

Scale-Out Processors

著者： P. Lotfi-Kamran, B. Grot, M Ferdman, S. Volos, O. Kocberber, J. Picore, A. Adileh *et al.*

出典： *Proc. of the 39th Annual International Symposium on Computer Architecture*, pp.500-511, 2012.

発表者： 高性能コンピューティング学講座 1353028 朴 傑

1 はじめに

データセンターでは、各サーバで使うプロセッサのスループット性能を向上させることでシステム性能の最大化を狙う。データセンターで使われてきた従来の汎用プロセッサやタイルプロセッサは大容量の LLC (Last-Level Cache) を搭載してもそれを有効に利用できず、LLC へのアクセスレイテンシが大きいという問題点がある。そのためデータセンターで実行されるスケールアウトアプリケーションに対してアーキテクチャがマッチしない。

本研究では、チップの有効面積や電力バジェットを考慮し、スケールアウトアプリケーション向けに LLC 容量やコア数を最適化した、スケールアウトサーバに適するプロセッサの設計方法論を提案する。

2 スケールアウトアプリケーション

スケールアウトアプリケーションとは、データセンターで動作し、ネットワークを介して、ユーザーが利用できるアプリケーションである。その特徴は以下のようなものである。

1. 並列度が非常に高い
2. 巨大な命令ワーキングセットを持っている
3. 各ノードはタスクの状態を共有しない
4. 各ノードは完全に独立した要求に対してサービスを提供する
5. ノード間接続は基本的に必要としない

3 Scale-Out Processors

現在サーバで使われている従来の汎用プロセッサやタイルプロセッサは、多くのコアや大容量の LLC をチップ上に集積することで性能向上を実現している。しかし、単一チップの構造を複雑化し、高性能化することは、スケールアウトアプリケーションに対しては有効とはならない。

以降ではスケールアウトアプリケーション向けのサーバ用チップである Scale-Out Processors を提案する。

3.1 性能密度

従来の研究ではチップ全体の性能に着目し、それを向上させることに主眼が置かれていた。しかし、図1のチップ全体の性能変化と単位面積あたりの性能の変化を見ると、コア数が増えるほどチップ全体の性能は向上するが、単位面積あたりの性能はコア数を増やし過ぎると逆に下がってしまう。特に、スケールアウトアプリケーションは

並列性が高く、多数のチップを用いる意ことでその分性能を向上するため、チップの単位面積あたりの性能が重要となる。そのため本研究では、チップの単位面積 (単位: mm^2) 当たりの性能である「性能密度」の観点から設計法論の構築や評価を行う。

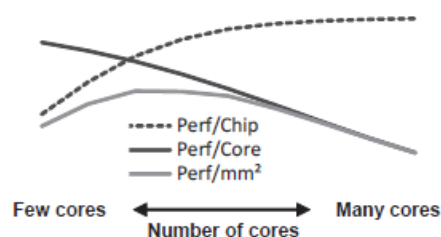


図1: チップ全体の性能変化、および性能密度の変化

3.2 チップ設計

チップ設計の過程を図2に示す。スケールアウトアプリケーションは大容量の LLC を搭載しても、それを有効活用できていない。逆に必要以上の LLC を搭載すると性能低下の原因となる。そのため、まずは命令ワーキングセットを保持できる程度の LLC 容量を設定し、その分より多くのコアを実装することで、スループットを高めることを狙う (図2の真ん中)。

LLC 容量が適切に設定された後、コア数の最適化を考える。コア数が少な過ぎると並列性を活かしきれず、またコア数が多過ぎても、性能がそれに比例して向上するとは限らないため性能密度が悪化してしまう。またコア数の増加に伴い、接続の配線長が増加し、アクセスレイテンシも増えてしまう。与えられた LLC 容量にあわせて、性能密度が最大になるように、コア数の最適化を行う必要がある (図2の右)。



図2: LLC 容量の最適化、およびコア数最適化

3.3 Pod 構造の導入

上述の LLC 容量やコア数を持つプロセッサ構成ではコア数が比較的に少なく、メモリバンド幅などチップ外部のハードウェアリソースや電力バジェットを有効に利用できない可能性がある。

そこでリソースを有効利用するため、ここまでの最適構造を Pod と定義し、リソースにまだ余裕がある限り、同一チップに複数の Pod を集積する。図 3 に Pod の構成とチップ上に複数 Pod を搭載した例を示す。複数の Pod を搭載しても個々の Pod の最適構成には影響しない。個々の Pod は完全かつ独立なサーバであり、それぞれが独立した OS やソフトウェアスタックを実行する。また Pod 間の接続は必要としない。

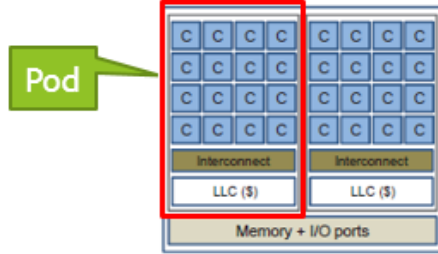


図 3: Pod 構造、および多重 Pod 構成

4 評価実験

4.1 実験環境

本実験において、使用するハードウェアの面積、および消費電力バジェットを図 4 に示す。

Component		Area	Power
Cores	Conventional	25mm ²	11W
	OoO	4.5mm ²	1W
	In-order	1.3mm ²	0.48W
LLC	16-way SA	5mm ² per MB	1W per MB
Component		Area	Power
Interconnect		0.2 - 4.5 mm ²	<5W
DDR3 interface (PHY+ controller)		(2 + 10) mm ²	5.7W
SoC components		42mm ²	5W

図 4: 使用するハードウェアの面積、消費電力バジェット

チップ全体の面積は 250~280mm² とし、電圧は 0.9V、電力予算は 95W、周波数は 2GHz と設定する。スケールアウトアプリケーションは CloudSuite 1.0[1] 中の 6 種類を使用する。それらは Solaris 10 オペレーティングシステム上で動作し、サイクル精度のフルシステムシミュレータである Flexus を用いて評価する。

評価では、汎用プロセッサ、タイルプロセッサおよび Scale-Out Processors の性能密度について比較評価を行う。汎用プロセッサとしては性能が比較的高いコアを使用し、タイルプロセッサと Scale-Out Processors については Out-of-Order Core (OOO)、および In-Order Core をそれぞれ使った場合を評価する。プロセステクノロジーは 40nm と 20nm の場合について評価を行った。

4.2 40nm テクノロジー、OOO 使用の場合

LLC の容量を 1MB~8MB に変更しつつ、Out-of Order Core 使用時の性能密度について 6 種類のアプリケーションの平均値を取った結果を図 5 に示す。図中の Ideal Crossbar はコアと LLC 間のアクセスレイテンシを固定したクロスバー構造で、Crossbar はコアと LLC 間の距離を考慮し、

それによってレイテンシが変わる、より現実なクロスバー構造である。また Mesh はメッシュ型ネットワーク構造を示している。

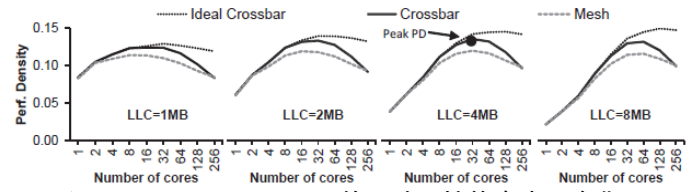


図 5: Out-of Order Core 使用時の性能密度の変化

図 5 より、32 コア、4MB の LLC 容量の場合が最も性能密度が高いが、32 コア前後では性能密度の変化が相対的に少ないことがわかる。コヒーレンシの複雑さ、多数のクロスバー相互接続を実現する難しさがあるため、コア数が少ない 16 コア、4MB の LLC 容量の構造が最適 Pod 構造と考えられる。この場合一つのチップ上に 2 つの Pod の設置が可能で、汎用プロセッサに対して 3.4 倍、タイルプロセッサに対して 1.6 倍の性能密度を達成できる。

4.3 40nm テクノロジー、In-Order Core 使用の場合

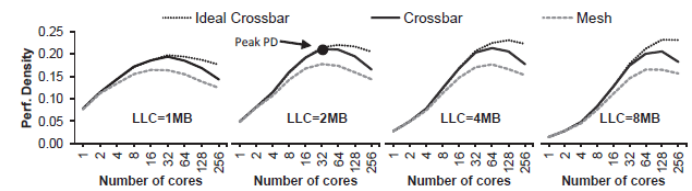


図 6: In-Order Core 使用時の性能密度の変化

次に、In-Order Core 使用時の結果を図 6 に示す。最適な Pod 構造として 32 コア、2MB の LLC 容量が最適な構成として選択され、一つのチップ上に 3 つの Pod が設置可能である。比較した結果、汎用プロセッサの 5.8 倍、タイルプロセッサの 1.5 倍の性能密度を達成した。

4.4 20nm テクノロジーの場合

20nm テクノロジー採用時は、40nm を採用する時よりも提案プロセッサと汎用プロセッサ、およびタイルプロセッサとの性能密度の差が広くなり、テクノロジー進歩に伴い、提案したプロセッサの優越性が広がることが確認できた。

5 おわりに

本研究では、スケールアウトサーバに適するプロセッサの設計方法論を提案した。

シミュレーションの結果、提案した設計法で設計したプロセッサは Out-of Order Core、In-Order Core それぞれを使用する場合において、汎用プロセッサ、およびタイルプロセッサより性能密度が優れていることがわかった。

参考文献

- [1] CloudSuite 1.0 <http://parsa.epfl.ch/cloudsuite>