B-012

Web ベース計算リソース管理ミドルウェア：ShareTask
Web-based Computing Resource Management Middleware : ShareTask

齊藤 隆之†, 善甫 康成‡
Takayuki Saito, Yasunari Zempo

1. はじめに

計算機の発達に伴い，構造・流体・材料解析などの解析計算ニーズは非常に高まっている．一方、多くの企業イントラネットでは，セキュリティ向上のために多くの規制が導入され，部門ネットワーク間の通信にも制約が多い．このような環境において，計算機クラスタ，クラウドサービスなどの HPC 計算資源を統合的に運用し，計算機シミュレーション業務を効率化が求められている．このような観点から, 計算サーバのコマンドとファイル操作を抽象化して解析アプリケーションの実行が行なえる GUI が求められている．セキュリティの観点から PC への個別アプリケーションのインストールが制限されるようになってきているため，GUI を Web ブラウザ上に統一して実現することは非常に意味があることと言える．

2. HPC 環境の課題

ハードウェアの高性能化と低廉化に伴い計算機の増設とリプレースが積極的に行える状況にある．しかし，計算機クラスター導入単位毎に管理ツール(ジョブスケジューラ等)が異なることも多く，ユーザは異なるツールの使い方を習得しなければならないことも多い．

計算ノード**(以下，ノード)**を増やすことは比較的容易になってきている一方で，それらを管理するための工数は増える傾向にある．解析用計算機について専任の管理者がいることは希で，解析業務の傍ら運用管理を行っている場合が多い．担当者の負担を軽減するために，ハードウェア障害等で機能不全に陥ったノードをシステムから切り離し，計算サービスを継続させるなどの障害対応操作を自動化することが求められる．

節電要求の高まりから，計算需要(ジョブ数)を超えた余剰のノードを一時的に休眠させるなど，積極的な電力制御が必要になってきている．システム全体の停止・起動の操作も自動化されることが望ましい．

計算環境の投資計画を立案するとき，既存設備の処理能力が有効に利用されているかを定量的に分析できることが重要である．設備だけでなく，アプリケーションのライセンスは高価なものが多いため，その使用量の分析も重要である．計算パワー調達の観点からは，クラウドサービスを活用し，ローカルの計算環境とクラウド上の仮想マシン環境を統合して，経済的効率性を追求することも求められる．

著者らは，以上の課題と要求を包括的に解決する計算システム運用管理ミドルウェア ShareTask を開発した．

3. ShareTask のアプローチ

機能は，以下の３種類を統合したものである．

1. 計算資源の利用分析と可視化(resource analysis)
2. 計算資源のコントロール(resource control)
3. ジョブスケジューリング(job scheduling)

多数のノード(仮想マシンも含む)からなる分散システムにおいて，ハードウェア障害と通信障害も考慮しながら，各ノードの状態を管理するとともに，システム全体の状況も管理しなければならないことから，自律分散制御の考え方を採用した．

構成は，以下の２つの要素からなる．

(1) レポジトリサーバ**（以下，RS）**

HTTP サーバ(Apache)，Perl で記述された CGI 群，ならびにデータベース(PostgreSQL)とから構成された Web アプリケーションである．ノードの管理情報，ユーザ認証，各種ログ，ジョブの待ち行列を保持し，Web ブラウザとエージェント（次項）からの CGI 呼び出しに応答する．Web アプリケーションであるため，ジョブ制御の GUI(Web 画面フォーム)の開発基盤にもなっている．エージェントからの報告によって収集された資源の状態は，データベースに蓄積され，分析され，Web 画面で可視化される．

(2) エージェント**（以下，AG）**

ノードに常駐して，CPU，メモリ，ファイルシステムの状態を監視・制御する Java プログラムであり，RS に対する HTTP クライアントである．AG は，ノードのさまざまな OS で稼働しないといけないために，マルチプラットフォーム性が高い Java で開発されている．

AG は，自律的かつ能動的に機能してノード内資源(CPU, メモリ，HDD，プロセス)を管理する．RS の CGI を呼び出すことにより，資源状態を報告するとともに，命令を受け取りそれを実行する．具体的には，

- CPU，メモリ，HDD，プロセスの使用状況報告
- ジョブの割当，実行開始，一時停止，強制終了
- シャットダウン等 OS レベルの制御

などがあげられる．

AG は，ノードの状況に応じて以下のような制御を行う．

- ジョブを実行しない遊休時間が続くとノードをシャットダウンあるいはサスペンドして消費電力を抑える．（一方，ジョブの待ち行列が長くなると，RS 側の判断によって休眠ノードに対して Wake-on-LAN あるいは IPMI 通信により起動命令が送信される）

†㈱アンクル，ANCL, Inc.
‡法政大学 情報科学部

- CPU，メモリなどの資源枯渇，あるいは障害を検出するとジョブを取り込まない．AG 自体が機能不全に陥る障害がノードに発生した場合，あるいは AG・RS 間の通信経路に障害が発生した場合には，自然にノードが切り離され，障害ノードが計算システムに影響を与えることが回避される．

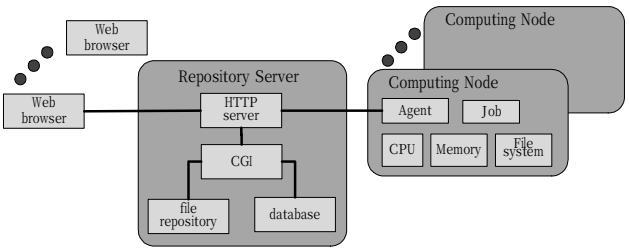

図 1 ShareTask の構成

RS は，ShareTask の制御の中心に位置しているが，受動的な存在であり，AG からの CGI 呼び出しによって，AG に情報を与え，AG 間の相互作用を媒介する．AG・RS 間の通信が，AG からの RS への CGI 呼び出しのみによって実現されることにより，CGI 呼び出し頻度によって制御が遅延してしまう欠点，ならびに AG 数の増大に伴い RS の CGI 呼び出し頻度が高くなる欠点を持つが，CGI 呼び出し頻度を動的に制御することによりこれらの欠点を緩和するしくみを開発中である．[5]

計算ジョブを蓄積しているサーバから，HTTP クライアントであるノードがジョブをダウンロードし，実行し結果をアップロードするというしくみは，ボランティアコンピューティング[3][4]でも活用されているが，ShareTask は，この方式を汎用のジョブスケジューリングと計算システム運用管理へと発展させたものである．

RS の核は，データベースである．ユーザ(Web ブラウザ)と AG がデータベースを更新することによって相互作用し，その結果として全体が機能する．複数ノードにまたがって実行する MPI ジョブの場合，ノード内資源に空きがあるノード群が RS を介して待ち合わせ（ランデブー）を行い，必要なノード(CPU 数)が揃ったところでジョブ実行を開始するしくみになっている．ノード群のレイアウト（ネットワーク的近接ノード，同一性能のノードを使用したいなどの制約条件）の制御が必要になるが，ノード間のランデブーにおいて制約条件をかけることによって可能である．すでに実地試験を行っており，良い結果を得ている．[6]

AG は，ノード毎に配備される他，クラスターのログインノードのみに配備され，既存のバッチシステムを操作するように構成することが可能である．つまり，ShareTask の配下に複数の既存バッチシステムを連携させることにより，複数のクラスターを仮想的にひとつの計算機プールとしてみせることができる．この手法を，メタスケジューリングと呼ぶ，メタスケジューリングと，AG・RS 間通信が HTTP のみで実現されていることから，イントラネットだけでなく遠隔地にある計算センターにある計算機クラスターあるいはクラウドサービスの仮想マシン群をファイアーウォールを越えて統合してひとつの計算資源として利用することが可能であり，実証している．（図 2）[1][2]

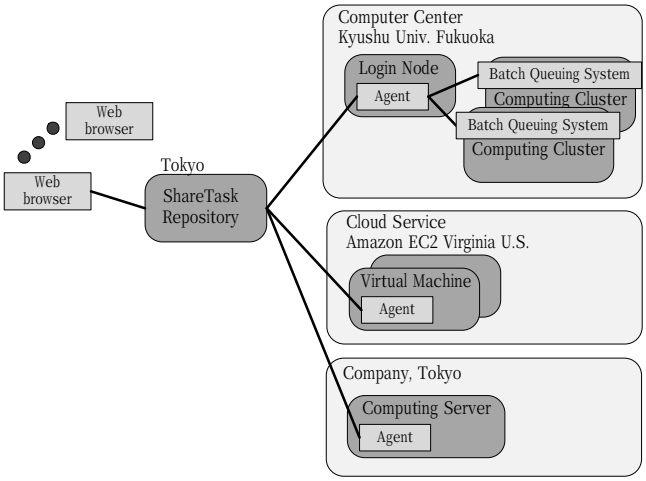

図 2 メタスケジューリングの実証実験環境

4. まとめ

HPC 計算環境が抱える課題を包括的に解決することを目指して開発した計算資源管理ミドルウェア ShareTask について述べた．そこでは，自律分散制御によって多数のノードが制御され，協調して機能することにより，ジョブの割当実行制御から障害ノードの切り離し，またノードの起動停止までを自動化できる．今後は，より多数のノードを制御するためのスケーラビリティの向上につとめるほか，管理対象を計算ノードだけではなく，共有ファイルシステム(ストレージ)などへ広げる計画である．

謝辞

メタスケジューリングの実証では，クロスアビリティ古賀良太氏，テンキューブ研究所千田範夫氏の協力に，また MPI ジョブ制御などユーザ実環境での実証では，キヤノン IT ソリューションズ岡戸晴彦氏，金井信之介氏，住友化学石田雅也氏，アルゴグラフィックス後藤成志氏の協力に感謝いたします．

参考文献

[1]善甫康成, 齊藤隆之, 岡戸晴彦, 近野利信, 千田範夫, “自律プル型制御方式によるインターネットワイドなジョブスケジューリングの性能試験”, 九州大学情報基盤研究センター 先端的計算科学研究プロジェクト成果報告会 (2010).
[2]善甫康成, 齊藤隆之, 岡戸晴彦, 近野利信, 古賀良太, 千田範夫, “メタスケジューリングを指向した計算環境”, 日本コンピュータ化学会 2010 年春季年会研究展示 RX02 (2010).
[3]D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. ”SETI@home: An experiment in publicresource computing”. Communications of the ACM, Nov. 2002, Vol. 45 No. 11, pp. 56-61.
[4]David P. Anderson. BOINC: A System for Public-Resource Computing and Storage. 5th IEEE/ACM International Workshop on Grid Computing, November 8, 2004, Pittsburgh, USA, pp 1-7.
[5]投稿準備中
[6]投稿準備中