

ANCL

# ShareTask 5.5

(doc#02)

エージェント

インストールマニュアル

アンクル  
2014 年 2 月



## 目次

1	はじめに	5
2	対応 OS と Java 実行環境について	6
3	ファイルの配備手順	7
3.1	インストールファイルを取得する	7
3.2	インストールディレクトリを決める	7
3.3	インストールファイルを展開する	8
3.4	make を実行する	8
3.5	Java のコマンドパスを設定する	8
4	設定	10
4.1	設定ファイル stagent-boot.conf を編集する	10
4.2	ジョブ実行ディレクトリを作成する	11
4.3	ログファイルを設定する	12
4.4	スプールディレクトリを設定する	13
4.5	エージェントの自動起動を設定する	13
5	エージェントの起動と終了	15
5.1	Linux の場合	15
5.2	Windows の場合	15
6	ジョブのポーリングを設定する	16
6.1	設定ファイルを作成・編集する	16
6.2	設定ファイルの例	16
6.3	ジョブポーリング設定ファイルのパラメーター	17
6.4	設定ファイルの例 2 〈Linux 計算ノードの場合〉	20
6.5	設定ファイルの例 3 〈Windows 計算ノードの場合〉	21
7	ポーリング設定をエージェントに配布する	22
7.1	操作手順	22
7.1.1	設定ファイルを計算ノードに割り当てる	22
7.1.2	エージェントを不活性状態 (inactive) にする	22
7.1.3	エージェントを活性状態 (operational) にする	23

8	問題を解析する	24
9	ジョブへ継承される環境変数	25
9.1	注意	25
9.2	ボーリング設定ファイルの例	25
9.3	mpirun の例	25
9.4	関連	25

## 図目次

7.1.1	エージェントコントロールリスト	23
-------	-----------------	----

## 表目次

4.2.1	配下のディレクトリ	11
4.2.2	設定ファイルの編集	12
4.3.1	Log4J 設定ファイル	12
4.4.1	スプールディレクトリ	13
5.1.1	エージェント起動の引数	15
6.3.1	ジョブボーリング設定ファイルのパラメーター	17
9.0.1	エージェントから継承される環境変数	25

## 1 はじめに

ShareTask システムは、ユーザーからジョブの登録を受け付けるジョブ仲介サーバーと、ジョブ仲介サーバーからジョブを取り込んで実行するエージェントとから構成されます。

この文書では、エージェントを計算ノードにインストールする方法について説明します。

## 2 対応 OS と Java 実行環境について

本ソフトウェア (エージェント) が対応する OS は、以下の通りです。

- Linux
- Windows
- Mac OS X

本ソフトウェア (エージェント) は、Java SE の上に実現されていますので、Java SE 実行環境 (JRE: Java Runtime Environment, 1.4.2 以上) が必要です。

インストール CD には、JRE が同梱されていませんので、

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

から、エージェントをインストールしようとするコンピューターの OS に適合した JRE をダウンロードして、インストールしてください。

2011 年 7 月現在、JRE の最新版は、

Java SE Runtime Environment 6u26

です。

まだ実績は少ないですが、OpenJDK でも動作します。OpenJDK のインストールは、以下のとおりです。

```
# yum install java-1.6.0-openjdk
```

OpenJDK については、以下をご覧ください。

<http://openjdk.java.net>

### 3 ファイルの配備手順

#### 3.1 インストールファイルを取得する

エージェントのインストール物件一式は、インストール CD の下記ファイルです。

CLIENT/sharetask\_client-{version}.zip

エージェントは Java SE の上に実現されていますので、Java SE 実行環境 (JRE: Java Runtime Environment) が必要です。インストール CD には、JRE が同梱されていませんので、

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

から、エージェントをインストールしようとするコンピューターの OS に適合した JRE をダウンロードしてインストールしてください。

2011 年 7 月現在、JRE の最新版は、

Java SE Runtime Environment 6u26

です。

#### 3.2 インストールディレクトリを決める

エージェントは、任意の場所をインストールディレクトリにできますが、

/usr/local/sharetask

を標準としています。

アプリケーション一般を /opt の下に配置する慣習もありますので、

/opt/sharetask

としてインストールしてもかまいません。その場合は、

/usr/local/sharetask -> /opt/sharetask

とシンボリックリンクを張るとよいでしょう。

もし、どうしても /usr/local/sharetask 以外に配置したい場合は、インストールファイルを展開したのち、bin\_agent/stagent.sh の 11 行目の

```
export SHARETASK_HOME=/usr/local/sharetask
```

を修正してください。

### 3.3 インストールファイルを展開する

インストールファイル `sharetask_client-{version}.zip` を解凍展開します。

この zip ファイル内のディレクトリ構造は、`sharetask` をトップディレクトリとしています。

```
# cd /opt
# unzip -q /media/cdrom/CLIENT/sharetask_client_4.5.0.zip
# ln -s /opt/sharetask /usr/local/sharetask
```

補足: `/media/cdrom` の部分は、CD のマウントディレクトリです。各自の環境に応じて読み換えてください。

### 3.4 make を実行する

インストールファイルを展開して得られたディレクトリ `sharetask` の直下にある `Makefile` を実行します。

```
# make install_agent
```

これで、ファイルパーミッションの設定・シンボリックリンクの生成などが自動的に行われます。

もし、NFS 等の共有ファイルシステム上にエージェントのインストールディレクトリを配置し、複数の計算ノードでエージェント実行環境を共有する場合は、2 台目以降の計算ノードでは以下のコマンドを実行してください。

```
# make install_agent_slave
```

### 3.5 Java のコマンドパスを設定する

#### 注意

バージョンアップのために上書きインストールした場合は、この節はスキップしてください。

```
/usr/local/sharetask/etc/stagent.env
```



は、エージェント起動スクリプト内で使用される変数を設定するファイルです。このファイルで Java コマンドのパスを表す変数 SHARETASK\_JVM の値を設定します。

はじめてエージェントをインストールする場合は、

```
/usr/local/sharetask/etc/stagent.env.template
```

を

```
/usr/local/sharetask/etc/stagent.env
```

にファイル名を変更した上で、SHARETASK\_JVM の値がインストール環境の Java と整合するように編集してください。OpenJDK の場合は、一般的には

```
SHARETASK_JVM = /usr/bin/java
```

となります。

#### 注意

GNU の Java を呼び出さないように注意してください。エージェントは、GNU の Java では動きません。java -version で確認するとよいでしょう。

```
% /usr/bin/java -version
```

```
java version "1.6.0_20"
```

```
OpenJDK Runtime Environment (IcedTea6 1.9.8) (rhel-1.22.1.9.8.el5_6-i386)
```

```
OpenJDK Server VM (build 19.0-b09, mixed mode)
```

のように表示されれば大丈夫です。

## 4 設定

### 4.1 設定ファイル stagent-boot.conf を編集する

**注意**

バージョンアップのために上書きインストールした場合は、この節はスキップしてください。

`/usr/local/sharetask/conf/stagent-boot.conf`

は、エージェントが接続するジョブ仲介サーバーの URL を設定するファイルです。

はじめてインストールする場合は、

`/usr/local/sharetask/conf/stagent-boot.conf.template`

を

`/usr/local/sharetask/conf/stagent-boot.conf`

にファイル名を変更した上で、以下の編集を行ってください。

`/usr/local/sharetask/conf/stagent-boot.conf` を環境にあわせて編集します。

```
define_server {  
    base_url = http://my-server-host/stask  
    name = my-server-name  
    priority = 0  
}
```

- `my-server-host` の部分は、ジョブ仲介サーバーが稼働するホストの IP アドレスあるいはホスト名です。
- `my-server-name` の部分は、ジョブ仲介サーバーにつける任意の名前です。管理画面上での表示名となります。ホスト名と同一でかまいません。
- `priority` は、ジョブ仲介サーバーが冗長構成をとっている場合に、優先順位を指定するゼロ以上の整数値です。ゼロが最も優先順位が高いです。

補足： `define_server` セクションは、複数個記述できます。

2 個以上記述された場合、priority の値で指定された順にジョブ仲介サーバーへの接続を試みます。

priority の値が同一のものが 2 個以上記述された場合、接続を試みる順番は不定です。

## 4.2 ジョブ実行ディレクトリを作成する

Linux の場合、この節の操作は、「3.3 インストールファイルを展開する」で

```
make install_agent
```

によって自動的に行われていますので、スキップしていただいてもかまいません。

エージェントは、ジョブを実行する際にワークディレクトリを動的に作成し、このディレクトリの中に入力ファイルを展開することにより、アプリケーション実行に必要なディレクトリ環境を用意します。そして、このワークディレクトリをカレントディレクトリとしてアプリケーションを実行します。

このワークディレクトリは、SHARETASK\_WORK で指定されるディレクトリの配下のサブディレクトリとして生成されます。

表 4.2.1 配下のディレクトリ

ディレクトリ	用途
\$SHARETASK_WORK/auth	ジョブ仲介サーバーへのアクセス認証情報ファイルを格納する
\$SHARETASK_WORK/jobs	ジョブ実行のためのワークディレクトリを生成するベースディレクトリ

この SHARETASK\_WORK の変更には、表 4.2.2 の設定を編集します。

例：

```
SHARETASK_WORK=/tmp/sharetask  
set SHARETASK_WORK=D:\sharetask\spool
```

表 4.2.2 設定ファイルの編集

OS	設定ファイル	編集する行
Linux	\$SHARETASK_HOME/etc /stagent.env	SHARETASK_WORK=/var/spool /sharetask
Windows	%SHARETASK_HOME%/bin_agent_win /stagent.bat	set SHARETASK_WORK = %SystemDrive%\sharetask\spool

Linux の場合は、以下の操作でディレクトリを作成してください。

```
# mkdir -p /var/spool/sharetask/{auth,jobs}
```

Windows の場合は、以下の操作でディレクトリを作成してください。

```
mkdir D:\sharetask\spool
```

### 4.3 ログファイルを設定する

エージェントの挙動のロギングについては、Log4J を使用しています。ログファイルのパス設定は、Log4J の設定ファイルで行います。

Log4J の設定ファイルは、OS 毎に表 4.3.1 のように用意されていますので、対応するファイルを編集してください。

表 4.3.1 Log4J 設定ファイル

OS	Log4J の設定ファイル	編集する行
Linux	\$SHARETASK_HOME/etc/log4j.prop	log4j.appender.file.File =/var/log/sharetask-agent.log
Windows	\$SHARETASK_HOME/etc/ log4j_win.prop	log4j.appender.file.File =C:\\sharetask\\var\\log\\ sharetask-agent.log

\$SHARETASK\_HOME は、標準のインストールでは /usr/local/sharetask です。

はじめてインストールする場合は、

```
$SHARETASK_HOME/etc/log4j.prop.template (Linux)
```

`$SHARETASK_HOME/etc/log4j_win.prop.template` (Windows)  
を

`$SHARETASK_HOME/etc/log4j.prop` (Linux)

`$SHARETASK_HOME/etc/log4j_win.prop` (Windows)

にファイル名を変更した上で、上記の編集を行ってください。

バージョンアップのために上書きインストールした場合は、この節はスキップしてください。

#### 4.4 スプールディレクトリを設定する

エージェントがジョブを実行する際のワークディレクトリを作成したりするためのディレクトリを、スプールディレクトリと呼びます。

標準では、`/var/spool/sharetask` としています。

これを変更するには、`$SHARETASK_HOME/etc/stagent.env` の次の行を編集してください。

```
SHARETASK_WORK=${SHARETASK_WORK:=/var/spool/sharetask}
```

この配下に表 4.4.1 の 2 つのディレクトリを用意します。

表 4.4.1 スプールディレクトリ

ディレクトリ	用途
<code>/var/spool/sharetask/auth</code>	ジョブ仲介サーバーへのアクセス認証情報ファイルを格納する
<code>/var/spool/sharetask/jobs</code>	ジョブ実行のためのワークディレクトリを生成するベースディレクトリ

具体的には、次の操作でディレクトリを生成します。

```
# mkdir -p /var/spool/sharetask/{auth,jobs}
```

#### 4.5 エージェントの自動起動を設定する

Linux の場合、この節の操作は、「3.3 インストールファイルを展開する」で

```
# make install_agent
```

によって自動的に行われていますので、スキップしていただいてもかまいません。

OS ブート時に自動的にエージェントが起動し、シャットダウン時は自動的にエージェントが終了処理を行うように、`/etc/init.d/` に以下の手順で起動スクリプトを設定します。

```
# ln -s /usr/local/sharetask/bin_agent/sharetask-agent/etc/init.d/sharetask-agent  
# /sbin/chkconfig --add sharetask-agent
```

Windows の場合は、タスクスケジューラーに

```
%SHARETASK_HOME%\bin_agent_win\stagent.bat
```

を登録します。

## 5 エージェントの起動と終了

### 5.1 Linux の場合

エージェントの起動と終了は、起動スクリプト `/etc/init.d/sharetask-agent` で行います。

表 5.1.1 のような引数があります。

表 5.1.1 エージェント起動の引数

操作	引数	補足
エージェントを起動する	<code>start</code>	
エージェントを終了する	<code>stop</code>	実行中のジョブがあればその終了を待つ
エージェントを強制終了する	<code>halt</code>	実行中のジョブがあれば強制終了する
エージェントを再起動する	<code>restart</code>	実行中のジョブがあればその終了を待つ
エージェントを強制再起動する	<code>hrestart</code>	実行中のジョブがあれば強制終了する
エージェントが稼働中かどうかを知る	<code>status</code>	

例：エージェントを終了する場合

```
# /etc/init.d/sharetask-agent stop
```

### 5.2 Windows の場合

エージェントを起動するには、

```
%SHARETASK_HOME%\bin_agent_win\stagent.bat
```

を起動します。

エージェントを終了するには、タスクマネージャーから `java.exe` を見つけて、これを終了させてください。

## 6 ジョブのポーリングを設定する

ジョブのポーリングを設定するには、ジョブ仲介サーバーの「エージェントコントロールリスト」画面を使います。

以下、設定ファイルとは、ジョブポーリング設定ファイルを意味します。

### 6.1 設定ファイルを作成・編集する

エージェントコントロールリスト画面を開きます。

新しく設定ファイルを作成する場合は、設定ファイル欄にファイル名を入力し、[new/edit] ボタンを押します。

例：mine.conf

"New/Edit config file"というウィンドウが開きますので、設定を記述します。設定例は、次節 6.2 をご覧ください。

### 6.2 設定ファイルの例

```
define_resource {
    max_procs = 1
    polling_interval = 60
    keepalive_interval = 15
}
define_program {
    name = myappl1
    version = single
    executable = /opt/myappl1/myprog1
    workdir_parent = /var/spool/sharetask/jobs
    mpi = false
}
define_program {
    name = myappl2
```



```
version = para
executable = /opt/myappl2/myprog2
workdir_parent = /var/spool/sharetask/jobs
mpi = true
}
```

この設定例では, myappl1 と myappl2 というアプリケーションについてのジョブのポーリングを設定しています.

myappl1 は逐次処理型実行であり, myappl2 は並列実行です (mpi).

キュー (ジョブクラス) の名前は, それぞれ single と para です (version).

実行ファイルのパスを executable で記述します.

workdir\_parent は, ジョブを実行するためのテンポラリディレクトリを作成するパス (親ディレクトリ) です.

## 6.3 ジョブポーリング設定ファイルのパラメーター

表 6.3.1 ジョブポーリング設定ファイルのパラメーター

キーワード	意味	値
define_resource	共通パラメーターを記述するセクションの開始を宣言	
max_procs	この計算ノードから払い出せる CPU コアの最大数	正の整数
polling_interval	ジョブ仲介サーバーにジョブを問い合わせる間隔	正の整数 (秒)
keepalive_interval	ジョブ仲介サーバーにジョブ実行状態を通知する間隔	正の整数 (秒)
rsrc_monitor	リソース情報を取得する外部プログラムのパス	パス文字列
check_filesystem	ファイルシステム空き容量取得対象のパス (ドライブ)	パス文字列
define_program	ジョブキューを記述するセクションの開始を宣言する	
name	アプリケーションの名前	文字列
version	ジョブキューの名前	文字列
executable	アプリケーションの実行ファイルのファイルパス	パス文字列
wrapper	ジョブ実行を監視制御するためのラッパープログラム	パス文字列
time_limit	ジョブ実行の打ち切り時間 (wrapper が必要)	正の整数 (秒)
workdir_parent	ジョブ実行のテンポラリディレクトリの親ディレクトリ	パス文字列

キーワード	意味	値
mpi	MPI 並列をサポートするか	true   false
define_server	ジョブ仲介サーバーへの接続情報を記述するセクション	
base_url	ジョブ仲介サーバーのベース URL	[http https]://.../stask
name	ジョブ仲介サーバーの管理上の識別名	文字列
priority	冗長構成の際の優先順位	ゼロ以上の整数

- executable の値が、/ 以外の文字から始まる場合は、ShareTask のインストールディレクトリ (\$SHARETASK\_HOME) からの相対パスとして解釈される。
- wrapper の値が、/ 以外の文字から始まる場合は、ShareTask のインストールディレクトリ (\$SHARETASK\_HOME) からの相対パスとして解釈される。
- wrapper として、\$SHARETASK\_HOME/bin\_agent/jobwrapper0 (シェルスクリプト) が提供されています (「6.4 設定ファイルの例 2」を参照)。
 

```
wrapper = bin_agent/jobwrapper0
```
- time\_limit は、wrapper として bin\_agent/jobwrapper0 を設定したときに有効になる。
- wrapper = bin\_agent/jobwrapper0 を設定しているにもかかわらず、time\_limit を設定しないときは打ち切り時間は設定されず、無限に実行される。
- workdir\_parent の値が、/ 以外の文字から始まる場合は、ShareTask のワークディレクトリ (\$SHARETASK\_WORK) からの相対パスとして解釈される。
- \$SHARETASK\_WORK は、<インストールディレクトリ>/etc/stagent.env の中で定義される。
- \$SHARETASK\_HOME は、<インストールディレクトリ>/bin\_agent/stagent.sh の 11 行目付近で定義される (近日、改善予定)。
- #以降はコメントとして無視される。

## 6.4 設定ファイルの例 2 〈Linux 計算ノードの場合〉

```
define_resource{
    max_procs = 1
    polling_interval = 60
    keepalive_interval = 15
    rsrc_monitor = /usr/local/sharetask/bin_agent/rsrc_monitor.sh
    check_filesystem = /
}
define_program{
    name = myappl1
    version = single
    executable = /opt/myappl1/myprog1
    wrapper = bin_agent/jobwrapper0
    time_limit = 86400
    workdir_parent = /var/spool/sharetask/jobs
    mpi = false
}

define_program {
    name = myappl2
    version = para
    executable = /opt/myappl2/myprog2
    wrapper = bin_agent/jobwrapper0
    time_limit = 86400
    workdir_parent = /var/spool/sharetask/jobs
    mpi = true
}
```

- wrapper の使用により、時間制限 time\_limit を指定できる。

## 6.5 設定ファイルの例 3 〈 Windows 計算ノードの場合 〉

```
define_resource {  
    max_procs = 4  
    polling_interval = 15  
    keepalive_interval = 15  
    rsrc_monitor = C:\\sharetask\\bin_agent_win\\rsrc_monitor.bat  
    check_filesystem = C:  
}  
  
define_program {  
    name = dispat  
    version = Sw  
    executable = C:\\sharetask\\appl\\dispat.bat  
    workdir_parent = C:\\sharetask\\var\\spool\\sharetask\\jobs  
}
```

## 7 ポーリング設定をエージェントに配布する

ジョブポーリング設定をエージェントに配布するには、ジョブ仲介サーバーの「エージェントコントロールリスト」画面を使います。

以下、設定ファイルとは、ジョブポーリング設定ファイルを意味します。

### 7.1 操作手順

#### 7.1.1 設定ファイルを計算ノードに割り当てる

「エージェントコントロールリスト」画面の設定ファイル欄に、配布した設定ファイル名を入力します。「\*」を入力するか、ダブルクリックで設定ファイルの全候補がリストされるので、そこから選択することもできます。また、はじめの文字を入れるだけでも、該当する設定ファイルの候補がリストされますので、そこからも選択できます。

[set] ボタンを押すと、エージェント一覧の設定ファイル欄に、選択した設定ファイルが設定されます。

もし、一部のエージェントに対してのみ適用したい場合は、画面上部のフィルター欄に選択条件を入力して [フィルター] ボタンを押すと、絞り込みが行われますので、その上で [set] ボタンを押すと、絞り込んだエージェントに対してのみ適用されます。

#### 7.1.2 エージェントを不活性状態 (inactive) にする

エージェント一覧のステータス欄が `inactive` になっていれば、この節の操作は不要です。次節へ進んでください。

「エージェントコントロールリスト」画面の [stop] ボタンを押します。エージェント一覧のコマンド欄が `stop` に変わります。数十秒以内に、ステータス欄が `inactive` に変わります。

(エージェントは、60 秒間隔で定期的にジョブ仲介サーバーに状態遷移要求を問い合わせていますので、最長 60 秒待てば状態が遷移します)

`inactive` になったエージェントは、それまでの設定情報を破棄して不活性な状態になっています。

### 7.1.3 エージェントを活性状態 (operational) にする

「エージェントコントロールリスト」画面の [start] ボタンを押します。エージェント一覧のコマンド欄が start に変わります。数十秒以内にステータス欄が operational に変わります。operational になったエージェントは、設定ファイルを読み込み、その設定にしたがってジョブのポーリングを開始しています。

エージェント コントロール リスト ようこそ sakto (admin) さん [Logout]

≪メインメニュー≫

[ホーム](#)
[設定](#)
[エージェント表示画面](#)

エージェントコントロール
 start stop halt suspended
設定ファイル: e
start
リセット
refresh
help

検索欄に入力して検索/右クリックでメニューを開く/左クリックでメニューを開く

≪ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000

ID	ホストグループ	エージェント名	ステータス	実行中ジョブ	ジョブタイプ	コマンド	設定ファイル	接続サーバー	接続時刻
12		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
11		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 00:00:51
10		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:03
9		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
8		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
7		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
6		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
5		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
4		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
3		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
2		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02
1		172.25.0.78	extra001		start	start	extra01.conf	demo	2011-09-01 14:55:02

440-master (2011) ANCL, Inc.

図 7.1.1 エージェントコントロールリスト

## 8 問題を解析する

期待通りに動作しない場合は、ログファイルを解析します。

ログファイルの所在は、「4.3 ログファイルを設定する」を参照してください。



## 9 ジョブへ継承される環境変数

wrapper を使用している場合において、ジョブが実行されるとき、次の表 9.0.1 の環境変数がエージェントから継承されます。

表 9.0.1 エージェントから継承される環境変数

環境変数名	意味
SHARETASK_JOB_ID	ジョブ ID
SHARETASK_PWD	ジョブの実行ディレクトリパス
SHARETASK_OUTDIR	制御ファイルを生成するディレクトリパス
SHARETASK_STDIN	ジョブの標準入力に接続されたファイルのパス
SHARETASK_STDOUT	ジョブの標準出力に接続されたファイルのパス
SHARETASK_STDERR	ジョブの標準エラー出力に接続されたファイルのパス
SHARETASK\_HOSTS	ジョブに割り当てられた計算ノードリストのファイルパス (MPI のマシンファイル)
SHARETASK\_NP	ジョブに割り当てられた CPU コア数 (stsubmit -np オプションで指定した CPU コア数と同じ)

### 9.1 注意

これらの環境変数は、以降の例のように、wrapper として bin\_agent/jobwrapper0 を指定したときに継承されます。

### 9.2 ポーリング設定ファイルの例

### 9.3 mpirun の例

ジョブとして実行されるスクリプトの中で、以下のように記述します。

```
mpirun -np $SHARETASK_NP -machinefile $SHARETASK_HOSTS $*
```

### 9.4 関連

stsubmit コマンド

```
define_program {  
    name = myappl2  
    version = para  
    executable = /opt/myappl2/myprog2  
    wrapper = bin_agent/jobwrapper0  
    time_limit = 86400  
    workdir_parent = /var/spool/sharetask/jobs  
    mpi = true  
}
```

以上