機械学習技術を応用したMapReduceタスク割当決定方法

藤森偉恭[†]　浅原理人[†]

1. 背　　景

MapReduce[1] や Dryad[2] を例とする，大量のデータを対象とした分散並列処理基盤が提案されている．こうした基盤は多数の安価な計算機で構成された計算機クラスタの上で動作し，利用者に対して計算機クラスタの共有環境を提供する．利用者はこの共有環境上で，計算機クラスタに格納された巨大なデータセットに対して様々なバッチ処理を実行することができる．共有環境を構築することで，大量データの複製や移動といった負担の大きい作業が不要となるため，様々なデータセットに対する処理を効率よく実行できるようになっている．

一方，分散並列処理基盤において，タスクの組み合わせによってタスクの実行時間が遅くなり，その結果ジョブの完了が遅れる現象が観測されている．一般に分散並列処理では，利用者が投入する処理の集合（ジョブ）は，集合を構成している処理の単位（タスク）に分割される．分散並列処理基盤は自身を構成する計算機に対してタスクの実行を指示し，タスクの出力結果を集約したものが最終的なジョブの出力として得られる．ここで，ジョブの完了とは全てのタスクが完了することである事実に注意する．つまり，タスクの組み合わせによって実行が遅れるタスクが存在すると，そのタスクによってジョブの完了が遅延することを意味している．

大規模な分散並列処理基盤においてはこのようなタスクによるジョブ完了時間の遅延の影響は大きい．例えば，Microsoft 社の Bing 検索サービス用計算機クラスタは 2009 年から 2010 年において，数万台の計算機で構成され，一日当たり数百のジョブが実行されそれらのジョブによって数ペタバイトのファイルが処理されていたが，この環境では実行が遅くなったタスクによってジョブ完了時間の中央値が 34%増加したと報告されている[3]．

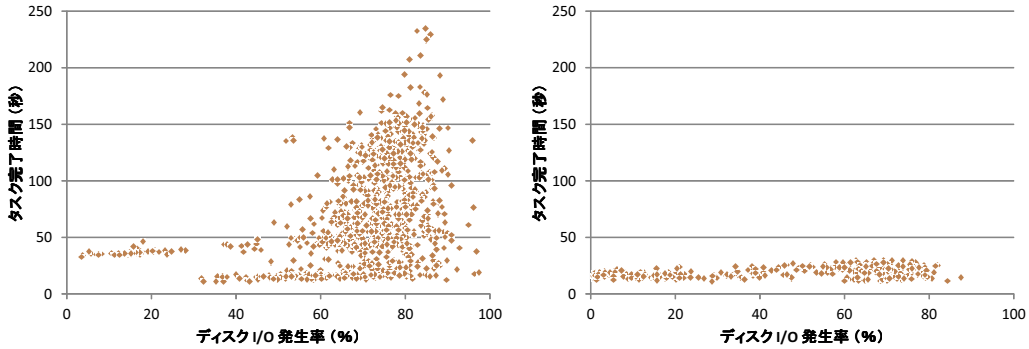
† NEC グリーンプラットフォーム研究所
Kawasaki, Kanagawa 211–8666, Japan

2. 従来手法と問題点

実行が遅くなったタスクによるジョブ完了時間の遅延を短縮する手法が研究されている．このようなタスクは，一般に利用計算機資源の競合やネットワークの輻輳といった、資源利用状況の変化やタスクの組み合わせによって突発的に発生する．このようなタスクを推定し，タスク配置を工夫することで，ジョブの処理時間の増加を抑えることが課題である．実行が遅くなったタスクによる処理時間の増加を抑制する技術のひとつとして Mantri[3] がある．Mantri では，タスクの実行中にその完了時間を推測し，他と比較して十分長いタスクに対して，再実行や他の計算機上での複数実行を行う．これにより異常タスクの影響が緩和され，処理時間の増加が抑えられるとされている．

ところが，従来のタスク完了時間推測方法は，計算機クラスタの資源利用効率を悪化させるおそれがあることが我々の観測によって明らかとなりつつある．従来手法ではタスクの完了時間を，タスクの進捗率やシステム負荷などから算出するモデル式を決定し，そのモデル式にしたがって一意の値を求めることで推測していた．この方法は，タスクの進捗率やシステム負荷に対して，タスクの実行時間の分散が小さい場合には有効である．一方で，性質の異なる複数のジョブが同時に実行された場合，ジョブによってはシステム負荷が同一であってもタスクの実行時間が大きく散らばることがある．このような場合，実際にはタスクの完了時間が短いにも関わらず，タスクの割り当てが行われないことで資源の利用効率が悪化する場合が生じる．

図 1 は，我々が実際に MapReduce クラスタで計測した，ディスク I/O 発生率とタスク完了時間の関係を示す散布図である．この結果は，30 台の計算機クラスタにおいて，MapReduce ベンチマークのひとつである GridMix2[4] に含まれる MonsterQuery ワークロードと JavaSorter ワークロードを実行し，それぞれのワークロードのタスク完了時間と，そのタスクを実行した処理サーバのディスク I/O 発生率を計測したものである．図 1 から，MonsterQuery ワークロー



(a) MonsterQuery ワークロード　　(b) JavaSorter ワークロード

図 1　ディスク I/O 発生率とタスク完了時間の関係.

ドは低負荷時には完了時間の散らばりが小さいが，高負荷時には大きくなっていることが分かる．一方で，JavaSorter ワークロードでは負荷に関わらず完了時間の散らばりが小さい．このような場合，従来手法では MonsterQuery のタスクは高負荷時に実行時間が常に長いと判定し，高負荷である処理サーバにタスクを割り当てないように動作する．しかし，高負荷時でも低負荷時と同様の完了時間であるタスクが一定数存在していることが，図 1 から読み取れる．したがって従来手法では，高負荷の場合は例え処理時間が短い可能性があったとしてもタスクを割り当てないことになる．このことは不必要に計算機クラスタの資源利用効率を悪化させる原因となりうる．

このことから，ジョブの種類とシステム負荷によって実行時間の散らばりが異なる様子を表現できるタスク完了時間の予測方法と，その予測方法を使用してタスクの割り当てを行う仕組みがあれば，資源利用効率を保ちながらジョブの完了時間を短縮することが可能になると考えられる．

3. 提　　案

本発表では，機械学習技術を応用してタスクの完了時間を予測し，さらにその予測結果からタスクの割り当てを判定する MapReduce タスク割当決定方法について述べる．本提案手法を用いると，タスクの完了時間の散らばりが大きい場合においても，その散らばりを反映した予測完了時間を推定することができる．また，その推定時間を用いて，計算機クラスタの資源利用効率を保ちながら，ジョブの実行時間を早めるようにタスク割り当てを行うことが可能になる．また，提案手法はジョブ間の資源割り当てポリシに大きな影響を与えることなく，タスク割り当てを行うことができる．

提案手法の詳細についてはポスターにて述べる予定である．

参　考　文　献

1) Dean, J. and Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters, *Proc. of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 137–149 (2004).
2) Isard, M., Budiu, M., Yu, Y., Birrell, A. and Fetterly, D.: Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks, *Proc. of ACM European Conf. on Computer Systems (EuroSys)*, pp. 59 – 72 (2007).
3) Ananthanarayanan, G., Kandula, S., Greenberg, A., Stoica, I., Lu, Y., Saha, B. and Harris, E.: Reining in the Outliers in Map-Reduce Clusters using Mantri, *Proc. of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 265–278 (2010).
4) Apache: GridMix, `http://hadoop.apache.org/docs/stable/gridmix.html`.