマイグレーションを用いた運用について

ライブマイグレーションを用いることで，効果的なリソースマネジメントも行うことができる．

VMware 社の VMware DRS (Distributed Resource Scheduler)[1] は複数の実計算機間で，Live Migration を用いて VM のロードバランシングを行う機能である．実計算機はそれぞれがリソースプールの一部として登録され，リソースプール内で複数の VM を負荷に応じて自動的に再配置する．また，仮想サーバクラスタの稼働中に新たな実計算機をリソースプールに追加することも可能で，追加処理完了後に自動的に再配置が行われる．逆に，実計算機にメンテナンスが必要になった場合はメンテナンスモードへ遷移させることで，その実計算機上で動作していた VM は他の実計算機へ移動し，リソースプールから削除され，実計算機のシャットダウンが可能となる．配置はポリシベースで行うことも可能で，リソース利用の偏りなどから排他的に配置したい VM などを指定可能である．

従来，仮想サーバクラスタの稼動前に十分なプロビジョニングを行い，配置を決定していた手法に対して，実行時の負荷に応じて動的に VM の動作する実計算機を変更可能なため，プロビジョニングにかかるコストそのものや，運用管理コストを削減することが可能である．

しかし，ライブマイグレーションを実施するには，移動先に CPU，メモリなどのリソースがそれぞれ十分に存在しなければならない．例えば，メモリが少量だけ余っているような場合においては，その余剰リソースを利用することはできない．

2.3 リモートスワップに関する先行研究

ネットワークを介するリモートスワップデバイスを利用した研究として，DLM[2] や Teramem[3] 等が存在する．これらは，HPC（High Performance Computing）の分野において，メモリを大量に必要とする計算を行うために，リモートメモリをスワップデバイスとして使用することを目的としている．

どちらも OS のスワップ機構とは独立したスワップ機構を提供し，ディスクではなくリモートメモリをスワップデバイスとして使用するという特徴がある．これは，計算性能を改善したい特定のアプリケーションでのみリモートスワップを使用させたいという目的であることと，ディスクへのスワップを前提として設計されている OS のスワップ機構の最適化がリモートスワップにおいては性能的に不利になる[3] などが理由である．

3. リモートメモリを用いるリソース共有手法の提案と設計

本章では，VM 技術を用いたサーバ仮想化を実施したサーバが複数存在し，それらを協調させて利用する環境において，実計算機を越えたメモリリソースの共有を実現するシステムを提案する．まず，想定環境について整理し，要求事項をまとめる．その後でシステムの設計を行う．

3.1 想 定 環 境

本研究では，データセンタや企業のサーバルームなど，同一箇所に設置した複数台の実計算機に VM 技術を導入し，管理している環境を想定する．以降，このような環境を「仮想サーバクラスタ」と呼ぶ．

物理サーバ（ホスト OS）の管理者と各仮想サーバ（ゲスト OS）の管理者は異なる可能

© 2010 Information Processing Society of Japan

性がある．物理サーバ群が統一的に管理され，ライブマイグレーション機能を用いた仮想サーバの移動も行われている．

このような環境に規模の大きな計算を行う特殊な用途の仮想サーバを導入する．このサーバは一時的に大量のメモリリソースを使用するため，スワップデバイスをも頻繁に利用する．通常，スワップデバイスはローカルハードディスクであり，メインメモリと比較して数百分の一程度の帯域である．このため，スワップデバイスの帯域がボトルネックとなり，計算性能が著しく低下する．対象とするユーザはこのような特殊な用途の仮想サーバの管理者とする．当該管理者がリモートスワップデバイス使用要求を出した際に，他のいずれかの実計算機の余剰メモリへアクセスできることが求められる．

一方で，他の物理サーバ上の仮想サーバにおいて，Web，DB など一般的なサービスを行っているサーバは俊敏な反応が求められることから，スワップデバイスの使用が生じないように十分大きな容量のメモリリソースの割り当てられる．そのため，このような仮想サーバでは，メモリリソースの余剰が生じている時間帯が存在する可能性がある．そこで，VMM によってはメモリリソースのオーバコミットを認めている場合がある．オーバコミットは，各 VM に割り当てた合計のメモリー容量が実計算機が搭載する物理メモリー容量を越えることを認める機能をさす．VMware や Xen においては，バルーニングという仕組みがオーバコミットを実現している．オーバコミットを実現することで，同一実計算機上で動作してる VM 間では自 VM に割り当てられているメモリリソースを他 VM に利用させることが可能となる．しかしながら，同一実計算機上に，余剰メモリリソースを必要とする VM が存在しない場合には，この余剰分を有効に利用する手立てはない．

このような環境において，大量のメモリリソースを利用したい仮想サーバが他の物理サーバにおいて余剰となっているメモリリソースをスワップデバイスとして利用できるシステムを提案する．

3.2 要 求 定 義

3.1 節で述べた内容から，システムへの要求定義を行う．

まず，通常の仮想サーバにおいてメモリリソースの使用状況を監視する必要がある．取得した情報に基づきメモリの余剰・不足を判定する必要がある．余剰が発生した際には，これを他の物理サーバに利用させることができるシステムが必要となる．この際，適切な余剰判定を行わなければ，過剰にメモリリソースがエクスポートされ，貸し出されてしまう可能性がある．これにより，メモリ貸し出し側において本来であれば発生しなかったスラッシングが生じるなど，当該仮想サーバの性能を著しく低下させる可能性がある．このような自体を避けるために，余剰判定は慎重に行わなければならない．

次に，貸し出しを行っているメモリリソースは貸し出し元の要求に応じて，速やかに返還できるものでなければならない．これも貸し出し側サーバの性能劣化を防ぐためである．

また，メモリの貸し出し，返還の際に，メモリ利用側仮想サーバの OS の停止が不要であることが求められる．

これにより，要求は以下のように整理される．

(1) 仮想サーバにおいてメモリ使用状況を監視し，余剰判定が行える
(2) 余剰が生じたメモリは他の物理サーバに貸し出し，利用させることが可能である
(3) メモリリソース貸し出し側の要求に応じて，速やかに返還することが可能である
(4) メモリ貸し出し，返還の際には，メモリ利用側の OS を停止させる必要がない

3.3 設　　計

提案システムの設計について説明する．提案システムは以下の要素から構成される．

- メモリ借り入れ側モジュール
- メモリ貸し出し側モジュール
- リモートスワップ管理マネージャ

システム全体の動作イメージを図 1 に示す．この図を用いて，本システムの動作を説明する．

(1) メモリ貸し出し側モジュールの動作
メモリ貸し出し側においては，常に VM（ゲスト OS）内部でメモリ使用状況を監視する．監視結果は，当該 VM がホストされている実計算機の管理 OS 上で動作するメモリ貸し出し側モジュールに通知される．
メモリ貸し出し側モジュールでは，監視結果に応じて，他の実計算機にエクスポート可能な領域があれば「エクスポート可能であること，エクスポート可能な容量，iSCSI ターゲットの IQN」をリモートスワップ管理マネージャに対して一定間隔で通知する．この時点で，貸し出し側の iSCSI ターゲットは接続できる状態で準備されている．

(2) マネージャが搭載する管理 DB
メモリ貸し出し側モジュールからの通知を受けて，リモートスワップ管理マネージャは自らの持つ実計算機ごとの情報を記録したデータベース（以下，DB）を構築する．この DB に記録される情報は,「実計算機 IP アドレス，iSCSI ターゲットの IQN，エクスポート可能容量，エクスポートの可否,（エクスポート中であれば）エクスポート先実計算機の IP アドレス」である．

© 2010 Information Processing Society of Japan

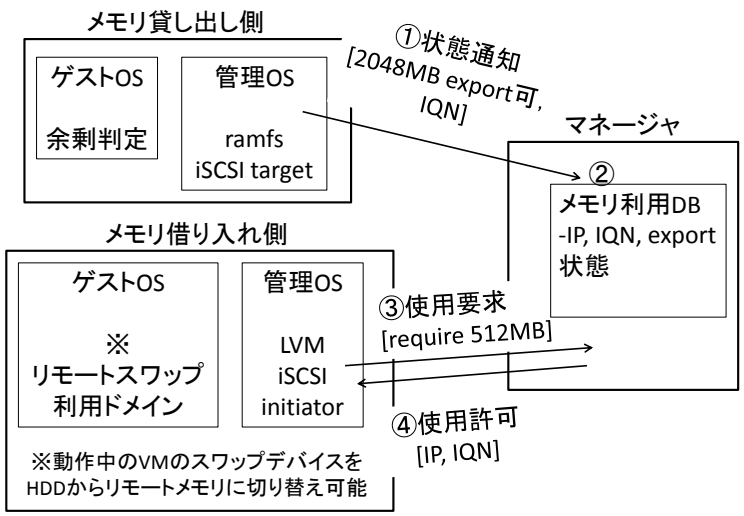

図 1 システムの概要

メモリ貸し出し側モジュールの一定間隔での通知によりこの DB は最新の状態に保たれる．この結果，リモートスワップ管理マネージャは仮想サーバクラスタ内の全実計算機のメモリ余剰状況とエクスポート状況を把握できる．

(3) 借り入れ側モジュールのリモートメモリ利用要求

必要に応じて，ユーザ（ゲスト OS 管理者）は借り入れ側モジュールを通じてマネージャに対してリモートスワップデバイス使用要求を送信する．この際，利用したい容量を通知する．

これを受けて，マネージャは DB より利用可能なリモートスワップデバイスをもつ実計算機を検索する．該当するものがあれば，利用許可を発行する．

(4) 借り入れ側モジュールに対するリモートメモリ利用許可

リモートスワップデバイスの利用許可が発行された場合，マネージャは借り入れ側モジュールに対して「実計算機 IP アドレス，iSCSI ターゲットの IQN，エクスポート可能容量」を送信する．

この情報を受け取った借り入れ側モジュールは iSCSI イニシエータでの接続を開始する．その後，LVM（Logical Volume Manager）の設定を変更することで，使用要求を出したユーザの管理するゲストのスワップデバイスがリモートメモリに切り替わる．

なお，図 1 では，貸し出し側，借り入れ側はそれぞれ 1 台，起動している Domain-U もそれぞれ 1 台であるが，実際にはともに複数の場合にも対応する．

4. 実　　装

3.3 節では，図 1 のすべてのモジュールについての設計を述べたが，本論文では，借り入れ側部分のみを評価対象とする．このため，図 1 の ② ~ ④ に関する機能のみを実装した．

実装対象は，Linux2.6（openSUSE 11.1）＋ Xen 3.3.0 とした．ゲスト OS もすべて，準仮想化カーネルの openSUSE 11.1 とした．

4.1 メモリ貸し出し側

メモリ貸し出し側についてはモジュールの実装をせず，固定の環境とした．Domain-0（Xen の管理 OS）において，ramfs を用いてメインメモリ上に一定容量のファイルシステムを構築した．ramfs は，Linux 2.4.0 以降で実装されているインメモリのファイルシステムである．ramfs は，古典的な RAM ディスク機構と異なり，メインメモリ上に疑似ブロックデバイスを作成しない．

構築したファイルシステムをマウントし，そこに空白のイメージファイルを作成した．このイメージファイルを iSCSI ターゲットに設定し，iSCSI イニシエータから接続可能な状態にしてある．また，メモリ貸し出し側の実計算機は 1 台とした．

4.2 リモートスワップ管理マネージャ

マネージャについては DB とメモリ借り入れ側モジュールとの通信部のみを実装した．

今回はメモリ貸し出し側を固定環境にしているので，DB の内容は固定とし，接続可能状態になっている貸し出し側の IP アドレス，iSCSI の IQN を書き入れた．

また，借り入れ側との通信部については，”want_iscsi”という文字列をリモートスワップデバイスの使用要求と定義した．メモリ借り入れ側モジュールから”want_iscsi”という文字列が送信されたことを検出すると，マネージャは DB より利用可能なデバイスが検索される．（この場合，先ほど DB に内容を書き入れた実計算機が発見される）

これに基づき，マネージャは”ok,*ipaddr*, *iqn*”という文字列を借り入れ側モジュールに対して送信する．

4.3 メモリ借り入れ側

メモリ借り入れ側モジュールについては，次のような環境の元で，必要な機能を実装した．

© 2010 Information Processing Society of Japan

4.3.1 リモートスワップデバイスを利用する VM のスワップデバイス

Xen においては，VM が利用可能なストレージとしてローカルディスクであれば，イメージファイル単位かパーティション単位で与えることができる．今回は，スワップデバイス（VM 内で動作する Linux でマウントされるデバイス）専用に LVM の LV（Logical Volume）で構築されたパーティションを渡した．この LV を内包する VG（Volume Group）には，標準状態としてローカルディスクの 1 パーティションを PV（Phisical Volume）として割り当てる．このため，通常時は VM のスワップデバイスとしてマウンさトれる LV の実体はローカルディスク上に存在することになる．

リモートメモリが利用できる際には，リモートメモリ（iSCSI イニシエータにより SCSI デバイスとして見える）を PV としてこの VG に追加する．この状態で，当該 LV をローカルディスクからリモートメモリに PV 移動すれば VM が利用するスワップデバイスはローカルディスクからリモートメモリに切り替わる．

なお，PV 移動はマウント中の LV についても可能であるため，VM の動作を止めることなく，また VM がこの変化を検知することもなく，スワップデバイスが切り替わる．

4.3.2 リモートスワップの要求から接続，スワップデバイスの切り替え

本節ではメモリ借り入れ側モジュールの実装について述べる．なお，本モジュールは C 言語で実装されているが，iSCSI で接続を行う箇所と LVM の設定を行う箇所に関しては，外部に用意したシェルスクリプトを呼び出している．

まずマネージャに対して，”want_iscsi”という文字列を送信することでリモートスワップデバイスの利用要求を行う．本ツールはこれに対する応答の通信を待つ．

マネージャから利用不許可あるいは利用許可の返答がくる．利用許可があった場合には，同時に iSCSI の接続情報となる IP アドレスと IQN が送られてくる．この情報を iscsiadm（iSCSI イニシエータの管理コマンド）を実行するシェルスクリプトに渡す．これにより，iSCSI の接続が確立される．

iSCSI の接続が確立されると，リモートメモリは SCSI のディスクとして認識される．(例えば，/dev/sdb のように認識される.)

4.3.1 節で述べたように，このデバイスをリモートメモリを利用する VM のスワップデバイスと置き換える．このために，LVM の操作を行うためのシェルスクリプトを記述した．このシェルスクリプトでは，pvcreate コマンドを用いて，マウントした iSCSI デバイスを LVM の PV として設定する．さらに vgextend コマンドを用いて，設定した PV を当該 VG に追加する．最後に pvmove コマンドを用いて，VM がスワップデバイスとしてマウントしている LV をリモートメモリに移動する．

以上により，VM が利用するスワップデバイスがローカルディスクからリモートメモリに切り替えられる．

表 1　5.2 節の測定環境

	メモリサーバ	メモリクライアント
CPU	Intel Xeon X3350	Intel Core2 Quad Q9450
Memory	8GB	1GB
NIC	Intel 10Gigabit AT2 Server Adaptor	Intel 10Gigabit AT2 Server Adaptor
OS	openSUSE 11.1(Native)	openSUSE 11.1(Native)

5. 評　　価

本章では，提案システムのうち借り入れ側モジュールの評価を行った．

5.1 評 価 手 法

メモリを大量に利用した性能測定のために，独自のマイクロベンチマークツールを作成した．本ツールではメモリ空間に指定したデータサイズ分の配列を確保する．確保した領域を初期化した後，連続読み込み（sequential read），連続書き込み（sequential write），ランダム読み込み（random read），ランダム書き込み（random write）のいずれかを選択して行う．読み込み，書き込みのブロックサイズも指定可能である．結果は単位時間あたりの読み込み，書き込みサイズ（スループット，MB/s）で出力される．

5.2 リモートスワップデバイスの基本特性

ramfs と iSCSI を組み合わせたリモートスワップの性能特性の評価をおこなった．

5.3 測 定 環 境

測定環境を表 1 に示す．以下で行った実験では，メモリサーバに 4GB の ramfs を構築した．その上に 4GB のイメージファイルを測定し，iSCSI ターゲットとして，エクスポートする．メモリクライアントは，iSCSI イニシエータを動作させて，メモリサーバ上のメモリを SCSI デバイスとしてマウントする．メモリクライアントは，実メモリを 1GB 使用可能である。スワップデバイスとして，ローカルディスク (4GB)，リモートメモリ (4GB) を切り替えて使用し，測定を行った．

なお，メモリサーバとクライアントはケーブルで直接接続した．

5.4 ブロックサイズに関する性能評価

データサイズを 1GB に固定して，ブロックサイズに関する性能評価を行った結果が，図

© 2010 Information Processing Society of Japan

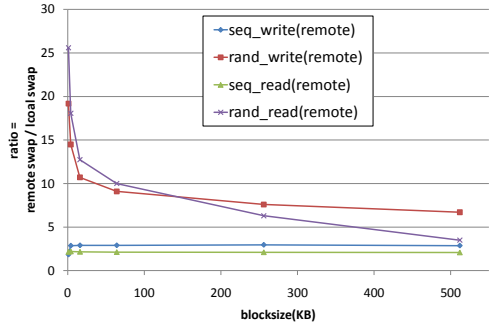

図 2 ブロックサイズによるスループット比

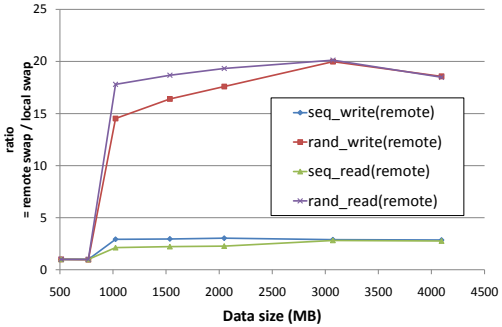

図 3 データサイズによるスループット比

2 である．縦軸は，ローカルスワップに対するリモートスワップのスループット比を表す．ランダムアクセスにおいては，ブロックサイズが小さい場合には特にスループット比が大きくなる．一方で，連続アクセスの場合にはブロックサイズの変化がスループット比に影響しない．これは連続アクセスの場合，ローカルディスクへのアクセスが十分速くなるためであると考えられる．

5.5 データサイズに関する性能評価

データサイズに関する性能評価を行った結果が，図 3 である．この実験では，ブロックサイズを 1KB に固定して，データサイズを変化させた場合の性能特性を調査した．データサイズは 512MB から 4GB まで変化させた．本実験環境では，実メモリ搭載量は 1GB であるので，実際にベンチマークを行うプロセスが使用できる実メモリは 900MB 程度であり，それを超えるデータサイズを指定した場合にスラッシングが起こり，性能が劣化する．連続読み込みにおいては，リモートメモリを使用するとローカルディスクした場合の 3 倍の性能向上がみられた．また，ランダムひょみこみにおいては，極端な性能劣化がみられるが，15〜20 倍の性能向上がみられた．

6. おわりに

本論文では，仮想サーバクラスタにおける，実計算機間でのメモリリソース共有手法について述べた．ramfs, iSCSI, LVM を組み合わせたリモートメモリの利用により他の実計算機における余剰メモリを有効に利用できることを確認した．

今後，提案システム全体の実装をすすめ，余剰メモリ判定アルゴリズムの検討や，大規模クラスタを用いた評価を行う．

参 考 文 献

1) VMware Distributed Resource Scheduler. http://www.vmware.com/products/drs/.
2) 緑川博子, 黒川原佳, 姫野龍太郎. 遠隔メモリを利用する分散大容量メモリシステム DLM の設計と 10GbEthernet における初期性能評価. 情報処理学会論文誌コンピューティングシステム, Vol.1, No.3, 2008.
3) 山本和典, 石川裕. テラスケールコンピューティングのための遠隔スワップシステム Teramem. *SACSIS 2009*, 2009.

© 2010 Information Processing Society of Japan