ShareTask 5

リリースノート (ShareTask 4 からの変更点)

アンクル
2014 年

目次

1 はじめに

本文書は、ShareTask 4 から 5 への変更点について記述しています。
2014 年 11 月現在、最新版は 5.5.5 です。

2 変更・改善点

ShareTask 4 から 5（最新版 5.5.5）への変更点／改善点は以下の通りです.

2.1 ライセンス関連

ライセンスファイルの登録のしくみが大きく変更されました．従来は,

```
/home/sharetask/sharetask_server/conf/license
```

の下にライセンスファイルを展開していましたが，ライセンスファイルをブラウザからアップロードすることでライセンスを登録／更新する方式へ変更しました．ファイル転送、ログインしての操作が不要になったため、特に Hyper-V の場合に利便性が向上しました。

2.2 グラフ関連

2.2.1 グラフ更新停止の警告

ホストグラフ，ライセンスグラフについて，グラフの更新が止まっている (データ更新が止まっている) 場合は、背景を赤で表示し、更新が停止していることに気がつきやすくしました.

2.2.2 グラフの種類が増えました

下記のグラフが追加されました。（メインメニュー＞ ShareTask グラフ）

- ジョブライセンスのグラフ
- ジョブを実行している計算ノード数のグラフ (全体とキュー毎)
- 実行中・待機中のジョブ数のグラフ (全体とキュー毎)

2.3 クラウド関連

2.3.1 EC2 のコントロール

ホストコントロール画面から EC2 の VM（マシンインスタンス）を制御できるようになりました。

2.3.2 電源制御

計算ノードの電源オン（ブート）／電源オフ（シャットダウン）をウェブ画面から集中制御できます（ホストコントロール画面）
電源オンには, IPMI プロトコルあるいは Wake-up on Lan(WoL) を計算ノードごとに使い分けることができます.
電源オフは, エージェントが shutdown コマンドを発行することによって行います.

2.3.3 オートスケール制御

ジョブの待ち行列の長さに応じて、計算ノードを自動的に停止・起動できます。待機中のジョブが少なくなると、遊休状態の計算ノードが自動的に停止します。待機中のジョブが増えてくると、停止した計算ノードを自動的に起動します。
計算ノードは、ローカルの物理マシンでも、AWS EC2 上の仮想マシンでも、制御可能です。

2.4 ジョブ実行制御

2.4.1 ジョブ間依存関係の指定

ジョブの実行開始条件に、他のジョブの状態遷移を指定できるようになりました。(5.5.4 から)
複数のジョブの状態遷移も条件（論理式）として指定できます。
指定例：

- ジョブ A が正常に終了したら、ジョブ B を実行開始してよい
- ジョブ A が異常終了したときに限り、ジョブ B を実行開始してよい
- ジョブ A が異常終了したら、ジョブ B を削除する
- ジョブ A とジョブ B の両方が正常終了したら、ジョブ C を開始してよい

2.4.2 ジョブ実行順変更の操作性向上

従来バージョンでもジョブ実行順を変更することができましたが、その操作は直観的ではなく使い勝手が悪いものでした。新バージョンでは、ジョブリストからドラッグ・アンド・ドロップ（D&D) 操作によって、直観的な実行順変更ができるようになりました。

2.4.3 待機中ジョブのキュー変更

待ち行列上で待機中状態のジョブについて、そのアプリケーション、キュー、並列度の値を変更することができます。ジョブリスト画面から開いたジョブ詳細画面に表示された「アプリ・キュー・CPU 変更」ボタンをクリックするとダイアログが表示され、その中で設定値を変更できます。

2.5 処理性能の向上

2.5.1 ユーザーグループ情報更新処理の効率化

ユーザーグループの変更に追従するために、従来はログインするたびにユーザーグループ情報を読み直して、データベースを更新していました。このため、ユーザーグループの行数が膨大にあると、ログイン処理に時間がかかるという問題がありました。特に、コマンドライン（stsubmit 等）は、起動のたびにログイン処理を伴いますので、これは大きな問題でした。そこで、ユーザーグループ情報の更新処理を、ログイン時に行わないモードを追加しました。ユーザーグループに変更を加えたときには、管理者が手動で ShareTask のデータベースを更新する操作を行うためのボタンを用意しました。

2.5.2 mod_perl の導入

`mod_perl` 対応を行い，サーバーの応答を高速化するモードを導入しました．ただし、実績が少ないため実験的な位置づけです。(experimental)

2.6 ディレクトリ・ファイル閲覧機能の強化

実行ディレクトリの選択、ファイルの閲覧のための画面（ダイアログ）が、大きく機能強化されました。（ディレクトリエクスプローラーと呼びます）

2.6.1 head, tail 機能

head, tail は、サイズが大きいファイルを効率よく閲覧するために導入されました。特に、tail では、更新ボタンを押すことによって、ファイル出力の最新状態を効率よく閲覧することができます。

2.6.2 edit 機能

edit により、テキストファイルの編集が可能になりました。入力ファイルのパラメータ変更などが、ブラウザから可能です。

2.6.3 sudo への移行

従来は、apache に対して許可されたディレクトリ、ファイルに対してのみアクセスが可能でした。このために、ユーザーグループ毎にメンバーに apache を追加するという煩雑な管理が必要でした。このために、本来はアクセスを制限したいディレクトリに対して ShareTask の画面からアクセスできてしまうというセキュリティ上の問題がありました。
5.5 からは、sudo のしくみを使用することによって、apache 権限ではなく、ログインユーザーの権限でファイルアクセスする方式に移行しました。これにより、以下のメリットがあります。

- ユーザーグループ毎にメンバーに apache を追加する必要がない
- ファイルシステム上のオーナー、パーミッションに従ったアクセス制限となる
- ファイルアクセスのセキュリティを緩和する必要がない

2.6.4 cd の階層数制限

ホームディレクトリから上位に上れる階層数を制限できます。上記 sudo との組み合わせにより、ユーザーグループ間のファイルアクセスを緻密に制御できますのでセキュリティを向上できます。

2.6.5 ドットで始まるファイルの非表示（管理者向け）

ドットで始まるファイルを表示、非表示を選択できます。現状では、システム全体での設定であり、ユーザー毎には変更できません。

2.7 その他の改善

2.7.1 計算ノードの属性値変更（管理者向け）

ホストコントロールの画面で、個々の計算ノードの制御パラメータ（ホスト属性値）を手動で変更可能になりました。運用中の設定変更、デバッグなどに便利になりました。

2.7.2 ACL の拡張（管理者向け）

conf/acl ディレクトリの設定ファイルとして、prog.acl が増えました。これによって、ユーザー、グループによって使用できるプログラムを制限できます。

2.7.3 エージェント設定ファイルの反映操作の自動化（管理者向け）

従来バージョンでは、エージェント設定ファイルの更新を反映するためには、stop 命令を発行して inactive 状態への遷移を確認したのちに、start 命令を発行する必要がありました。新バージョンでは、reload 命令が導入されました。reload 命令を発行すると、inactive 状態への遷移から、start 命令による operational 状態への遷移までを自動的に行います。

3 バグ修正

3.0.1 ノード間並列実行時の stop 操作の不具合修正（管理者向け）

修正前：(5.4 までが持つバグ)
ノード間並列ジョブを実行しているエージェントに対して stop 命令を発行した場合に、mpirun を実行したノードのエージェントは cleaning 状態に遷移してジョブ実行の終了を待つが、もう一方のエージェントはジョブ実行終了を待たずに即座に inactive 状態に遷移してしまう。
修正後：(5.5 で修正)
ノード間並列ジョブを構成するすべてのエージェントは、stop 命令を受け取ると、必ず cleaning 状態に遷移し、ジョブ終了を待って inactive 状態に遷移します。

以上