不完全な情報共有による分散型ジョブスケジューリングシステム

梅田典宏[†]　中田秀基[††,†]　松岡　聡[†,†††]

1. はじめに

近年、ネットワークによって結合された複数の計算機上で大規模計算を行うグリッドが普及しつつある。グリッド環境では、利用可能な計算資源の状態とユーザから投入されるジョブの情報を収集し、適切な割り当てを行うジョブスケジューリングシステムによって計算資源の効率的な利用を可能にしている。

Condor [3], XtremWeb [1] といったジョブスケジューリングシステムでは、計算資源の情報収集とジョブの割り当てを一台ないしは少数の固定された計算機によって行っており、それらに障害が発生すると、管理下におかれた資源全体の利用が不可能になってしまう。また、情報収集にかかる通信量や資源とジョブのマッチングを求めるコストが少数の計算機に集中することから、今後予想される計算資源および投入されるジョブの増大に対するスケーラビリティに難がある。

本研究では、コストの低い不完全な情報共有手法によって各計算資源の状態を複数のノードで共有し、スケジューリングを複数の計算機で分散して行うことにより、負荷集中と単一故障点の排除を目指すシステムを提案した。そして、本提案手法と既存のシステムが基盤としているモデルをシミュレーションで比較し、情報の不完全性と計算資源および投入ジョブ数が性能に与える影響を評価した。

2. 分散型ジョブスケジューリングシステムの設計

スケジューリングに必要な計算資源の情報を、スケーラブルな情報拡散手法を用いて複数のノードで共有し、複数ノードでジョブのマッチングを可能にする分散型ジョブスケジューリングシステムを提案する。

2.1 資源情報の共有

各計算資源は、定期的ないしは状態が大きく変化した際に CPU 使用率、空きメモリ、ディスク容量に代表される資源情報を他ノードに送信し、多数のノードで互いの情報を保持する。ジョブがユーザから投入された時には、投入されたノードが持つ各資源の情報とジョブ実行に必要な条件のマッチングを行い、より適した計算機にジョブの実行を指示する。

資源情報を共有することで複数のマシンによるスケジューリングが可能になることから、既存のスケジューリングシステムの課題であった計算資源と投入ジョブの増加によって生じるマッチングコストを分散でき、また耐故障性を向上させることができる。

また、適切な情報拡散手法を用いることで、大量の計算資源から情報を収集するために起こる通信の集中を軽減することが可能になる。

3. 不完全な情報共有手法

本提案で用いる情報共有手法は、情報を生成する計算資源の数の増加に対し、通信量および伝達速度の面でスケーラブルである必要がある。また、ノード数が増加すると通信路およびノードの故障数も増加するため、それらによって全体の通信が阻害されないことも重要である。つまり、大規模な環境で完全な情報を収集するのはコストが高くまた困難が多いため、妥当な精度でより負荷の軽い手法を取る必要がある。本研究では、次に述べる Gossip Protocol を採用した。

3.1 Gossip Protocol

Gossip Protocol [2] は、多数のノードにおける情報流通・共有を行うための通信手法である。次に示す過程を反復して実行され、最終的にすべてのノードで停止する収束状態をもって伝達が終了する。

(1) メッセージの発信元は接続可能なノードをランダムに選択し、メッセージを転送する。
(2) メッセージを受信したノードは、メッセージが既知か否かで次の動作に分かれる:
  (a) 既知であったときは、それを送信元に伝える。
  (b) 新しいメッセージを受信したときには、自身もそのメッセージの発信元となり、過程 (1) を開始する。
(3) 送信元は、一定回数以上既知の応答を受けるまで繰り返す。回数を超えたら停止する。

毎回通信経路が動的に選択されるため、あるノード

† 東京工業大学
Tokyo Institute of Technology
†† 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
††† 国立情報学研究所
National Institute of Informatics

間の通信路ないしはノード自身に障害が発生した場合でもそれを迂回する耐故障性を備えている。

Gossip Protocol では、全ノードへメッセージを伝達させる保証はできないが、高い伝達率を実現している。また、周期ごとに伝達ノードが指数的に増加するため、全体のノード数増加に対する収束回数の増加は小さい。一周期に選択する伝達ノード数を 1、既知応答を受ける回数を 1 としたときの収束時における全ノードに対するメッセージの伝達率を表 1 に示す。

ノード数	100	1000	5000	10000
平均メッセージ伝達率	0.978	0.953	0.996	0.992
平均収束回数	11.6	18.8	18.9	20.6

表 1 メッセージ伝達率

4. 性 能 評 価

提案した分散型ジョブスケジューリングシステムと、Condor をモデルとした既存の集中管理型システムを、一定の計算量を持つジョブを任意の間隔でシステムに投入するシミュレーションを行うことで評価した。なお、資源情報収集を除く資源とジョブのマッチングなど他の要素については両者とも同様の機構とした。

4.1 複数マシンによるスケジューリングの性能

本提案手法の上で計算資源とジョブのマッチングを行うノードの台数による性能の変化を測定した。計算資源 1000 台のシステムに対し、処理能力の総和に対する計算量が 100 %となるような投入間隔に従うジョブ列を投入し、それぞれ 1 台、2 台、4 台、8 台でマッチングした際のシステム全体の稼働率を表 2 に示す。台数が増えることでより効率的な割り当てが行われている結果が得られた。

マッチングノード台数	1	2	4	8
システム稼働率	0.939	0.983	0.986	0.988

表 2 マッチングを行う計算機の数によるシステム全体の稼働率

4.2 投入ジョブの計算量による性能の変化

計算資源の処理能力の総和に対する計算量が 10 %,50 %,100 %,200 %となるような投入間隔にしたがうジョブ列を、一台のマッチング計算機からそれぞれ 100 台,1000 台,5000 台のマシンに投入するシミュレーションを行った。集中管理型を 1 とした際の本提案手法の性能比を表 3 に示す。資源が少ないときは、計算量が大きくなると相対性能が悪くなるが、資源数が増えるにつれ相対性能は逆に向上している。

	資源全体の能力に対するジョブの計算量			
	0.1	0.5	1.0	2.0
100 台	0.998	0.995	0.853	0.823
1000 台	1.000	1.002	1.003	1.012
5000 台	1.007	1.016	1.017	1.060

表 3 計算資源数と稼働率の性能比

5. 考　　察

4.1 の結果では、マッチングを行うノードが多いほど稼働率が向上し、効率の良いスケジューリングが行われている。これは、マッチングの負荷が分散されたためと考えられる。

4.2 の結果では、資源の数によって投入ジョブ列の多少による性能変化の形が大きく異なる。これは、Gossip の伝播速度に原因があると考えられる。システムの稼働率が高く、実行待ちジョブが多量にある状態では、計算資源情報の伝達速度がスケジューリングの効率に大きい影響を与える。表 1 によると、ノードが少ないときの収束回数は多いときのそれと比べ相対的に大きい。つまり、計算ノードおよび投入ジョブの絶対数が少ないときはより単純な収集機構である集中管理型のほうが高速となる。

しかし、集中管理型では情報収集のコストが計算資源数に対し線形に増加するため、資源数が増えると Gossip の伝達速度より通信の集中の影響が相対的に大きくなるため、本稿で提案した手法のほうがより効率的なスケジューリングが可能になると考えられる。

6. まとめと今後の課題

不完全な情報共有手法で資源情報の収集を行う分散型ジョブスケジューリングシステムを提案し、シミュレータ上で既存の集中型との性能比較を行った。今後の課題としては、Gossip 以外の情報共有手法の採用、遅延や通信方向の非対称性が存在するネットワークの考慮、そして各資源から発せられる情報の検証機構の検討などが挙げられる。

参 考 文 献

1) Vincent Ne'ri Gilles Fedak, Ce'cile Germain and Franck Cappello. XtremWeb : A Generic Global Computing System. In *CCGRID2001, workshop on Global Computing on Personal Devices*, 2001.

2) K. Jenkins, K. Hopkinson, and K. Birman. A Gossip Protocol for Subgroup Multicast. In *International Workshop on Applied Reliable Group Communication (WARGC 2001)*, April 2001.

3) M. J. Litzkow, M. Livny, and M. W. Mutka. Condor - a hunter of idle workstations. In *Proceedings of the 8th International Conference on Distributed Computing Systems (ICDCS)*, pp. 104–111, Washington, DC, 1988. IEEE Computer Society.