第1版 　 2016年4月27日

RAS監視を行い、Mail通知する。障害検知時には、自動的に再起動をする。

RasEye/RasMail

内容

[1. はじめに 2](#_Toc449534890)

[2. 方式 2](#_Toc449534891)

[2.1. 監視・通知・再起動(RasEye) 2](#_Toc449534892)

[2.2. 多重化多段化自律方式（参考） 3](#_Toc449534893)

[3. 機能 4](#_Toc449534894)

[3.1. パスによる管理 4](#_Toc449534895)

[3.2. イベントパス 4](#_Toc449534896)

[4. APIとコマンド 5](#_Toc449534897)

[4.1. PFSRasRegister/PFSOpen/PFSWrite/PFSClose RAS対象資源登録API 5](#_Toc449534898)

[4.2. PFSRasActiveコマンド(単体用) 5](#_Toc449534899)

[4.3. PFSRasClusterコマンド(Cell用) 6](#_Toc449534900)

[4.4. RasEye 監視 6](#_Toc449534901)

[4.5. RasEyeAdm 管理コマンド 6](#_Toc449534902)

[4.6. RasMail メール通知/アドレス管理 7](#_Toc449534903)

[4.7. 生死の一覧参照 8](#_Toc449534904)

[5. 運用 8](#_Toc449534905)

[5.1. ntp環境 8](#_Toc449534906)

[5.2. メール環境 8](#_Toc449534907)

[5.3. 遠隔操作 8](#_Toc449534908)

[5.3.1. 構成ファイル 8](#_Toc449534909)

[5.3.2. 生死確認及び強制正常終了コマンド 9](#_Toc449534910)

[5.3.3. セルのコマンドシェルテンプレート 9](#_Toc449534911)

[5.3.4. 遠隔操作コマンドシェル 19](#_Toc449534912)

# はじめに

　Ras監視依頼を行った対象のイベントを監視し、登録されたメールアドレスにメール通知を行う。

　また、障害検知時には自動再起動を行うことができる。対象をRasEyeとすれば永久監視システムが構築できる。

# 方式

## 監視・通知・再起動(RasEye)

RASセル

対象

RasEye

RasEye

ﾒｰﾙﾎﾞｯｸｽA

ｲﾍﾞﾝﾄ通番ﾌｧｲﾙ

更新時刻

ｲﾍﾞﾝﾄ通番

ﾒｰﾙﾎﾞｯｸｽB

RasMail

RasAutonomic

RasMail

RasAutonomic

ｲﾍﾞﾝﾄ

登録

変更

設定

設定

　監視対象はRASセルにイベント資源を登録する。RasEyeはイベント資源に対してイベント発生時の通知指示を設定する。

RasMailにはイベント受信時の通知先メールアドレスを登録する。

RasAutonomicは、対象毎にRasEyeに登録されるプロセス名であり、イベント資源が削除されたときに呼び出される。

　イベント資源にはイベント通番が設定されている。イベント発生時、通番をインクリメントし、設定されているRasEye全てにイベント通知を行う。RasEyeはイベント受信時、該当イベントの通番ファイルを調べ、送信済みであれば破棄する。そうでなければ、イベント通番ファイルを更新し、RasMailに通知を行う。また、削除イベント時にはRasAutonomicを呼び出す。なお、各RasEyeの通番チェックは順番に行われるので飛び越しは発生しない。ただし、再起動の場合、古い通番が欠落して、飛び越しの可能性がある。また、全てのRasEyeが停止すると、ファイル保存通番が残ってしまう。

　そこで、保存通番に更新時刻を付与し、参照事項と一定時間内（個々の時刻の差も考慮し10秒程度）の範囲で競合チェックを行うことにする。

まとめるとつぎのようになる。

1. 保存時間が一定時間より過去であれば、処理を実行する。
2. 通番が保存通番以下であれば処理済なので、処理を行わない。
3. 保存通番との差が2以上であれば再試行する。

## 多重化多段化自律方式（参考）

　前節の方式を一般化すると多重化多段化の自律システムが実現できる。これは、通番ファイルがPaxosCore(css)で共有されている。

Lock

Read

Send/restart

Write

Unlock

通番

更新時刻

Lock

Read

Send/restart

Write

Unlock

Lock

Read

Send/restart

Write

Unlock

通番

更新時刻

Lock

Read

Send/restart

Write

Unlock

　なお、星印で重複送信が発生するが、通番も送信すれば受信側で重複の認識でき、対処できる。

# 機能

## パスによる管理

　イベント、ファイル及びロックは木構造ではなく、パスを識別子として管理される。すなわち、パスそのものがタグであり、フラットに管理されている。

A/B/C

A/D

　木構造の全要素に対応してあるわけでない。木構造へのマッピングはタグを解析して行う。

　イベントはこのパスタグに設定される。

　ファイルはバックストーレージとして木構造のUNIXファイルシステムが使われる。このためのUNIXファイルシステムとの整合性確保に、ディレクトリ操作、ファイル操作がある。なお、ファイルはファイルキャッシュを介したセグメント化されたメタファイルとして実装されている。

　ロックも同様にパスタグとして管理される。

## イベントパス

　イベントはディレクトリパスでのファイル操作である。

　パスは、

サービス（ロードモジュール名）

セルorクラスター

ID

の三層であり、IDをエフェメラルファイルとしている。

RAS用にサービス配下のディレクトリをそのまま使用すると、固有サービスのファイル操作イベントも取得してしまい処理が煩雑になる。そこで、RAS用エフェメラルファイルは、“RAS/”をプリフィックスとする。

また、RAS監視用の通番ファイルは、“/RasEye/監視クラスターファイル”をサフィックスとする。サフィクスにRasEyeを挟んで監視クラスターファイルの更新イベント発生を回避している。

さらに、RasEyeは自己監視ができる。

例）

RAS/MemCache/SPACE/0,1,2,x\_BAK… 監視対象ファイル

RAS/MemCache/SPACE/RasEye/TEST 監視クラスターロック・ファイル

RAS/RasEye/RAS/0,1、0\_BAK,1\_BAK 監視対象

RAS/RasEye/RAS/RasEye/TEST 監視クラスターロック・ファイル

TESTはRasEyeの監視クラスターである。

　監視対象ファイルはエフェメラルファイル（一時ファイル）であり、削除時には”\_BAK”のサフィックスファイルとして保存される。

　エフェメラルファイルの作成、更新、削除でイベントが発生し、メール通知される。

エフェメラルファイルは、メールの添付ファイルとして通知される。本ファイルに詳細、統計情報等を記録することもできる。

# APIとコマンド

## PFSRasRegister/PFSOpen/PFSWrite/PFSClose RAS対象資源登録API

int

PFSRasRegister( struct Session \*pSession, char \*pPath, char \*pEntry )

File\_t\*

PFSOpen( struct Session\* pSession, char \*pathname, int Flags )

int

PFSWrite( File\_t\* pF, char\* pBuff, size\_t Len )

int

PFSClose( File\_t\* pF )

説明

RASセッションにエフェメラルファイル(pPath/pEntry=pathname)を作成、更新する。

パス名には、”RAS/”プリフィックスを前置して呼び出す。

Paxosセル以外は、これらのAPIによる。

## PFSRasActiveコマンド(単体用)

「PFSRasActive ras\_cell path　file」

ras\_cellのpathにfile（セルのidに相当）の登録を指示する。

Pathは「RAS/path」とする。

このパスディレクトリにエフェメラルファイルfileが作成される。

## PFSRasClusterコマンド(Cell用)

　「PFSRasCluster cell id ras\_cell path」

Cellのidに対して、ras\_cellのpathに登録することを指示する。

Pathは「RAS/ロードモジュール名/cell」とする。

このパスディレクトリにエフェメラルファイルidが作成される。

また、自己監視も実行し、障害検知時にはリーダー選びも直ちに行う。

## RasEye 監視

コマンド

「RasEye　[オプション]　cluster　Id」

Cluster　クラスター(セル相当)

Id 数字

オプション

-d　ddd 通知済みイベント通番の無効判定時間（ディフォルト：10秒）

説明

　Id（管理用ポート）とクラスターグループで起動する。起動後、管理用ポートを開設し、コマンドを待つ。

　イベント設定解除、メールアドレスの登録削除等の管理コマンドを受け付ける。

イベント受信後、メール通知済みイベント通番を調べ、メール通知を行う。

監視対象の設定はRasEyeによるが、RasEyeAdmの設定コマンドでRasEyeはRasセルと通常及びイベントセッションを張り、callbackスレッドを起動し、PFSEventDirSetで設定を行う。

## RasEyeAdm 管理コマンド

本コマンドは、RasEyeの管理ポートを介して、登録削除、情報取得を行う。

コマンド

RasEyeAdm cluster id set cell path mail “autonomic　…”

RasEyeAdm cluster id unset cell path

RasEyeAdm cluster id unset\_all

RasEyeAdm cluster id ras\_dump

RasEyeAdm cluster id regist\_me

RasEyeAdm cluster id unregist\_me cell

RasEyeAdm cluster id regist\_dump

RasEyeAdm cluster id session\_dump

RasEyeAdm cluster id stop

RasEyeAdm cluster id log

説明

Cluster 監視グループ

Id 監視グループ内のIad

Set/unset RASセル(cell)内の監視パス（path）に対する設定解除

Set時にはメイラーとautonomicも登録する

本コマンドでは、プリフィックスRASは付与される

Regist\_me/unregist\_me RASセル（cell）に監視対象であることを通知する

Setと組み合わせれば相互監視ができる

Xxx\_dump 情報を取得する

Stop 正常終了させる

Log バッファーにあるログをフラッシュする

Mail 「Mail　cluster　id　event　本文　[添付ファイル]」で起動される

autonomic 「autonomic cluster –r ras\_cell –i Id …」で起動される

cluster/cell 対象クラスター・セル

-r　ras\_cell RASセル

-I　Id

-s　seq イベント通番

-m　mode イベントモード

設定は、次のように行う。

「RasEyeAdm 監視クラスター id set ras\_cell VV/vv RasMail “cell\_autonomic.sh　tdtdVV　vv　css　xjPing“」

tdtgVV、vv、css、xjPing固有の位置パラメータは、set時に登録する。

## RasMail メール通知/アドレス管理

　管理コマンドも統合した。

コマンド

「RasMail　cluster　id」 デーモンとなる。

「RasMail　cluster　id　add/del　MailAddr」 メールアドレスの設定解除

「RasMail　cluster　id　dump」 メールアドレスの参照

「RasMail　cluster　id　event　本文　[添付ファイル]」 イベント

説明

　RasMailは、本文と添付ファイルでメールを送信する。

## 生死の一覧参照

現在、一覧参照はcssのpath参照コマンドによる。

「css\_adm.sh css service id path」

「PFSProbe cell id path」

による。

# 運用

## ntp環境

　ntpで時刻あわせを行う。

## メール環境

　Mailutils、sharutilsをインストールする。

.mailrcにrelayを設定する。

例)

set smtp=redbird.tritech.co.jpを

## 遠隔操作

### 構成ファイル

構成ファイルには、セル用とクラスター用がある。

セル用

# id host tcp paxos udp bin root

0 paxos03 4020 paxos03 4030 ~/bin /pss/css/0

1 paxos04 4021 paxos04 4031 ~/bin /pss/css/1

2 paxos05 4022 paxos05 4032 ~/bin /pss/css/2

3 paxos06 4023 paxos06 4033 ~/bin /pss/css/3

クラスター用

#　id　host　bin　root

0 paxos03 　~/RAS　 /pss/RAS/0

1 paxos04 　~/RAS　/pss/RAS/1

id Id

host ホスト名

bin ロードモジュールの配置先

root 動作ディレクトリ、ログ保存先

tcp TCPポート番号(host:tcp)

paxos paxosセルメンバー

udp UDPポート番号(paxos:udp)

　構成ファイルは、環境変数PAXOS\_CELL\_COFIGの指定先ディレクトリ（”/”で終端）に「XXX.conf」あるいはHOMEに「\_XXX.conf」、「.config/cell\_config/xxx.conf」あるいは「/etc/cell\_config/XXX.conf」あるいは「/etc/XXX/server.conf」に設定する。

### 生死確認及び強制正常終了コマンド

　監視対象となるセル及びクラスターは生死確認プロトコルを実装する。セルはＰａｘｏｓレベルで共通ツール(PaxosAdminコマンド)として整備されているが、クラスターは個別に用意しなければならない。

### セルのコマンドシェルテンプレート

「cell\_start.sh load cell [-i id] [-r ras\_cell]」

Loadはサービスの実行モジュールを指定する。

Cellの起動されていないプロセスを起動する。

Idが指定されれば当該idを起動する。

Ras\_cellが指定されれば監視被対象登録を行う。また、相互監視を行い、障害検知時には直ちにリーダ選択を行う。また、ras\_cellにcellを指定すると自己監視となる。

「cell\_stop.sh load　cell comment [-i id]」

Cellの起動されているプロセスを強制正常終了する。SHUTDOWNファイルが作成される。ただし、リーダは最後に強制正常終了される。

Idが指定されれば当該idを強制正常終了する。

本コマンドによる終了は自動的な再起動と対象としない。

「cell\_restart.sh load cell/cluster [-i id] [-r ras\_cell]」

Cellの起動されていないプロセスを起動する。

CellではSHUTDOWNファイルを調べ、最新のidを最初に「TRUE　seq」で起動する。

その他は、ＳＨＵＴＤＯＷＮファイルがあれば、「TRUE　seq」、なければ「FALSE」で起動する。

「cell\_autonomic.sh cell [-i id] [-r ras\_cell] …」

本shellは、RasEyeから起動される。

指定idに SHUTDOWNファイルがないときに、「FALSE」で起動する。

強制正常終了時は、何かをしたい場合と考えられるので再起動を行わない。

Cell\_start.sh

#!/bin/sh

#

# Analyze parameters

#

args=$#

if [ $args -lt 2 ]

then

echo "USAGE:cell\_start.sh load cell [-i id] [-r ras\_cell]"

exit 1

fi

load=$1

shift

cell=$1

shift

while [ "$1" != "" ]

do

case "$1" in

"-r") shift; ras\_cell=$1; shift ;;

"-i") shift; ID=$1; shift ;;

\* ) shift;;

esac

done

env0="LOG\_SIZE=20000"

files=20

conf=~/bin/$cell.conf

#

# Active check

#

while read id paxos udp host session bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ "X$ID" != "X" -a "$ID" != "$id" ]; then continue; fi

ssh -n $host "$bin/PaxosAdmin $cell $id is\_active > /dev/null 2>&1"

ret=$?

if [ $ret != 0 ]; then

IDs=$IDs"$id "

eval bin\_$id=\$bin

eval root\_$id=\$root

eval host\_$id=\$host

fi

done < $conf

#

# Start

#

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="$env0 PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host \"cd $root;$env $bin/$load $cell $id $root TRUE 0 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files\""

ssh -f $host "cd $root;$env $bin/$load $cell $id $root TRUE 0 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files"

sleep 1

done

#

# Register

#

if [ "X$ras\_cell" != "X" ]; then

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host $env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

ssh -f $host "$env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

done

fi

exit 0

Cell\_stop.sh

#!/bin/sh

#

# Analyze parameters

#

args=$#

if [ $args -lt 2 ]

then

echo "USAGE:cell\_start.sh load cell [-i id] [-r ras\_cell]"

exit 1

fi

load=$1

shift

cell=$1

shift

while [ "$1" != "" ]

do

case "$1" in

"-r") shift; ras\_cell=$1; shift ;;

"-i") shift; ID=$1; shift ;;

\* ) shift;;

esac

done

env0="LOG\_SIZE=20000"

files=20

conf=~/bin/$cell.conf

#

# Active check

#

while read id paxos udp host session bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ "X$ID" != "X" -a "$ID" != "$id" ]; then continue; fi

ssh -n $host "$bin/PaxosAdmin $cell $id is\_active > /dev/null 2>&1"

ret=$?

if [ $ret != 0 ]; then

IDs=$IDs"$id "

eval bin\_$id=\$bin

eval root\_$id=\$root

eval host\_$id=\$host

fi

done < $conf

#

# Start

#

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="$env0 PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host \"cd $root;$env $bin/$load $cell $id $root TRUE 0 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files\""

ssh -f $host "cd $root;$env $bin/$load $cell $id $root TRUE 0 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files"

sleep 1

done

#

# Register

#

if [ "X$ras\_cell" != "X" ]; then

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host $env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

ssh -f $host "$env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

done

fi

exit 0

nw@paxos00:~/PAXOS/Paxos\_2/PFS/bin$ cat cell\_stop.sh

#!/bin/sh

if [ $# -lt 3 ]

then

echo "USAGE:cell\_stop.sh load cell comment [-i id]"

exit 1

fi

load=$1

shift

cell=$1

conf=$HOME/\_$cell.conf

shift

comment=$1

shift

while [ "$1" != "" ]

do

case "$1" in

"-i") shift; ID=$1; shift ;;

\* ) shift;;

esac

done

while read id paxos udp host session bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ "X$ID" != "X" -a "$ID" != "$id" ]; then continue; fi

env="PAXOS\_CELL\_CONFIG=$bin/"

ssh -n $host "$env $bin/PaxosAdmin $cell $id is\_active > /dev/null 2>&1"

ret=$?

if [ $ret = 0 ]; then

ssh -n $host "$env $bin/PaxosAdmin $cell $id is\_leader > /dev/null 2>&1"

ret=$?

if [ $ret = 0 -a ! $leader\_id ]; then

leader\_id=$id

leader\_host=$host

leader\_bin=$bin

else

IDs=$IDs"$id "

eval bin\_$id=\$bin

eval root\_$id=\$root

eval host\_$id=\$host

fi

fi

done < $conf

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

shutdown=$bin/PaxosSessionShutdown

echo "ssh -f $host $env $shutdown $cell $id $comment"

ssh -f $host "$env $shutdown $cell $id $comment"

done

if [ $leader\_id ]; then

env="PAXOS\_CELL\_CONFIG=$leader\_bin/"

shutdown=$leader\_bin/PaxosSessionShutdown

echo "LEADER:ssh -f $leader\_host $env $shutdown $cell $leader\_id $comment"

ssh -f $leader\_host "$env $shutdown $cell $leader\_id $comment"

fi

exit 0

#!/bin/sh

Cell\_restart.sh

#

# Analyze parameters

#

args=$#

if [ $args -lt 2 ]

then

echo "USAGE:cell\_restart.sh load cell [-i id] [-r ras\_cell]"

exit 1

fi

load=$1

shift

cell=$1

shift

while [ "$1" != "" ]

do

case "$1" in

"-r") shift; ras\_cell=$1; shift ;;

"-i") shift; ID=$1; shift ;;

\* ) shift;;

esac

done

env0="LOG\_SIZE=20000"

files=20

conf=~/bin/$cell.conf

#

# Check active

#

max=$((-1))

while read id paxos udp host session bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ "X$ID" != "X" -a "$ID" != "$id" ]; then continue; fi

ssh -n $host "$bin/PaxosAdmin $cell $id is\_active > /dev/null 2>&1"

ret=$?

if [ $ret != 0 ]

then

IDs=$IDs"$id "

eval bin\_$id=\$bin

eval root\_$id=\$root

eval host\_$id=\$host

scp $host:$root/SHUTDOWN /tmp/SHUTDOWN > /dev/null 2>&1

ret=$?

if [ $ret = 0 ]

then

read seq comment < /tmp/SHUTDOWN

num=$((seq))

if [ $max -eq -1 ] # first

then

max=$num; young=$id

elif [ $max -lt $num ]

then

diff=$((num-max))

if [ $diff -lt 2147483647 ]

then

max=$num; young=$id

fi;

else

diff=$((max-num))

if [ $diff -ge 2147483647 ]

then

max=$num; young=$id

fi;

fi

eval seq\_$id=$seq

eval comment\_$id=$comment

fi

fi

done < $conf

#

# Start youngest

#

if [ $young ]

then

eval bin='$bin\_'$young

eval root='$root\_'$young

eval host='$host\_'$young

env="$env0 PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host \"cd $root;$env $bin/$load $cell $young $root TRUE $max 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files\""

ssh -f $host "cd $root;$env $bin/$load $cell $young $root TRUE $max 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files"

sleep 1

fi

for id in $IDs

do

if [ $young ]; then

if [ $young = $id ]; then continue; fi

fi

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

eval seq='$seq\_'$id

env="$env0 PAXOS\_CELL\_CONFIG=$bin/"

if [ $seq ]

then

echo "ssh -f $host \"cd $root;$env $bin/$load $cell $id $root TRUE $seq 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files\""

ssh -f $host "cd $root;$env $bin/$load $cell $id $root TRUE $seq 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files"

else

if [ $young ]

then

echo "ssh -f $host \"cd $root;$env $bin/$load $cell $id $root TRUE 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files\""

ssh -f $host "cd $root;$env $bin/$load $cell $id $root TRUE 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files"

else

echo "ssh -f $host \"cd $root;$env $bin/$load $cell $id $root FALSE 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files\""

ssh -f $host "cd $root;$env $bin/$load $cell $id $root FALSE 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files"

fi

fi

sleep 1

done

#

# Register

#

if [ "X$ras\_cell" != "X" ]; then

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host $env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

ssh -f $host "$env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

done

fi

exit 0

Cell\_autonomic.sh

#!/bin/sh

#

# Analyze parameters

#

args=$#

if [ $args -lt 6 ]

then

echo "USAGE:css\_autonomic load cell -i id -r ras\_cell [-s seq] [-m mode] ..."

exit 1

fi

load=$1

shift

cell=$1

shift

while [ "$1" != "" ]

do

case "$1" in

"-i") shift; ID=$1; shift ;;

"-r") shift; ras\_cell=$1; shift ;;

"-s") shift; seq=$1; shift ;;

"-m") shift; mode=$1; shift ;;

"-B") shift; BIN=$1; shift ;;

\* ) shift;;

esac

done

if [ "X$ID" = "X" -o "X$ras\_cell" = "X" ]; then

exit 1

fi

env0="LOG\_SIZE=20000"

files=20

conf=~/bin/$cell.conf

#

# Check active

#

while read id paxos udp host session bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ $ID ]; then if [ $ID != $id ]; then continue; fi; fi

eval bin=$bin

env="PAXOS\_CELL\_CONFIG=$bin/"

ssh -n $host "$env $bin/PaxosAdmin $cell $id is\_active > /dev/null 2>&1"

ret=$?

if [ $ret != 0 ]

then

scp $host:$root/SHUTDOWN /tmp/SHUTDOWN > /dev/null 2>&1

ret=$?

if [ $ret != 0 ]

then

IDs=$IDs"$id "

eval host\_$id=\$host

eval bin\_$id=\$bin

eval root\_$id=\$root

fi

fi

done < $conf

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="$env0 PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host \"cd $root;$env $bin/$load $cell $id $root FALSE 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files\""

ssh -f $host "cd $root;$env $bin/$load $cell $id $root FALSE 3>&1 1>&2 2>&3 | $bin/LogFiles $load $files"

sleep 1

done

#

# Register

#

if [ "X$ras\_cell" != "X" ]; then

for id in $IDs

do

eval host='$host\_'$id

eval bin='$bin\_'$id

eval root='$root\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host $env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

ssh -f $host "$env $bin/PFSRasCluster $cell $id $ras\_cell RAS/$load/$cell"

done

fi

exit 0

### 遠隔操作コマンドシェル

遠隔操作では、構成ファイル名がグループ名となる。

|  |  |
| --- | --- |
| ras\_deploy.sh　cluster | ホームディレクトリ下の構成ファイル（clusterに従って、ディレクトリの作成、コマンド配布、定義ファイル配布を行う。Localでは、「~/bin」に配布する。 |
| ras\_start/restart.sh　cluster　[-i id] [-r ras\_cell]  ras\_eye\_data.sh、ras\_mail\_data.shが必要 | クラスターグループ(cluster)でRasEye/RasMailを遠隔起動する。  Idが指定されていなければ全てに適用される。  Ras\_eye\_data.sh、ras\_mail\_data.shで初期値設定を行う。 |
| ras\_stop.sh　cluster　[-i id] | クラスターグループ(cluster)でRasEye/RasMailを遠隔停止する。  Idが指定されていなければ全てに適用される。 |
| ras\_sutonomic.sh　cluster　[-i id] [-r ras\_cell] | RasEyeからの自動起動プロセス |
| ras\_adm.sh set cluster ras\_cell path mailer “autonomic”  ras\_adm.sh unset cluster cell path  pathは、「サービス/セルあるいはクラスター」、RASプリフィックスが付与される。  Mailerは、例えば、/home/nw/bin/RasMail。  Autonomicは、例えば  ”/home/nw/bin/cell\_autonomic.sh　PFSServer”。 | クラスターグループ(cluster)の全て にRASセル(cell)のイベントディレクトリ(directory)監視の設定解除を行う。  また、イベント発生時のメール通知、障害時の再起動を行う。 |
| ras\_adm.sh　{add|del} cluster　mailaddr | クラスターグループ(cluster)の全て に全てのRasMail に通知先メールアドレス(mailaddr)の追加削除を行う。 |
| ras\_adm.sh {regist\_me|unregist\_me}　cluster　cell | クラスターグループ(cluster)の全て にRASセル(cell)での自己監視の登録削除を行う。 |
| Ras\_adm.sh xxx\_dump cluster id  (xxx:ras、mail、regist、session) | クラスターグループ(cluster)のidの情報を取得する。  対象は、RAS、メールアドレス、自己監視、セッション情報である。 |
| Ras\_eye/mail\_adm.sh log group id | グループ(group)のidのログ情報を取得する。 |

例）

Ras\_eye\_data.sh

#!/bin/sh

#

# Analyze parameters

#

args=$#

if [ $args -lt 4 ]

then

echo "USAGE:ras\_eye\_data.sh cluster {set|unset|reg|unreg} -r ras\_cell"

exit 1

fi

cluster=$1

shift

cmd=$1

shift

while [ "$1" != "" ]

do

case "$1" in

"-r") shift; ras\_cell=$1; shift ;;

"-i") shift; ID=$1; shift ;;

\* ) shift;;

esac

done

conf=~/bin/$cluster.conf

#

# Active check

#

while read id host bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ "X$ID" != "X" -a "$ID" != "$id" ]; then continue; fi

eval bin=$bin

env="PAXOS\_CELL\_CONFIG=$bin/"

ssh -n $host "$bin/RasEyeAdm $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -eq 0 ]

then

EyeIDs=$EyeIDs"$id "

eval Eyebin\_$id=\$bin

eval Eyeroot\_$id=\$root

eval Eyehost\_$id=\$host

fi

done < $conf

#echo EyeIDs[$EyeIDs]

for id in $EyeIDs

do

eval host='$Eyehost\_'$id

eval bin='$Eyebin\_'$id

eval root='$Eyeroot\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

case $cmd in

"set")

#self set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id set $ras\_cell RasEye/$cluster $bin/RasMail $bin/ras\_autonomic.sh"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id set $ras\_cell RasEye/$cluster $bin/RasMail $bin/ras\_autonomic.sh"

#css set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id set $ras\_cell PFSServer/css $bin/RasMail $bin/cell\_autonomic.sh PFSServer"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id set $ras\_cell PFSServer/css $bin/RasMail \"$bin/cell\_autonomic.sh PFSServer\""

#xjPing set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id set $ras\_cell xjPingPaxos/xjPing $bin/RasMail $bin/cell\_autonomic.sh xjPingPaxos"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id set $ras\_cell xjPingPaxos/xjPing $bin/RasMail \"$bin/cell\_autonomic.sh xjPingPaxos\""

#lvs\_0 set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id set $ras\_cell LVServer/lvs\_0 $bin/RasMail $bin/cell\_autonomic.sh LVServer"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id set $ras\_cell LVServer/lvs\_0 $bin/RasMail \"$bin/cell\_autonomic.sh LVServer\""

#tgtdVV set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id set $ras\_cell VV/vv $bin/RasMail $bin/vv\_autonomic.sh vv css xjPing"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id set $ras\_cell VV/vv $bin/RasMail \"$bin/vv\_autonomic.sh vv css xjPing\""

#Memcache set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id set $ras\_cell PaxosMemcache/Memcache $bin/RasMail $bin/mem\_autonomic.sh css"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id set $ras\_cell PaxosMemcache/Memcache $bin/RasMail \"$bin/mem\_autonomic.sh css\""

;;

"unset")

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset $ras\_cell RasEye/$cluster"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset $ras\_cell RasEye/$cluster"

#css set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset $ras\_cell PFSServer/css"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset $ras\_cell PFSServer/css"

#xjPing set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset $ras\_cell xjPingPaxos/xjPing"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset $ras\_cell xjPingPaxos/xjPing"

#lvs\_0 set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset $ras\_cell LVServer/lvs\_0"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset $ras\_cell LVServer/lvs\_0"

#tgtdVV set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset $ras\_cell VV/vv"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset $ras\_cell VV/vv"

#Memcache set

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset $ras\_cell PaxosMemcache/Memcache"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset $ras\_cell PaxosMemcache/Memcache"

;;

"reg")

#self reg

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id regist\_me $ras\_cell"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id regist\_me $ras\_cell"

;;

"unreg")

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unregist\_me $ras\_cell"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unregist\_me $ras\_cell"

;;

esac

done

exit 0

Ras\_mail\_data.sh

#!/bin/sh

#

# Analyze parameters

#

args=$#

if [ $args -lt 2 ]

then

echo "USAGE:ras\_mail\_data.sh cluster {add|del}"

exit 1

fi

cluster=$1

cmd=$2

conf=~/bin/$cluster.conf

#

# Active check

#

while read id host bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ "X$ID" != "X" -a "$ID" != "$id" ]; then continue; fi

eval bin=$bin

ssh -n $host "$bin/RasMail $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -eq 0 ]

then

MailIDs=$MailIDs"$id "

eval Mailbin\_$id=\$bin

eval Mailroot\_$id=\$root

eval Mailhost\_$id=\$host

fi

done < $conf

#

# RasMail

#

for id in $MailIDs

do

eval host='$Mailhost\_'$id

eval bin='$Mailbin\_'$id

eval root='$Mailroot\_'$id

echo "ssh -f $host $bin/RasMail $cluster $id $cmd nw@tritech.co.jp"

ssh -f $host "$bin/RasMail $cluster $id $cmd nw@tritech.co.jp"

done

exit 0

Ras\_start.sh

#!/bin/sh

#

# Analyze parameters

#

args=$#

if [ $args -lt 1 ]

then

echo "USAGE:ras\_start.sh cluster [-i id] [-r ras\_cell]"

exit 1

fi

cluster=$1

shift

while [ "$1" != "" ]

do

case "$1" in

"-r") shift; ras\_cell=$1; shift ;;

"-i") shift; ID=$1; shift ;;

\* ) shift;;

esac

done

env0="LOG\_SIZE=20000"

files=20

conf=~/bin/$cluster.conf

#

# Active check

#

while read id host bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ "X$ID" != "X" -a "$ID" != "$id" ]; then continue; fi

eval bin=$bin

ssh -n $host "$bin/RasMail $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -ne 0 ]

then

MailIDs=$MailIDs"$id "

eval Mailbin\_$id=\$bin

eval Mailroot\_$id=\$root

eval Mailhost\_$id=\$host

fi

ssh -n $host "$bin/RasEyeAdm $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -ne 0 ]

then

EyeIDs=$EyeIDs"$id "

eval Eyebin\_$id=\$bin

eval Eyeroot\_$id=\$root

eval Eyehost\_$id=\$host

fi

done < $conf

#echo cluster[$cluster] MailIDs[$MailIDs] EyeIDs[$EyeIDs]

#

# RasMail

#

for id in $MailIDs

do

eval host='$Mailhost\_'$id

eval bin='$Mailbin\_'$id

eval root='$Mailroot\_'$id

env="$env0 PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host cd $root;$env $bin/RasMail $cluster $id 3>&1 1>&2 2>&3 | $bin/LogFiles RasMail $files"

ssh -f $host "cd $root;$env $bin/RasMail $cluster $id 3>&1 1>&2 2>&3 | $bin/LogFiles RasMail $files"

sleep 1

. ras\_mail.data

done

#

# RasEye

#

for id in $EyeIDs

do

eval host='$Eyehost\_'$id

eval bin='$Eyebin\_'$id

eval root='$Eyeroot\_'$id

env="$env0 PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host cd $root;$env $bin/RasEye $cluster $id 3>&1 1>&2 2>&3 | $bin/LogFiles RasEye $files"

ssh -f $host "cd $root;$env $bin/RasEye $cluster $id 3>&1 1>&2 2>&3 | $bin/LogFiles RasEye $files"

sleep 1

data="start"

. ras\_eye.data

done

exit 0

Ras\_stop.sh

#!/bin/sh

args=$#

if [ $args -lt 1 ]

then

echo "USAGE:ras\_stop.sh cluster [id]"

exit 1

fi

cluster=$1

conf=~/bin/$cluster.conf

#

# Check inactive

#

while read id host bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ $args -ge 2 ]; then if [ $2 -ne $id ]; then continue; fi; fi

eval bin=$bin

env="PAXOS\_CELL\_CONFIG=$bin/"

ssh -n $host "$env $bin/RasMail $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -eq 0 ]

then

MailIDs=$MailIDs"$id "

eval Mailbin\_$id=\$bin

eval Mailroot\_$id=\$root

eval Mailhost\_$id=\$host

fi

ssh -n $host "$env $bin/RasEyeAdm $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -eq 0 ]

then

EyeIDs=$EyeIDs"$id "

eval Eyebin\_$id=\$bin

eval Eyeroot\_$id=\$root

eval Eyehost\_$id=\$host

fi

done < $conf

#

# unset\_all

#

for id in $EyeIDs

do

eval host='$Eyehost\_'$id

eval bin='$Eyebin\_'$id

eval root='$Eyeroot\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset\_all"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset\_all"

done

sleep 1

#

# stop RasMail

#

for id in $MailIDs

do

eval host='$Mailhost\_'$id

eval bin='$Mailbin\_'$id

eval root='$Mailroot\_'$id

echo "ssh -f $host $bin/RasMail $cluster $id stop"

ssh -f $host "$bin/RasMail $cluster $id stop"

done

sleep 1

#

# stop RasEye

#

for id in $EyeIDs

do

eval host='$Eyehost\_'$id

eval bin='$Eyebin\_'$id

eval root='$Eyeroot\_'$id

echo "ssh -f $host $bin/RasEyeAdm $cluster $id stop"

ssh -f $host "$bin/RasEyeAdm $cluster $id stop"

done

exit 0

Ras\_autonomic.sh

#!/bin/sh

args=$#

if [ $args -lt 1 ]

then

echo "USAGE:ras\_stop.sh cluster [id]"

exit 1

fi

cluster=$1

conf=~/bin/$cluster.conf

#

# Check inactive

#

while read id host bin root

do

if [ ! $id ]; then break; fi

if [ `echo $id|cut -c1` = '#' ]; then continue; fi

if [ $args -ge 2 ]; then if [ $2 -ne $id ]; then continue; fi; fi

eval bin=$bin

env="PAXOS\_CELL\_CONFIG=$bin/"

ssh -n $host "$env $bin/RasMail $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -eq 0 ]

then

MailIDs=$MailIDs"$id "

eval Mailbin\_$id=\$bin

eval Mailroot\_$id=\$root

eval Mailhost\_$id=\$host

fi

ssh -n $host "$env $bin/RasEyeAdm $cluster $id active > /dev/null 2>&1"

ret=$?

if [ $ret -eq 0 ]

then

EyeIDs=$EyeIDs"$id "

eval Eyebin\_$id=\$bin

eval Eyeroot\_$id=\$root

eval Eyehost\_$id=\$host

fi

done < $conf

#

# unset\_all

#

for id in $EyeIDs

do

eval host='$Eyehost\_'$id

eval bin='$Eyebin\_'$id

eval root='$Eyeroot\_'$id

env="PAXOS\_CELL\_CONFIG=$bin/"

echo "ssh -f $host $env $bin/RasEyeAdm $cluster $id unset\_all"

ssh -f $host "$env $bin/RasEyeAdm $cluster $id unset\_all"

done

sleep 1

#

# stop RasMail

#

for id in $MailIDs

do

eval host='$Mailhost\_'$id

eval bin='$Mailbin\_'$id

eval root='$Mailroot\_'$id

echo "ssh -f $host $bin/RasMail $cluster $id stop"

ssh -f $host "$bin/RasMail $cluster $id stop"

done

sleep 1

#

# stop RasEye

#

for id in $EyeIDs

do

eval host='$Eyehost\_'$id

eval bin='$Eyebin\_'$id

eval root='$Eyeroot\_'$id

echo "ssh -f $host $bin/RasEyeAdm $cluster $id stop"

ssh -f $host "$bin/RasEyeAdm $cluster $id stop"

done

exit 0