

ストレージイベント駆動型ジョブ制御システム：TaNAS Storage Event-driven Job Control System : TaNAS

平松和剛[†]

Kazutake Hiramatsu

齊藤隆之[†]

Takayuki Saito

1. はじめに

パラメータスタディなど膨大な数のジョブをハンドリングする状況では、ジョブ管理システムとストレージとの連携が重要である。一方、クラウドストレージを利用した計算業務が増えている。我々は、ストレージのイベント通知機能によってジョブ管理とストレージの連携を実現する TaNAS(タナス) と呼ぶシステムを開発した。

2. 計算ジョブ操作における課題

HPC においてジョブ操作を効率良く行うことは非常に重要である。パラメータスタディでは、ジョブごとに使用する入力ファイルをストレージに配置してから、ジョブを投入する。この作業の問題点は、本来の解析業務に集中することが難しくなることである。入力ファイルと実行ディレクトリを指定して手動でジョブ投入をしなければならないため、パラメータスタディを実行するまでの作業の負担が大きい。

一方、昨今のクラウド環境の発達からストレージとして Amazon S3 などに代表されるクラウドストレージを利用する場合も多い。クラウドストレージは、クラウド環境の設定や配備を行う必要があるため、さらに効率が悪くなってしまふ。

3. TaNAS のアプローチ

このような計算ジョブ操作に関する課題を解決するために TaNAS では以下のアプローチをとった。

ジョブ投入の自動化 ジョブ操作において、これまでに入力ファイルを変更した後、ユーザー自身が手動でジョブ投入を行っていた。入力ファイルが大量にある場合、非常に多くのジョブを投入する必要がある。ジョブ投入を自動化すれば、ユーザーがジョブ投入する負担をなくすることができる。

UI の簡略化 一般的にジョブ操作には、ジョブ管理システムごとに専用の UI が必要である。ジョブが多くなればなるほど、条件にマッチしたジョブを検索したり、結果を確認したりすることに時間がかかる。UI を簡略化できれば、ジョブの操作にかかる時間を短縮することができる。

計算結果の整理・蓄積 ジョブ管理システムを使っている場合、ジョブ実行のディレクトリが分散されていることが多く、計算結果を整理・蓄積することが難しい。特定のディレクトリツリーに計算結果を集約できれば、計算結果の整理・蓄積がしやすくなる。

4. TaNAS の実現手法

ジョブ投入の自動化を実現するためには、入力ファイルの変更を検知する必要がある。入力ファイルの変更がイベントとして通知されれば、ジョブ投入の判断を素早く行うことができる。

またジョブを実行するためのルールを記述するファイルが必要になる。このファイルをルールファイルと呼ぶ。入力ファイルが変更された時には、ルールファイルからファイルのタイムスタンプの前後関係を判断し、必要があればジョブ投入コマンドを実行する。

このようにルールファイルと入力ファイルの変更検知を組み合わせることで、自動的にジョブを投入する機能を実現することができる。

また UI を簡素化するために、ディレクトリ自体にジョブ状態を表すファイルを作成することにした。ディレクトリを見ればジョブの状態が把握できるので、専用の UI を使う必要がなくなり、ジョブ状態の確認が容易になる。

5. TaNAS の仕組み

TaNAS を使って ShareTask と連携する場合の仕組みを説明する (図 1)。

5.1. 基本機能

ルールファイル 最初にユーザーは解析対象のディレクトリに .tanas と呼ばれるルールファイルを配置する (図 1 の 1)。ルールファイルは、UNIX における Makefile と似た構文を持っており、ジョブに使用する入力ファイルと出力ファイル、ジョブ投入のコマンドを記述する。例えば .tanas ファイルは以下になる。

```
%.out: %.inp
    stsubmit -p MYPROG -np 8 -a $<
```

```
foo.inp: bar.txt
    cp $0 $<
```

.tanas は、下位ディレクトリに対しても有効となる。よって .tanas は個々のディレクトリに配置する必要はなく、ルートディレクトリに置けば、その配下の全てのディレクトリに対して有効となる。

次にジョブ操作に使う入力ファイルを配置する (図 1 の 2)。ファイルの変更を検知するために、最近の OS で実装されているファイルシステムのイベント通知機能を使う。入力ファイルを配置するとイベント通知機能が働きファイルの変更が TaNAS へ通知される (図 1 の 3)。通知を受け取った TaNAS は、対象のディレクトリに適用される .tanas を検索して、その内容を読み込み (図 1 の 4)、ジョブ投入コマンドを実行する (図 1 の 5)。

[†](株) アンクル, ANCL, Inc.

制御ファイル ジョブ投入が正常に行われると `{jobid}_submitted` という制御ファイルを作成する (図 1 の 6). TaNAS はジョブの状態に従って、(ドット) から始まる制御ファイルを作成する. `{jobid}_executing` は、ジョブ実行中に、`{jobid}_completed` はジョブ実行完了時に作成される. `{jobid}` には実行中のジョブ ID が入る.

TaNAS では、このように制御ファイルだけを見ればジョブの状態を確認できるようになっているため、特別な UI を必要としないことが大きな特徴である.

ジョブ実行 ジョブ投入後、しばらくすると Agent が ShareTask 管理サーバーからジョブを取り込み、ジョブを実行して結果ファイルを出力する (図 1 の 8). この結果ファイルは、Agent から ShareTask 管理サーバーへ転送されて TaNAS の管理するディレクトリへ出力される.

5.2.AWS への適用

Amazon S3 をストレージとして使う場合は、Amazon でのイベント通知機能である SNS (Amazon Simple Notification Service) を使ってイベント通知を実現する. SNS は Amazon S3 でファイルが追加された場合に、通知先として HTTP の URL を指定できるため、ShareTask 管理サーバーを通知先として設定しておけばファイルの追加/変更を検知することができる.

6.TaNAS の効果

計算ジョブ操作において、多数の入力ファイル管理してジョブ投入を実行することは大きな負担であり、その自動化・効率化が重要である.

TaNAS を利用するとジョブ操作の時間を短縮することができる. その結果、本来の解析業務に集中することができる. 入力ファイルを追加/変更していくだけで、ジョブが自動的に投入されていくので、解析業務を大きく効率化することができる. 計算結果がディレクトリツリーに集約されるため、知見が共有しやすい環境を作ることができる. TaNAS と ShareTask を連携すれば、従来通り WebUI からジョブの状態を監視したりジョブ制御を行うこともできる. クラウド環境にも対応しているため、オンプレミスからクラウドへ環境が変わった場合でも対応が可能である.

7. 今後の展開

ファイルシステムのイベント通知機能に対応していない環境では、ファイルシステムを再帰的に検索することで実現可能であるが、入力ファイルが大量にある場合、効率が悪く負荷も上がってしまう. 今後の展開として、再帰検索でも高速にファイルの変更を検知できる機能を開発する予定である.

参考文献

- [1] 齊藤隆之, 平松和剛, 善甫康成, "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" FIT2014 B-015

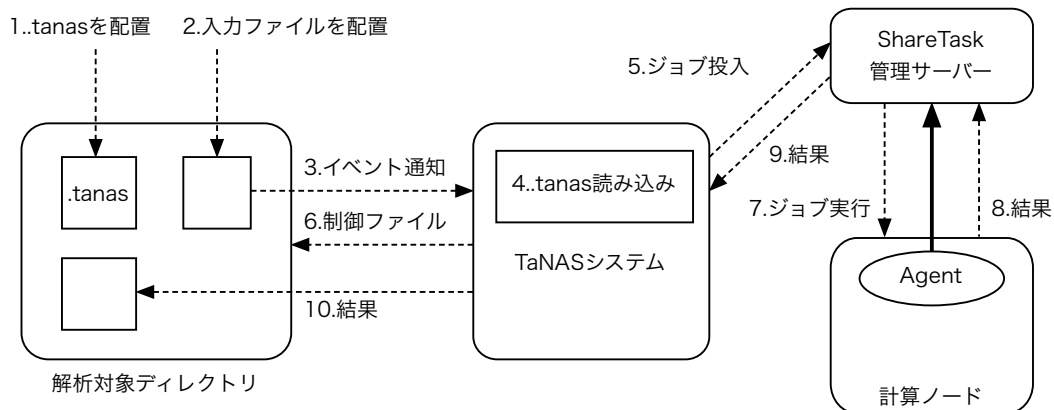


図 1: TaNAS の仕組み