B-015

HPC 計算資源管理におけるソフトウェア動的配備技法について
Software Dynamic Deployment Methods for HPC Resource Management

齊藤隆之 † 平松和剛 † 善甫康成 ‡
Takayuki Saito Kazutake Hiramatsu Yasunari Zempo

1. はじめに

筆者らは，HPC 計算機クラスターの計算資源管理ミドルウェア ShareTask を開発してきた．[1, 2, 3, 4]

このソフトウェアでは、計算ノードに常駐するエージェントが主導権をもち、ジョブ、入力ファイル、制御命令を管理サーバーからダウンロードし、計算ノード側で発生したイベントと出力ファイルを管理サーバーにアップロードすることにより、ジョブ制御から計算資源監視までをプル型で実現している。(図 1) 一般的な計算機クラスターから，物理的に分散した計算機群、あるいはクラウド環境までを統一的に扱うために，通信プロトコルとしては HTTP(HTTPS) のみを使用している。計算ノード数の動的な増減への対応が容易なため、クラウド環境において多数の仮想マシンを動的に生成・消滅して計算資源を調達する際に、この特徴が有利であることが実証されている。

一方、プル型の問題点であるノード数の増加に伴う通信遅延と管理サーバーの負荷 (C10K 問題) については、WebSocket の導入により、管理サーバー側のイベント発生を遅延なくノードへ伝達することによって解決した。[1]

しかし、動的に起動した計算ノードのソフトウェア構成 (OS 設定、ライブラリ、アプリケーションプログラムなど) を管理することは煩雑な作業を必要とする。物理マシンで構成された計算機クラスターにおいても、ノードのソフトウェア環境の構成管理は、ノード数が大きくなるにつれて管理者にとって大きな負担となる。

本稿では、ShareTask の計算資源管理の一機能としてソフトウェアの配備・構成管理の機能を統合した方法について報告する。

2. 従来手法とその課題

計算ノードへのソフトウェア環境の配備 (deployment) および構成管理 (configuration) には、以下の操作が必要である。

1. OS のインストール
2. ネットワーク設定
3. ランタイム (ライブラリ) のインストール
4. アプリケーションプログラムのインストール
5. インストールと設定状態の検証

これらについて、従来、大別して 2 つの手法が用いられている。

OS イメージ配信　OS イメージのマスターとするノードにおいて、ソフトウェア環境構成を手動で行い、そのハードディスクの状態を凍結した OS イメージファイルを作成する。この OS イメージファイルを各計算ノードに一括配信してハードディスクに展開する。展開後に、各ノード固有の情報 (ネットワーク設定等) について設定を行う。

この手法では、追加的な構成変更が必要となった場合にも、OS イメージファイルの作成に立ち戻る必要があり、イメージの再配信にも時間を要するため、計算ノード数が増えるに従いその効率が課題となる。

構成手順配信　ソフトウェア環境の構成手順あるいは状態記述 (以下、スクリプトという) を各計算ノードで実行する手法がある。前提として基本的な OS のインストールとネットワーク設定がなされている必要があるが、スクリプトの中で、ランタイムとアプリケーションのパッケージのダウンロードと、さまざまな設定操作を実行することにより、追加的・差分的な構成変更を効率よく迅速に行うことができる。[5] 多数のノードを構成対象とする場合には、中央の配信サーバーから SSH でプッシュする方法をとるが、対象ノードが NAT 内部に存在する場合には対応が困難である。

3.ShareTask の配備手法

ソフトウェア配備・構成の機能を統合するにあたり、プル型制御の観点から上述の課題を解決するために、以下のようなアプローチを採用した。

物理マシンの場合　OS インストールとネットワーク設定については、Kickstart を用いる。ネットワークブートによって配布される OS イメージにエージェントを含めておくことにより、OS 起動後はエージェントによって構成管理が行われる。エージェントは、管理サーバーから構成手順をダウンロードし実行する。

仮想マシンの場合　クラウド環境などの仮想マシンにおいても、エージェントが含まれた状態で起動することにより、エージェントが管理サーバーから構成手順をダウンロードして実行する。

物理・仮想いずれの場合も、エージェントによって以下の手順で、ソフトウェアの配備と構成変更を行うことができる。

1. ノードの OS が起動した時点でエージェントが起動する
2. エージェントが管理サーバーから構成手順をダウンロードして実行する
3. エージェントが構成の検査手順を実行して管理サーバーに報告する

†(株) アンクル, ANCL, Inc.
‡法政大学 情報科学部

4. 検査により不備が発見されれば、再度構成手順を実行する

コンテナー仮想化の場合　ランタイムとアプリケーションの複数の組み合わせの構成を、すべての計算ノードにおいて事前に行っておくことが、従来の構成管理方法である。これに対して、ランタイムとアプリケーションの構成を、ジョブ実行時にその実行ノードにおいて動的に行う方法が考えられる。

Docker[6] に代表されるコンテナー仮想化技術を用いることにより、異なるランタイムとアプリケーションの組み合わせを容易に配備できる。エージェントがジョブを実行する際に、アプリケーションプログラムを直接起動するのではなく、コンテナーに包んだ形態で起動する。

さらに、同じ計算ノード上に複数のコンテナーを配備することも可能であるため、ランタイムとアプリケーションの異なる構成のジョブを同じ計算ノード上で同時に干渉することなく実行することができる。CPU の many core 化の流れにおいて、ひとつの計算ノードの CPU コアを複数のジョブで分割利用するための手法としても、コンテナー仮想化は有用であると考えられる。

4.ShareTask の機能構成

我々が開発した計算資源管理機能に、ソフトウェア配備・構成機能を統合した手法について述べる。

基本となるシステムは、計算ノードに常駐するエージェントと、それらから HTTP 要求を受け付けてエージェントとの命令配信・情報収集を行う管理サーバーとから構成され、つぎの 3 つの機能が実現されている。

- 計算資源の監視・分析・可視化 (resource analysis)
- 計算資源の制御 (resource control)
- ジョブスケジューリング (job scheduling)

ソフトウェア配備・構成の機能　ソフトウェア配備・構成の機能は、計算資源の制御とジョブスケジューリングを組み合わせて実現した。

計算資源制御機能は、各エージェントに対する構成命令をジョブとして生成し、各エージェント毎に定義された専用のジョブキューに登録する。

エージェントは、当該構成ジョブを取り込み、そのソフトウェア構成手順を実行する。(図 2)

コンテナー仮想化型ジョブの実行機能　コンテナー仮想化型ジョブでは、そのソフトウェア環境を定義したコンテナーイメージが指定されているので、エージェントは、当該コンテナーイメージからコンテナープロセスを起動する。

5. まとめ

HPC において、多数の物理・仮想の計算ノードのソフトウェア構成を管理することが大きな負担であり、その自動化・効率化が計算資源の運用管理において重要である。

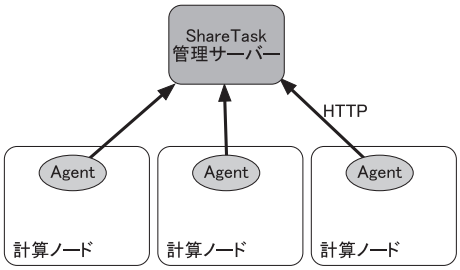

図 1: ShareTask の構成

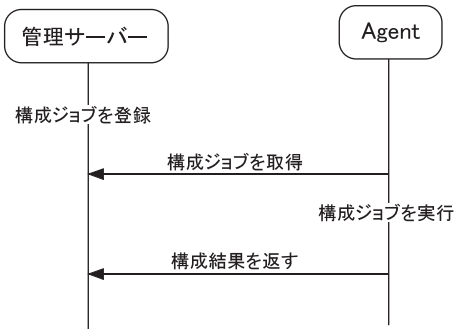

図 2: 構成ジョブの実行

ShareTask がもつプル型の特徴を活かして、エージェントが構成命令をジョブとして管理サーバーから受け取り実行する方式を採用した。また、ジョブの実行形態としてコンテナーによるソフトウェア構成と計算ジョブを一体化した取り扱いも試みている。

これらの工夫によって、クラスターからクラウドに至るさまざまな環境において、各計算ノードのソフトウェア構成を固定的に維持管理するだけでなく、計算ジョブの要求に応じて動的に変更する運用も可能になる。現在、その有効性について実証実験を進めており、その結果を報告する。

参考文献

[1] 齊藤隆之, 善甫康成, ”プル型ジョブスケジューラにおける動的通信量制御方式”, FIT2013 B-016.
[2] 齊藤隆之, 善甫康成, ”Web ベース計算リソース管理ミドルウェア：ShareTask”, FIT2012 B-012.
[3] 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).
[4] 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).
[5] http://www.getchef.com/chef/
[6] http://www.docker.com/