

ソフトウェア FT 環境の応答性向上にむけた CPU コア割当て方法

渡邊 和樹[†] 佐藤 隆佑[†] 金城 聖[†] 鶴 薫[†]

[†]三菱電機株式会社 情報技術総合研究所

1 はじめに

近年、PC サーバの高性能化と仮想化技術の普及に伴うサーバ集約が進んでいる。これにより、ハードウェア障害がサービスへ与える影響が増大している。そのため、ハードウェア障害に対応できる高可用システム構築技術が求められる。

既存の高可用システムには、FT サーバに代表されるハードウェアによるフォールトトレラントシステムがある。これは、サーバコンポーネント多重化と系の同期により、障害時の処理継続を実現するものである。しかし、サーバの調達が特定のベンダに限られることになる。

ベンダ依存を回避し、高可用システムを構築する手段として、仮想化技術で多重化を実現するソフトウェア FT が注目されている。仮想マシンモニタ (VMM) で複数の仮想計算機 (VM) を同期し、専用ハードウェアを用いない多重化を実現する。

一方、実際にソフトウェア FT を用いるにあたって、下記の課題が判明した。

- 1) 複数の VM が計算機資源を共有することで、ある VM の負荷が他の VM の性能に影響する
- 2) VM の動作に対して、同期処理が加わることで、ゲスト OS の応答性能が低下する

そこで、オープンソースのソフトウェア FT である kemari [1] とディスク同期ソフトウェアである DRBD [2] に着目し、適用環境の応答性能を計測した。結果、割込応答時間の増大が確認されたため、CPU 資源の固定割当てによる解決を試みた。

以下、2章で kemari と DRBD の概要と課題を述べ、3章で CPU 資源の固定的割当てについて述べる。次に4章で評価実験について述べ、5章で結果と考察を述べる。最後に6章でまとめる。

2 KVM/kemari と DRBD

2.1 概要

KVM [3] は、Linux に実装された VMM であり、I/O エミュレーションに QEMU [4] を利用する。kemari は QEMU に対し、VM 間のメモリ同期機能を提供する。KVM/kemari とディスク同期ソフトウェアである DRBD の組合せにより、専用ハード

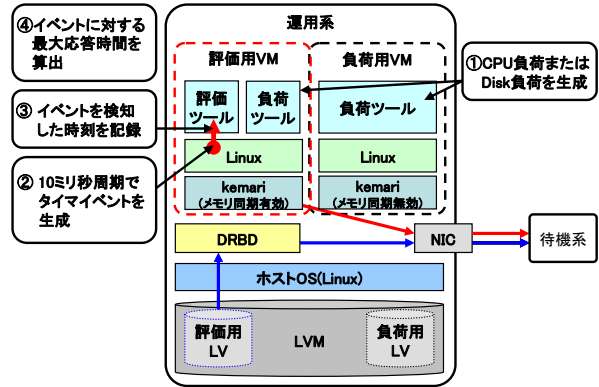


図1 実験概要

ウェアを用いない多重化が可能となる。

2.2 課題

KVM は、VM へ割り当てる CPU 資源を Linux のスケジューラで決定する。スケジューラは、割り当てる CPU 資源をプロセスの実行時間を基に決定する。その際、複数 VM に同一 CPU 資源が割り当てられ、性能が低下する可能性がある。また、kemari や DRBD も同様にスケジューラされるため、VM が用いる CPU 資源が競合し、ゲスト OS の応答性能が低下する可能性がある。この解決には CPU 資源の競合を避ける機構が必要になる。

3 CPU資源の固定割当て

CPU 資源の競合を回避するためには、各プロセスに固定的に割り当てるのが有効であると考えられる。CPU 資源の固定割当てには、Linux で提供される taskset コマンドと isolcpus オプションを用いる。taskset はプロセスに割当て可能な CPU のリスト (CPU アフィニティ) を操作するコマンドである。また、isolcpus はデフォルトの CPU アフィニティから除外する CPU を指定できる。isolcpus で除外した CPU を taskset で指定することで、プロセスに特定の CPU を固定的に割り当て、資源競合を回避できる。

これにより、各 VM の負荷に関わらず安定した性能が維持できると考えられる。

4 評価実験

4.1 実験概要

CPU 資源競合による性能低下の確認、及び CPU 資源の割当てパターンにおける性能差を比較する目的で評価実験を行った。実験環境を表 1 に示し、実験の手順を図 1 と以下に示す。

- ① 性能を計測する評価用 VM と、外乱を生成

Assignment method of CPU resource to improve responsiveness of software FT.

Kazuki WATANABE[†], Ryusuke SATO[†], Akira KANASIRO[†] and Kaoru TSURU[†]

[†]Information Technology R&D Center, Mitsubishi Electric Corporation.

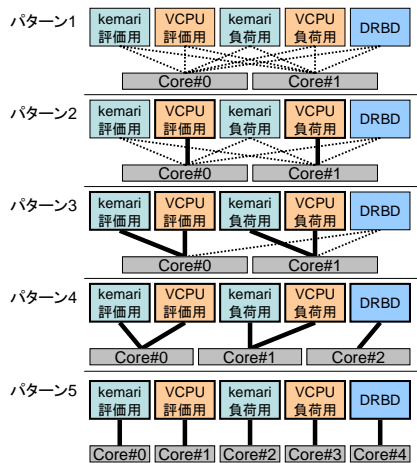


図2 コア割当てパターン

表1 実験環境

環境	ホスト	ゲスト
CPU	Xeon X5650 (6コア 2CPU)	VCPU 1個
RAM	12GB	512MB
NIC	NetXen NX3031	なし
HDD	SAS 80GB	LVM LV 10GB
OS	CentOS 6.2 x86_64 (kernel 2.6.32)	
VMM	QEMU 1.0 / kemari-next	
CPU 負荷	Dhrystone 2.1	
I/O 負荷	IOZone 3.283	

する負荷用 VM を用意し、いずれかの VM で CPU 負荷または I/O 負荷を生成する。

- ② 評価用 VM 上のアプリケーション(RT 優先度で実行)に対し、10000 回のタイムイベントを周期的に発生させる。
- ③ タイムイベントを検知した時刻を計測し、最大応答時間を算出する。

4.2 コア割当てパターン

CPU コアの割当てパターンを図2 と以下に示す。

- パターン1: コア割当てなし
- パターン2: VCPU に 1 個
- パターン3: VCPU と kemari に 1 個
- パターン4: VCPU と kemari に 1 コア, DRBD に 1 コア
- パターン5: VCPU, kemari, DRBD に 1 コアずつ

5 実験結果

実験結果を表2 に示す。この結果より考察できる内容を以下に示す。

■性能低下の原因 コア割当てをしない場合、負荷なし状態の最大応答時間は約 25.8 ミリ秒 (I), 評価用 VM で I/O 負荷を生成した状態で約 343.6 ミリ秒となり (II), 性能低下が確認できた。

一方、負荷用 VM で I/O 負荷を生成した場合、約 17.8 ミリ秒となり (III), 性能が向上した。これは、kemari が I/O を契機に同期する仕様であり、負荷用 VM の I/O 処理中、評価用 VM の I/O 要求が保留され、同期回数が減少したためである

コア割当てパターン	負荷なし	負荷用 VM		評価用 VM	
		CPU 負荷	I/O 負荷	CPU 負荷	I/O 負荷
パターン1	(I)25.786	(IV)24.005	(III)17.750	46.698	(II)343.605
パターン2	23.151	18.327	13.625	6.914	245.887
パターン3	14.229	7.924	10.015	12.108	(VI)145.647
パターン4	13.750	8.675	9.209	7.554	(V) 56.474
パターン5	14.193	21.597	42.891	15.309	(VII)419.046

表2 実験結果

ると考えられる。

また、負荷用 VM で CPU 負荷を生成した場合、約 24.0 ミリ秒となり (IV)性能差は見られない。

以上より、性能低下は他の VM ではなく、同期処理との競合による影響が大きいと判断できる。

■コア割当ての効果 応答性能が最も低下した、評価用 VM で I/O 負荷を生成した場合に着目すると、VCPU と kemari に同一コアを、DRBD に異なるコアを割り当てた場合に約 56.5 ミリ秒となり (V), 最も性能が良いことが分かる。これは、DRBD は VM の動作を同期の契機とせず、ゲスト OS と並列に動作が可能のためである。

また、コアと VM 数が同じパターンでは、VCPU と kemari にコアを割り当てた場合に約 145.6 ミリ秒となり (VI), 最も良い性能となった。これは、DRBD 以外にコアが割り当てられ、kemari 同士の競合が回避されたためと考えられる。

■コア割当てによる性能低下 評価用 VM へ I/O 負荷を生成し、VCPU, DRBD, kemari 全てに異なるコアを割り当てた場合では約 419.0 ミリ秒となり (VII), コア割当てしない場合より性能が低下した。これは、メモリ同期の際にゲスト OS が停止する kemari の仕様により、コア割当てによる効果が得られなかったためと考えられる。また、VCPU と kemari が別のコアで動作するため、ゲスト OS に対する割込や排他制御の際、CPU 間通信が発生し、性能が低下したと考えられる。

6 おわりに

本稿では、ソフトウェア FT の課題である資源競合による性能低下に対し、CPU コアを固定的に割り当てることで性能改善を試みた。その結果、最大割込応答時間が 343.6 ミリ秒から 56.5 ミリ秒に短縮され、性能改善を確認できた。

参考文献

- [1] 大村 圭ら : KVM を利用した耐故障クラスタリング技術の開発, 情報処理学会研究報告, Vol.2010-OS-115, No.20, pp.1-8 (2010).
- [2] LINBIT Information Technologies : DRBD, <http://www.drbd.org/> (2012/12/19 閲覧).
- [3] Kivity, A., et al.: kvm: the Linux virtual machine monitor, OLS'07, pp.225-230 (2007).
- [4] Bellard, F.: QEMU, a fast and portable dynamic translator, ATEC'05, pp.41-46 (2005).