

BladeSymphony ファームウェアの開発（４） Virtage スケジューラのマルチプロセッサ制御

松本 周平^{†1}長谷川 智英^{†2}(株)日立製作所 ITプラットフォーム事業本部^{†1}同 ITプラットフォーム事業本部^{†2}

1 はじめに

仮想化技術の発展・普及が進み、PC サーバ上位機や UNIX サーバ上位機を大規模サーバに統合する要望が出てきている。日立では、サーバブレード間 SMP 接続機能を有する BladeSymphony BS2000 と Virtage を用いてこの要望に応えている。しかしながら物理 CPU コア数が最大 80 個となる SMP 接続においては論理 CPU コア数増加に対する性能のスケラビリティ確保が課題となった。本論文では、Virtage のスケジューラの基本的な方式と前記課題を解決するスケラビリティ改善の内容を示す。

2 Virtage のスケジューラ

Virtage は、論理 CPU のスケジューリングモードとして占有モードと共有モードをサポートする。占有モードは、LPAR の論理 CPU を特定の物理 CPU に占有的に割り当てる方式である。これに対して、共有モードは、サービス率と呼ぶ CPU 資源の相対的な配分と実際の使用量に基づいて、物理 CPU を LPAR とその論理 CPU の間で時分割共有する方式である。以降では、共有モードについて述べる。

Virtage のスケジューラの基本的なデータ構造に実行キューがある。実行キューは、実行可能な論理 CPU を格納する待ち行列である。Virtage は実行キューを複数の物理 CPU で共有するグローバル実行キューを用いてきた。仮想化のスケジューラでは、割り当て配分の比率制御の、物理 CPU の効率的利用、及び同じ LPAR の論理 CPU の並列実行が非常に重要である。グローバル実行キューを用いた Virtage のスケジューラはこれらの要件を自然に満たすことができる。

(1) 割り当て配分の比率制御

まず、Virtage ではサービス率を LPAR ごとに持たせるので、LPAR 上のゲスト OS がどの論理 CPU を使うかを予め細かく考慮することが不要である。

このサービス率を例えば、LPAR1:LPAR2 = 100:300 と与えた場合、1 分間の実行時間の比率が LPAR1:LPAR2 = 100:300 でも、LPAR1 が最初の 15

秒間、LPAR2 が残りの 45 秒間実行されたのでは、問題がある。Virtage のスケジューラは、LPAR のスケジューリング優先度を使用量/サービス率とし、優先度が最小の LPAR の論理 CPU をディスパッチすることで、どの時間間隔でも実行時間の比率を LPAR1:LPAR2=100:300 とすることができる。

更に、スケジューラは、グローバル実行キューを用いることで、負荷を全ての物理 CPU で共有するので、複数の物理 CPU を 1 つの大きな CPU とみなすことができ、物理 CPU 数に依存せずに、上述の比率を保証することができる。

(2) 物理 CPU の効率的利用

スケジューラは、グローバル実行キューを用いて負荷を複数の CPU で共有するので、負荷変動に対して、特別な負荷分散の必要なしに、物理 CPU を偏りなく均等に使うことができる。

(3) LPAR の論理 CPU の並列実行

OS は一般に全ての CPU が同じ比率で実行されることを想定しているので、LPAR の複数の論理 CPU が同時に並列に実行されないと、大幅な性能低下等の問題を引き起こす可能性がある。Virtage のスケジューラは、自然に LPAR の複数の論理 CPU の並列実行を実現できる。

Virtage のスケジューラは、スケジューリング優先度の更新を論理 CPU の実行開始時ではなく、実行終了時に行うことで、ある時点では全ての物理 CPU が同時に同じ LPAR の優先度が最高だと判断する。Virtage はグローバル実行キューを用いるので、複数の物理 CPU が同時にスケジューラを呼び出すと、同じ LPAR の論理 CPU が順にディスパッチされて、並行に動く。

3 スケラビリティ改善

BladeSymphony BS2000 のサーバブレード間 SMP 接続機能でソケット数を増やし、そこに統合する LPAR 数と論理 CPU 数も増やすと、複数の物理 CPU で同時にスケジューラを呼び出す機会が増える。

実行キューを物理 CPU ごとのローカル実行キューにすれば、ディスパッチの競合は発生しないが、Virtage のスケジューラの望ましい特性が失われて

しまう。そこで、スケジューラの特性とオーバーヘッドのトレードオフと、最近のマルチコアプロセッサでは大容量キャッシュが CPU のソケット単位にあることを考慮して、実行キューを CPU ソケットごとに設け、CPU ソケット内の物理 CPU で実行キューを共有するように修正することにした。合わせて、実行キューを複数の物理 CPU で共有する場合のディスパッチ処理の競合を更に削減した。

(1) ディスパッチ処理の競合の削減

従来のディスパッチ処理は、広く知られた方法 [1] と同様に、ディスパッチ処理専用のスタックを持たず、切り替え元の論理 CPU の実行キューへの挿入、切り替え先の論理 CPU の実行キューからの取り出し、切り替え元と切り替え先の論理 CPU の間のコンテキスト切り替えの順で行っていた。この制御方式をグローバル実行キューに適用すると、デッドロックに陥る可能性があった。Virtage では対策はとっていたがスケーラビリティには不利となっていた。

今回、物理 CPU ごとにディスパッチ処理専用のスタックを持たせ、これを利用して、切り替え元論理 CPU と切り替え先論理 CPU の間のコンテキスト切り替えを、切り替え元論理 CPU のコンテキスト保存と切り替え先論理 CPU のコンテキスト復元の 2 つの処理に分け、ディスパッチ処理を、切り替え元論理 CPU のコンテキスト保存、実行キューへの挿入、切り替え先論理 CPU の実行キューからの取り出し、コンテキスト復元の順に行うようにした。これにより、従来のデッドロックの問題を、スケーラビリティを損なうことなく解決することができる。

(2) マルチ実行キューと負荷分散の導入

Virtage は、1 つ以上の物理 CPU をグループ化して、LPAR 群が使用する物理 CPU を制限する CPU グループ機能を持つ。この機能を用いればスケジューラのキュー操作を減少することができるので、スケーラビリティを向上できる。しかし、論理 CPU 数の多い LPAR を作れない、手動の負荷分散が必要、など構成の制約が増加する。

そこで、サブグループと呼ぶ内部的な CPU グループを CPU ソケット単位で作り、実行キューをサブグループごとに設け、1 つの LPAR の複数の論理 CPU を異なるサブグループに所属可能にする。論理 CPU のディスパッチはサブグループ間で独立に動作するため、Virtage のスケジューラは、サブグループ間の負荷分散を行う必要がある。Virtage のスケジューラは、サブグループの負荷を定期的に算出して、その平均値を求め、平均値とのずれを元に論理 CPU をサブグループ間で移動させて、負荷を調整する。物理 CPU の空きは、各 LPAR が望むだけ CPU 資源をもらえていることを意味するので、空きがあるときは、負荷の指標に論理 CPU の

走行時間情報を用いる。空きが少なくなると、サービス率の比率制御が必要になるので、このときは、負荷の指標にサービス率の重みと論理 CPU の走行時間の両方を考慮した情報を用いる。

4 評価結果

アイドル LPAR の統合はスケーラビリティ評価の重要なベンチマークの一つである。本論文では、サーバブレード間 SMP 接続機能を用いてブレードを 4 枚接続して 8CPU ソケット構成 (物理 CPU 80 コア) にし、Virtage 上で論理 CPUx16 の LPAR を 60 個つくり、22 個の Windows と 38 個の Linux を順不同にブートして、定常状態の CPU 使用量を評価した。比較として従来の Virtage と CPU ソケットごとに CPU グループをつくった場合にも同様の事を行った。

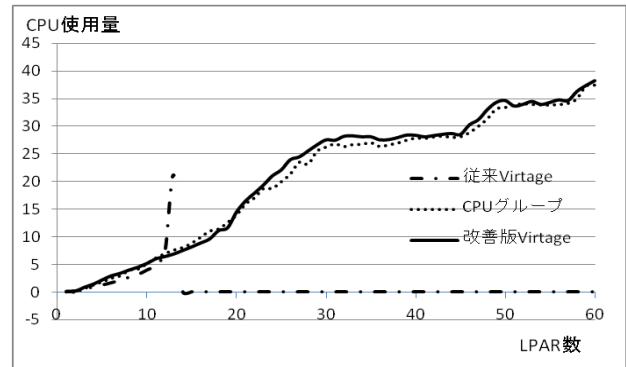


図1 LPAR 数と CPU 使用量(比率)

従来方式では 10LPAR を超えたところで CPU 使用量の限界となり、それ以上の LPAR 追加ができなかった。改善版では、CPU グループ作成時とほぼ同等の CPU 使用量にまで抑えることができた。

5 おわりに

Virtage スケジューラのスケーラビリティを大幅に向上させることができた。今後は更なるスケーラビリティの向上に加えて、占有モードの性能向上等に取り組んでいきたい。

参考文献

- [1] U. Vahalia, UNIX Internals - The New frontiers, Prentice Hall, 1996
- [2] VMware vSphere™ 4: The CPU Scheduler in VMware ESX™ 4, VMware, 2010
- [3] C.A. Waldspurger, Lottery and Stride Scheduling: Flexible Proportional-Share Resource Management. Ph.D thesis, MIT/LCS/TR-667, 1995
- [4] G.W. Dunlap, Scheduler development update, Xen summit Asia at Intel, 2009