第1版 　 2016年2月18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](#_Toc443572774)

[2. 方式 2](#_Toc443572775)

[2.1. 監視・通知・再起動(RasEye) 2](#_Toc443572776)

[2.2. 多重化多段化自律方式（参考） 3](#_Toc443572777)

[3. 機能 3](#_Toc443572778)

[3.1. パスによる管理 3](#_Toc443572779)

[3.2. イベントパス 4](#_Toc443572780)

[4. APIとコマンド 5](#_Toc443572781)

[4.1. PFSRasRegister/PFSOpen/PFSWrite/PFSClose RAS対象資源登録API 5](#_Toc443572782)

[4.2. RAS登録 5](#_Toc443572783)

[4.2.1. PFSRasCluster 5](#_Toc443572784)

[4.3. RasEye 5](#_Toc443572785)

[4.4. RasEyeAdm 6](#_Toc443572786)

[4.5. RasMail 7](#_Toc443572787)

[5. 運用 7](#_Toc443572788)

[5.1. ntp環境 7](#_Toc443572789)

[5.2. メール環境 7](#_Toc443572790)

[5.3. 遠隔操作 7](#_Toc443572791)

[5.3.1. 構成ファイル 7](#_Toc443572792)

[5.3.2. 遠隔操作コマンドシェル 8](#_Toc443572793)

# 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。

# 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

RasRestart

RasMail

RasRestart

　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。

RasRestartは、対象毎にRasEyeに登録されるプロセス名であり、対象が削除されたときに呼び出される。

また、RasEyeを相互監視対象とすることにより、自律的監視システムが実現できる。

　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、該当イベント通番ファイルを調べ、以下であれば破棄する。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RasRestart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。

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

　星印で重複送信が発生するが、通番を記録しているので認識できる。

# 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。

　ロックも同様である。

## 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。

　一般にサービス配下にそれぞれのファイルを作成する。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することになる。

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用の通番ファイルは、“/RasEye/監視グループ”をサフィックスとする。

さらに、RasEyeは自己監視ができる。

TESTはRasEyeの監視グループである。

例）

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

RAS/MemCache/SPACE/RasEye/TEST 監視グループロック・ファイル

RAS/RasEye/RAS/0,1、0\_BAK,1\_BAK 監視対象

RAS/RasEye/RAS/RasEye/TEST 監視グループロック・ファイル

　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。

# 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する。

## RAS登録

　各APサーバの登録方法は次のようになる。

### PFSRasCluster

　「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/{css|cmdb|lvs}/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を直ちに行う。

## RasEye

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。

## RasEyeAdm

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　restart

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restart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

restart 「restart　–c cluster –r ras\_cell …」で起動される

-c　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

-B　directry 指定ディレクトリ

通常、-c、-r必要ががあれば-BでOK。

## RasMail

　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。

# 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

ホームディレクトリ下に「~/\_RAS.conf」の構成ファイルを配置する。

#　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

### 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。 |
| ras\_start/restart.sh　cluster　[id]  ras\_eye.data、ras\_mail.data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、ras\_mail.dataで初期値設定を行う。 |
| ras\_stop.sh　cluster　[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\_adm.sh {set|unset}　cluster　cell　path | 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。 |
| 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\_eye/mail\_probe.sh xxx\_dump group id  (xxx:ras、mail、regist、session) | グループ(group)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

#!/bin/sh

if [ $data = "start" ]

then

ras\_cell=css

#ras cell directory

#echo "ssh -f $host $bin/RasEyeAdm $cluster $id set $ras\_cell css/css RasMail css\_restart.sh"

#ssh -f $host "$bin/RasEyeAdm $cluster $id set $ras\_cell css/css RasMail css\_restart.sh"

#echo "ssh -f $host $bin/RasEyeAdm $cluster $id set $ras\_cell cmdb/cmdb RasMail cmdb\_restart.sh"

#ssh -f $host "$bin/RasEyeAdm $cluster $id set $ras\_cell cmdb/cmdb RasMail cmdb\_restart.sh"

#echo "ssh -f $host $bin/RasEyeAdm $cluster $id set $ras\_cell MemCache/Memcache RasMail mem\_restart.sh"

#ssh -f $host "$bin/RasEyeAdm $cluster $id set $ras\_cell MemCache/Memcache RasMail mem\_restart.sh"

#self cell

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

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

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

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

elif [ $data = "unset" ]

then

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

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

elif [ $data = "unregist" ]

then

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

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

fi

Ras\_mail.data

#mail addr

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

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

Ras\_restart.sh

#!/bin/sh

#

# Analyze parameters

#

args=$#

while [ "$1" != "" ]

do

case "$1" in

"-c") shift; cluster=$1; shift ;;

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

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

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

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

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

\* ) shift;;

esac

done

if [ ! $cluster ]; then

echo "USAGE:ras\_restart.sh -c cluster/cell -r ras\_cell ..."

echo " -I id"

echo " -s event sequence"

echo " -m event mode"

echo " -e event element"

echo " -B BIN directory"

exit 1

fi

#echo "cluster[$cluster] ras\_cel[$ras\_cell] ID[$ID] seq[$seq] mode[$mode] BIN[$BIN]"

env="LOG\_SIZE=20000 "

files=20

conf=$HOME/\_$cluster.conf

#

# Check active

#

while read id host 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

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

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

if [ ! $BIN ]; then BIN=$bin; fi

. $BIN/ras\_mail.data

done

#

# RasEye

#

for id in $EyeIDs

do

eval host='$Eyehost\_'$id

eval bin='$Eyebin\_'$id

eval root='$Eyeroot\_'$id

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

if [ ! $BIN ]; then BIN=$bin; fi

data="start"

. $BIN/ras\_eye.data

done

exit 0