

Heartbeatで簡単・手 軽にクラスタリング！

2007年10月5日

NTT OSSセンター

岡田 敏



本日の話題

- I. Heartbeatの紹介
- II. Heartbeatに関する情報
- III. Heartbeatを動かそう！(デモ)



お話の前提とおことわり

■ 前提

- OSはRedHat EL4U5を使用
- HeartbeatはVer.2を使用(最新版2.1.2)し、Ver.2モードで動作させる

■ おことわり

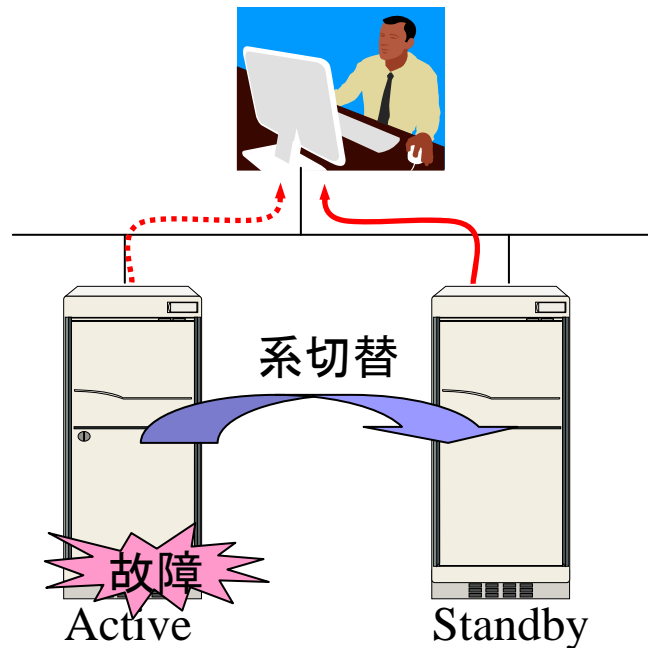
- OS、Heartbeatのバージョンによっては若干、動きがちがうかもしれません。
- 本日、Heartbeatのすべての機能を紹介することはできません
- 詳しくは後ほど紹介するWebページを参照してください




I . Heartbeatの紹介

Heartbeatとは？

- オープンソースのHAクラスタソフトウェア。
 - システムを高可用化し、サービスを連続提供させるためのソフトウェア。



- 複数台のサーバでActive-Standby構成を採る。
- 障害発生時の動き
 1. Activeサーバで障害発生
 2. Standbyサーバへサービス切替
 3. Standbyサーバで継続的にサービス提供



開発コミュニティと歴史

- Linux-HAコミュニティ(<http://www.linux-ha.org/>)で開発

- プロジェクトリーダー: Alan Robertson氏

- <http://www.linux-ha.org/AlanRobertson>

- 歴史

- 1999 年 最初の Heartbeat ソフトウェアをリリース

- 2003 年 Heartbeat バージョン1系リリース

- 2005 年 Heartbeat バージョン2系リリース

- 現時点でバージョン2.1.2



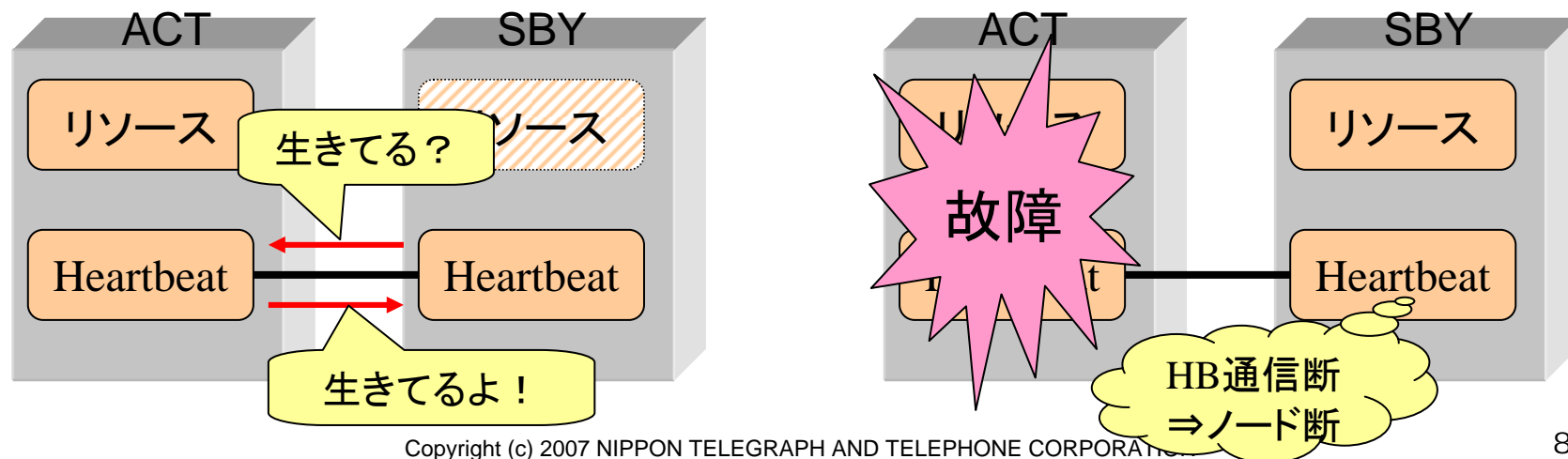
用語

- ノード: サーバのこと
 - Active(ACT): サービスを提供しているノード
 - Standby(SBY) 稼動系に何らかの障害が発生した場合にサービスを引き継ぐノード
- リソース: Heartbeatの制御(起動/停止/監視)対象。
 - 例: PostgreSQL, 仮想アドレス, Apache, など
- リソースエージェント: リソースを制御するためのプログラム(主にシェルスクリプト)
 - PostgreSQL, MySQL, Oracle, DRBD, Apache, 仮想アドレス, ファイルシステム等は提供されている。

基本的動作：ノードの監視

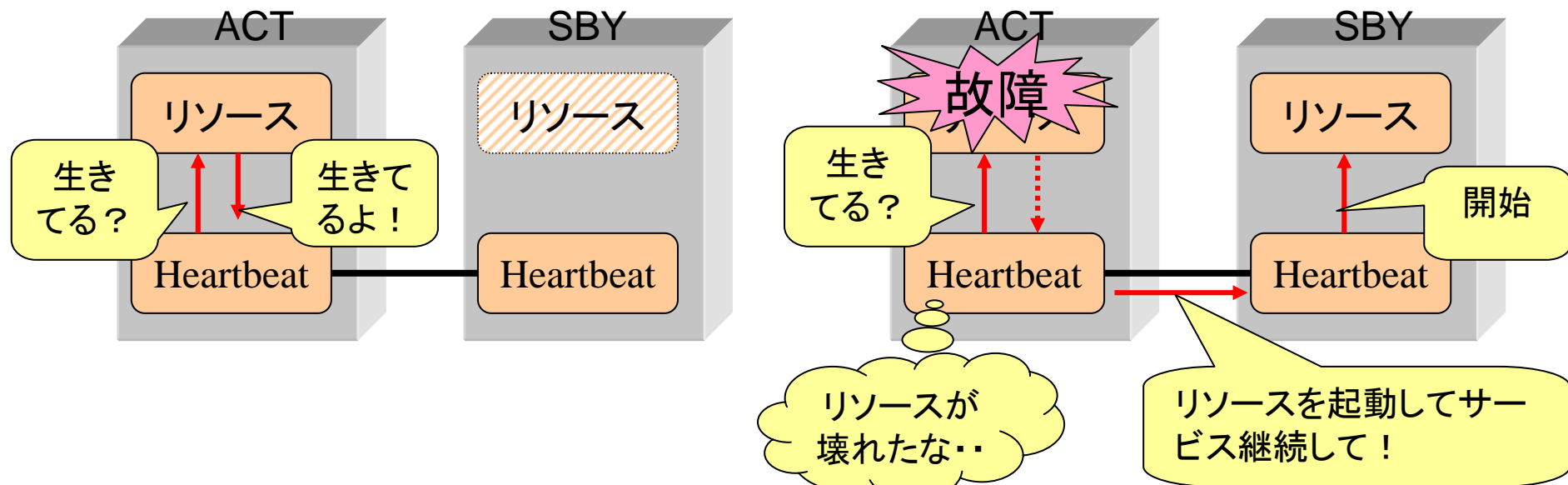
■ 相手ノードの監視

- 一定間隔で相手ノードと通信し、相手ノードの生死を確認する(Heartbeat通信)。
- 相手ノードと通信できなくなった場合に、相手はダウンしたと判断し、フェイルオーバ等の処理を行う。



基本的動作：リソースの制御

- リソースの制御：起動、停止、監視
 - 起動後は一定間隔で監視。正しく動作していない場合にはフェイルオーバ等の処理を実施。





その他の特徴的機能

- Ver.2より柔軟なリソースの制御が可能
 - リソースのグループ化
 - 複数のリソースをグループ化して扱うことが可能
 - 例: PostgreSQLと仮想アドレスをグループ化 ⇒ デモ
 - リソースに対する制約の付与
 - リソースが動くべきノードの指定
 - 複数リソース間で起動・停止順序の指定可 など
 - clone、MasterSlave機能
 - clone 複数ノードで同じ設定のリソースを起動可
 - MasterSlave 3状態を持つリソースを制御可

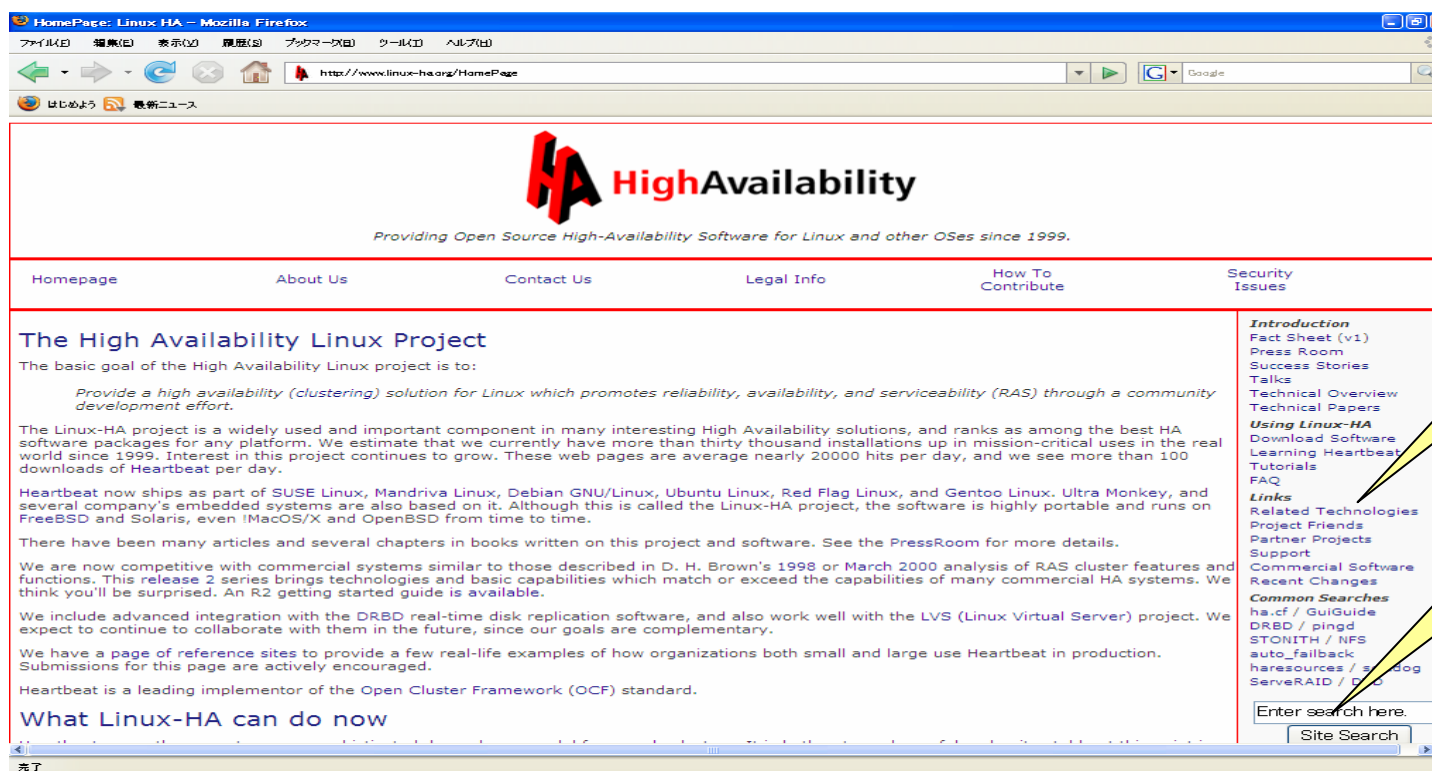


Ⅱ . Heartbeatに関する情報

Webページ(英語)

■ Linux-HA のトップページ

<http://www.linux-ha.org/>



主な
ピックの
メニュー

Web
ページ
内検索



メーリングリスト(英語)

- 本家のメーリングリスト(英語)

<http://www.linux-ha.org/ContactUs>を参照

各メーリングリストのアーカイブ、参加方法のリンクあり

- Main Mailing List**

- HBの使用方法、各種問合せなど

- Development mailing list**

- 機能追加、バグ報告など

- Announcement Mailing List**

- 新規リリースの情報など(一般ユーザは投稿不可)

Webページ(日本語)

- 日本語Webページ 10月5日開設！！
 - 本家ページの主要部分を日本語化
- URL http://linux-ha.org/ja/HomePage_ja

The screenshot shows the Japanese version of the High Availability website. The browser address bar displays http://172.20.26/mozz.php/ta/HomePage_ja. The page features the High Availability logo and navigation links for Home Page, Site Information, Contact Information, Usage Conditions, Collaboration Methods, and Security. The main content area is titled "The High Availability Linux" and includes a section for "Linux-HAの機能" (Features of Linux-HA). A sidebar on the right contains a "ダウンロード" (Download) section with links for Linux-HA releases and a "Webページ内検索" (Search) section with a search box and a "Site Search" button. Three yellow callout boxes are overlaid on the page: "ML情報" (ML Information) points to the "Linux-HAの機能" section, "主なピックアップのメニュー" (Main Pick-up Menu) points to the "ダウンロード" section, and "Webページ内検索" (Web Page Search) points to the search box.



メーリングリスト(日本語)

- 日本語ML Linux-ha-jp
 - 10月5日開設！！
- 登録方法は以下のURLを参照
 - 日本語Webページ → コンタクト情報



Ⅲ. Heartbeatを動かそう！

動作させるまでの手順

0. 構成、動作の検討

- ノード構成、リソースの構成などを検討・設計

1. インストール

2. 起動に必要なファイルの設定

1. ha.cfの設定
2. logd.cfの設定
3. 認証キーファイルauthkeys

3. Heartbeat 起動

4. 設定ファイルcib.xmlの設定

5. チューニング

- 監視間隔などのチューニング。設定ファイルを編集

本日はここを説明



1. Heartbeatのインストール

- インストールは簡単！！
 - 動作環境に応じたファイルをダウンロード
 - 必要なファイル(RedHatの場合)
 - heartbeat-2.1.2-1.i386.rpm
 - heartbeat-gui-2.1.2-1.i386.rpm
 - heartbeat-pils-2.1.2-1.i386.rpm
 - heartbeat-stonith-2.1.2-1.i386.rpm
 - Heartbeat用ユーザ／グループの設定
 - グループ グループ名 : haclient / グループ番号 : 90
 - ユーザ ユーザ名 : hacluster / ユーザ番号 : 90
 - ダウンロードしたファイルをインストール

2.1. ha.cfの設定

ha.cf 主な設定項目と意味

crm	HeartbeatのVer.2モードを使うかどうかの指定
use_logd	logデーモンを使ってログを出力するかどうかの指定
keepalive	クラスタ内のノードが活着しているかどうか確認するためのHeartbeat通信の間隔。
deadtime	Heartbeat通信が途絶えて、ノードが死んだと判断する時間
initdead	nodeで指定された全ノードの起動を待つ時間
udpport	Heartbeat通信に使用するポート番号を指定
bcast	Heartbeat通信の通信方式をbroadcast通信と指定し、使用するネットワークインタフェース (eth??など)を指定。 複数ネットワークインタフェースを使う場合は併記。 ucastでunicast, mcastでmulticastの使用を指定可。
node	クラスタを構成するノード名 (uname -nの値)を指定。

2.2. logd.cfの設定

- Heartbeatのログ出力先を指定
- クラスタ内の全ノードに、同じ内容のファイルを配置
 - ファイル名 `logd.cf`
 - 配置場所 `/etc/logd.cf`

設定例

```
logfile /var/log/ha-log  
debugfile /var/log/ha-debug
```

意味

logfile	ログの出力先ファイルを指定
debugfile	デバッグ情報の出力先ファイルを指定

2.3. 認証キーファイルの設定

- クラスタを構成する認証キーを保持するファイル
- 認証キーが同じノード群でクラスタを構成
- クラスタ内の全ノードに、同じ内容のファイルを配置
 - ファイル名 authkeys
 - 配置場所 /etc/ha.d/authkeys
 - 権限・ユーザ/グループ 600 root/root

設定例

```
auth 1  
1 sha1 RecommendationModelCluster01
```

認証キー: 任意の文字列

認証キーの計算方法: sha1, md5, crcを指定可

3. Heartbeatの起動

- 一度Heartbeatを起動

起動方法

```
# service heartbeat start  
Starting High-Availability service: [OK]
```

- OK表示が出ても、起動成功とは限りません
- しばらくたった後(3分程度)、正常に起動されているか確認
- 正常起動されていなければ、ログなどを参照して設定ファイルの内容を修正

停止方法(参考)

```
# service heartbeat stop  
Stopping High-Availability service: [OK]
```

3. Heartbeatの起動

稼動状況の確認

- Heartbeatのコマンド/usr/sbin/crm_monを利用して稼動状況を確認

```
Refresh in 13s...
```

```
=====
```

```
Last updated: Fri Sep 28 14:26:03 2007
```

```
Current DC: node02 (1e619500-6293-42b1-978e-9f6a58cd2c99)
```

```
2 Nodes configured. 0 Resources configured.
```

```
=====
```

```
Node: node02 (1e619500-6293-42b1-978e-9f6a58cd2c99) : online
```

```
Node: node01 (c7f2dfd6-63f9-4783-8005-efaba370cf43) : online
```


4. cib.xmlの設定

- cib.xml = 主に、リソースの定義を設定するxmlファイル
 - どのようなリソースをどのように扱うか
 - 起動、監視、停止時に関連する時間
 - リソースの配置などを指定
- 3.Heartbeatの起動で、雛形ができていますので、編集することにより、リソースの定義を行う。
- 編集方法：以下の2つの方法がある
 - cib.xmlを直接編集
 - 細かい設定が可能
 - cib.xmlの記法(DTD)を知る必要があり、難しい。
 - GUIを使って編集
 - 簡単
 - 設定できる内容に限界がある。

本日はここを説明



4. cib.xmlの設定

GUIによるリソースの追加

1. heartbeat GUIを起動

```
# /usr/lib/heartbeat-gui/haclient.py
```

2. ログイン画面よりログイン(ユーザ:hacluster / パスワード:haclusterのパスワード)

3. リソースの追加

4. 制約の追加

- 詳しくはデモで



Heartbeatの操作コマンド

- Heartbeatで用意されている主なコマンド(CUI)

crm_mon	クラスタのノード、リソースの状態を表示する。表示方法にはテキスト形式とHTML形式が存在する
crm_standby	ノードのステータスを表示／制御する
crm_verify	cib.xmlの正当性をチェックする。
crm_resource	リソースを動的に操作(表示／設定／削除)する。
cibadmin	クラスタ状態をXML形式で表示／変更する。



ログ情報

■ Heartbeatの動作は/var/log/ha-logに出力

出力例

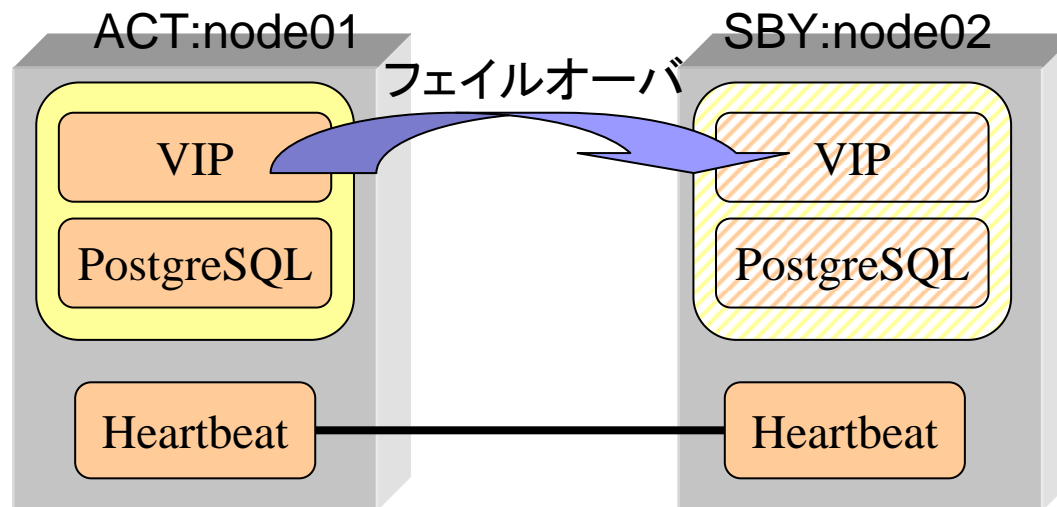
```
lrm[5109]: 2007/10/01_18:22:13 info: RA output:  
(ip01:start:stderr) Rewrote octal netmask as: 24  
IPaddr[5266][5343]: 2007/10/01_18:22:13 INFO: eval ifconfig  
eth0:0 192.168.201.10 netmask 255.255.255.0 broadcast  
192.168.201.255  
crmd[5112]: 2007/10/01_18:22:13 info: process_lrm_event: LRM  
operation ip01_start_0 (call=7, rc=0) complete
```

ログフォーマット

```
ログ出力プロセス名[プロセスID]:出力時間 ログレベル: ログ情報
```

デモ


- サーバ2台構成、ノード名 : node01, node02
- リソース : 仮想アドレス(VIP)とPostgreSQLをグループ化リソースとし、GUIより設定
- フェイルオーバーの実験





まとめ

- Heartbeatは手軽！
 - ダウンロード&インストールで使える状態に
 - 主なリソースのリソースエージェントは付いてくる。
- Heartbeatは簡単！？
 - 簡単な構成ならGUIで設定可能。
 - 複雑な構成をとる場合は面倒なこともあるが...
- 一度、試してみましよう！！



まとめ(続き)

- 日本語Webページ開設 ⇒ 日本語の情報も入手可能に！
- 日本語ML開設 ⇒ 日本語で問合せ等も可能に！
- コミュニティを活性化して、さらに使いやすいソフトウェアにしていきましょう！！