1H-03

アプリケーション実行性能の類似性を用いた計算資源選択手法の提案

真鍋 優† 小板 隆浩†

†同志社大学理工学部情報システムデザイン学科

1. はじめに

計算量が日々増大する科学技術計算ワークフローを実行するにあたり，商用クラウドは経済的に優れた計算資源として期待されている.しかし，科学技術計算ワークフローを低い利用コストで実行できる計算資源を判断する方法が無いため，科学者は多くの計算資源から選択することが困難であることが問題となる．本研究では商用クラウドの一つであるAmazon EC2[1]において，アプリケーション実行性能の類似性に着目し，実行時間と利用コストを考慮した資源選択手法の実現を目指す.

科学技術計算ワークフロー[2]を実行するにあたり，科学者は自ら所有するPCクラスタやOpen Science Grid などのグリッドコンピューティングを利用してきた．しかし，ハードウェアの発展による処理能力の向上により，科学技術計算ワークフローが扱うデータ量は日々増大しており，求められる計算資源の処理能力やストレージ容量も同様に増大している.

そこで注目されているのが計算資源として商用クラウドを利用する事である．商用クラウドでは多様な性能の計算資源やストレージを迅速に追加する事が可能であり，データ量の増大に合わせて柔軟に対応できる．また，利用した時間のみ料金を支払うという従量制課金であるということなどから，利用コストの面でも期待されている.

2. 商用クラウドの課題

商用クラウドにおいて科学技術計算ワークフローを実行する課題となるのが次の4点である.

1) 仮想化によるオーバーヘッド
2) 共有・並列ファイルシステムの欠如
3) 低いネットワーク性能
4) 計算資源の利用コストが不明

これらの課題により，商用クラウドはHPC同等の性能を実現することができないのが現状である[2]．1から3の課題は商用クラウドの発展によって，ハードウェアの面から解決される事が期待されている．しかし，実際に科学者が商用クラウドを用いる際に発生する4の課題は解決されておらず[3]，科学者は計算資源を選択する際，自らの知識と経験を用いる必要がある.

本研究では実行時間を考慮した上で，利用コストについての課題に着目する．科学技術計算ワークフローを商用クラウドにおいて実行するにあたって，実行時間を大きく増加させることなく，利用コストの低い計算資源を選択する手法の実現が本研究の目的となる.

3. 提案手法

目的を実現するため，アプリケーションの実行性能の類似性を用いた資源選択手法を提案する．アプリケーションの必要とする性能が同等であるとき，実行時間が短く，利用コストの低い計算資源は同一であると考えられる．本提案手法は，事前に実行したアプリケーションから必要とする性能が類似するものを選択することで，対象のアプリケーションを短い実行時間，低い利用コストで実行可能な計算資源を選択する．本手法の手順を以下に示す.

1)サンプルアプリケーションの実行性能を計測
2)対象アプリケーションの実行性能を計測
3)対象アプリケーションと類似した実行性能を持つサンプルアプリケーションを選択
4)計算資源の選択

本提案手法において用いるサンプルアプリケーション，対象アプリケーションおよび実行性能について述べる．サンプルアプリケーションとはあらかじめ実行しておく，処理の異なる複数のアプリケーションである．対象アプリケーションとは実行する計算資源を選択する，本提案手法を適用する対象のアプリケーションである．実行性能とはCPU・メモリ・I/O性能による実行時間と利用コストへの影響を表し，それぞれを実行性能値，コスト性能値とする．例としてCPUによる実行性能値とコスト性能値を求める

Resource Provisioning Method using Performance Similarity of Applications
†Doshisha University

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.

式を示す．メモリあるいはI/Oによる実行性能値とコスト性能値は，それぞれECUの値をメモリ容量，ストレージ容量とすることで以下の式により算出する．

$$T = time \div \frac{1}{ECU}$$

$$C = cost \div \frac{1}{ECU}$$

手順1においてサンプルアプリケーションを実行し，最短の実行時間，最低の利用コストで実行可能な計算資源をそれぞれ選択する．また，任意に選択した1つの計算資源において実行性能を計測する．

手順2では対象アプリケーションを手順1において選択した計算資源において実行する．その際に実行性能を記録する．

手順3では手順1，2において記録した実行性能の距離を類似度として算出する．T_{Ae}，T_{Am}，T_{Ai}は，アプリケーションAのCPU・メモリ・I/Oそれぞれの正規化した実行性能値とする．T_{Be}，T_{Bm}，T_{Bi}は，同様にアプリケーションBの実行性能値とする．類似度D_{AB}の計算式は次の通りである．

$$D_{AB} = \sqrt{(T_{Ae} - T_{Be})^2 + (T_{Am} - T_{Bm})^2 + (T_{Ai} - T_{Bi})^2}$$

類似度が最小のサンプルアプリケーションを対象アプリケーションと最も類似しているものとする．

最後の手順4では，類似するサンプルアプリケーションにおいて手順1で選択した計算資源を，対象アプリケーションの最短の実行時間および最低の利用コストで実行できるインスタンスとして選択する．

4. 評価実験

商用クラウドの一つであるAmazon EC2において実験を行う．Amazon EC2の計算資源（インスタンス）を用い，複数のアプリケーションの実行時間と利用コスト，類似度を計測する．類似するアプリケーションの最短の実行時間，最低の利用コストで実行可能なインスタンスが一致すれば資源選択は成功となる．使用したインスタンスを表1に示す．

表1　インスタンス性能表

	vCPU	ECU	Memory	Storage	$/h
m4.large	2	6.5	8	8	0.139
m4.xlarge	4	13	16	8	0.278
c4.large	2	8	3.75	8	0.126
c4.xlarge	4	16	7.5	8	0.252
r3.large	2	6.5	15	32	0.2
r3.xlarge	4	13	30.5	80	0.399

本実験で実行するアプリケーションとして総合ベンチマークであるSysbenchおよびUnixBench，分散フレームワークであるhadoopを利用した．

5. 結果と考察

各アプリケーションを実行し，最短の実行時間および最低の利用コストで実行したインスタンスをそれぞれ記録した．次にm4.largeにおける実行性能を用いて類似するインスタンスをそれぞれ選択した．結果を表2に示す．

表2 実行結果

application		best time	best cost	Similarity application	best time instance	best cost instance
Sysbench	prime number	c4.xlarge	c4.xlarge	word count	Same	Different
	memory read	c4.xlarge	c4.large	file sort	Same	Same
	memory write	c4.xlarge	c4.large	memory read	Same	Same
	random read/write	m4.xlarge	m4.xlarge	Dhystone	Different	Different
	sequential read/write	m4.xlarge	c4.large	file sort	Different	Same
hadoop	pi	c4.xlarge	c4.large	prime number	Same	Different
	word count	c4.xlarge	c4.large	memory write	Same	Same
	file sort	c4.xlarge	c4.large	memory read	Same	Same
UnixBench	Dhrystone	c4.xlarge	c4.xlarge	fsdisk	Different	Different
	Whetstone	c4.xlarge	c4.large	fsdisk	Different	Same
	fsdisk	c4.large	c4.large	Dhystone	Different	Different

Sysbenchの素数計算処理は最も類似するインスタンスはhadoopのワードカウントであり，最短の実行時間であるインスタンスは一致，最低コストであったインスタンスは不一致となった．類似するアプリケーションにおいて実行時間および利用コストに優れるインスタンスの一致率は，共に約55%となった．

実行時間が最短であったインスタンスではc4.xlargeが，利用コストが最低であったインスタンスではc4.largeが多いという結果となった．この結果は本実験で使用したアプリケーションの多くがCPUバウンドであることによると考えられ，一致率に大きな影響を与えている．本提案手法の有用性を示すためには，処理の大きく異なるアプリケーションの利用が必要である．

6. まとめ

本実験では類似度を用いた計算資源の選択手法の提案を目指し，複数のアプリケーションを用いた実験を行った．一致率は約55%となり，本提案手法の有用性を示すには不十分な結果となった．原因として利用したアプリケーションの多くがCPUバウンドであったことが考えられる．

参考文献

[1] Amazon Elastic Compute Cloud (EC2), http://www.amazon.com/ec2/.

[2] G.Juve. Scientific workflows and clouds. Crossroads,Vol.16,pp.14-18, 2010.

[3] G. Juve. Scientific workflow applications on Amazon EC2. 5th IEEE International Conference on e-Science Workshops, pp.59-66, 2009.

Copyright ©2017 Information Processing Society of Japan.
All Rights Reserved.