

進化を続けるPacemaker

Linux-HA Japan プロジェクトの これまでとこれから



2020年02月21日
Linux-HA Japan プロジェクト
<http://linux-ha.osdn.jp/>
森 啓介

自己紹介

■ 名前: 森 啓介 (Keisuke MORI)

- twitter: @ksk_ha

■ Linux-HA Japanプロジェクト関連の活動

- Pacemakerリポジトリパッケージのリリース
- <http://linux-ha.osdn.jp/>

■ ClusterLabs プロジェクトのコミッタ

- Pacemaker、resource-agents などHAクラスタ関連の開発コミュニティ
- <https://github.com/ClusterLabs/>

■ 本業

- 普段の業務: NTT OSSセンタ

- NTTグループ内におけるPacemaker/Heartbeatの導入支援・サポート
- バグ報告・パッチ作成などによるNTTから開発コミュニティへのフィードバック・貢献

本日のトピックを一言で言うと…



Linux-HA Japan プロジェクトの
活動方針が少し変わります！

- Pacemakerとは
- Linux-HA Japan プロジェクトのこれまでと今後
- 使い方はどう変わる?
- Pacemaker/HAクラスタの未来

Pacemaker? なにそれおいしいの?



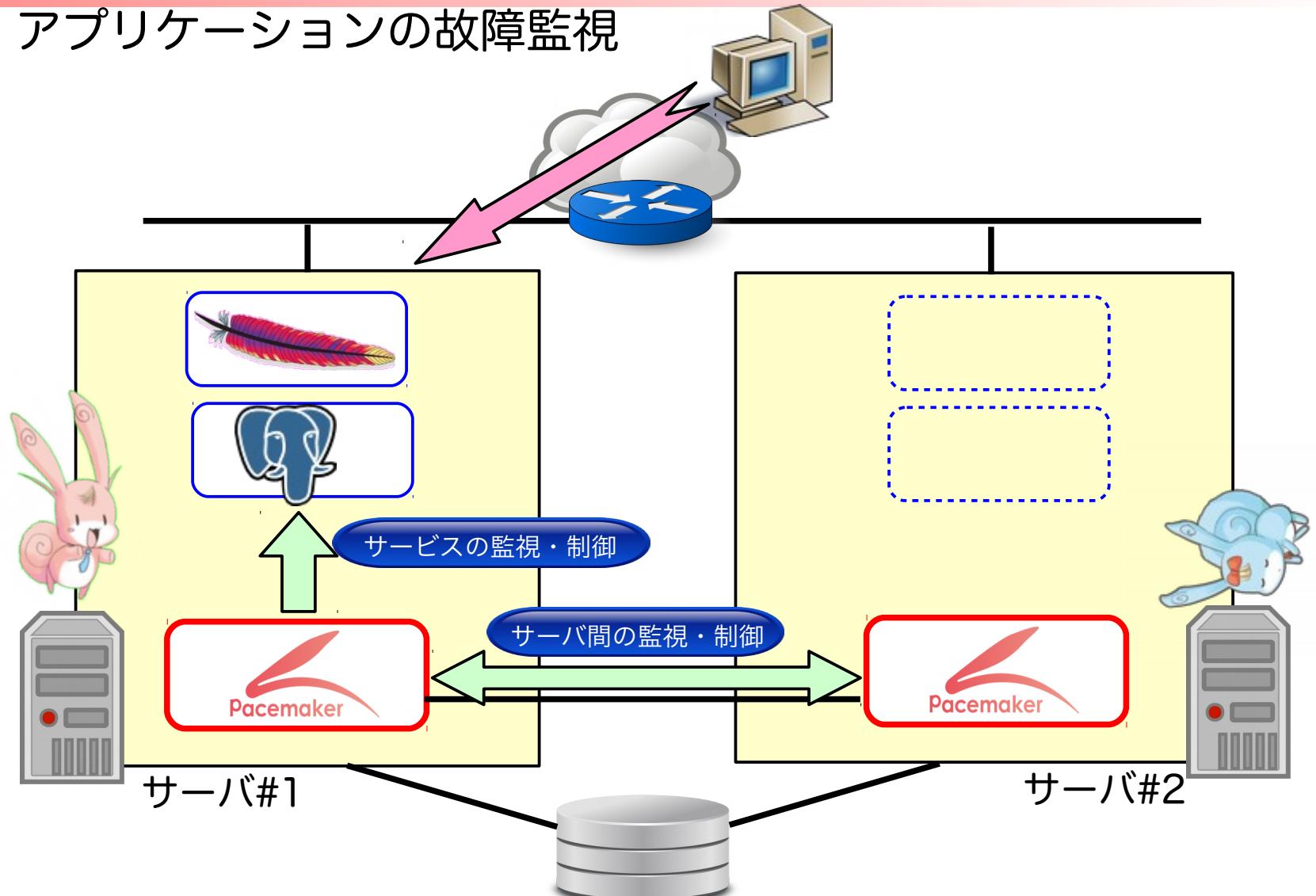
- Pacemakerとは、オープンソースのHAクラスタソフトウェアです。

High **A**vailability = 高可用性



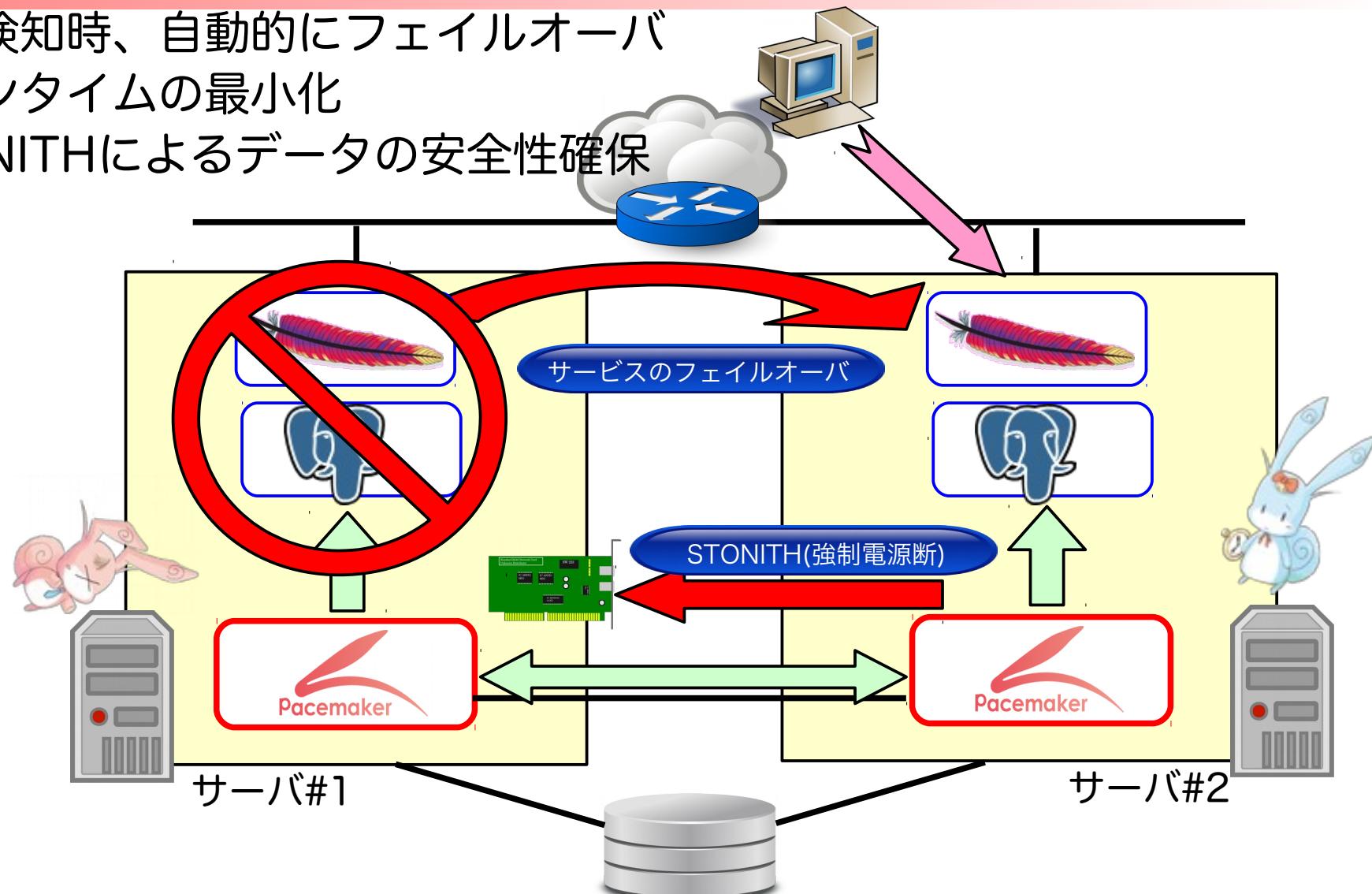
Pacemakerの概要

■ サーバ・アプリケーションの故障監視



Pacemakerの概要

- 故障検知時、自動的にフェイルオーバ
- ダウンタイムの最小化
- STONITHによるデータの安全性確保



Pacemakerを詳しく知りたかったら…



Screenshot of the Linux-HA Japan website (linux-ha.sourceforge.jp/wp/). The page shows a video player for a Pacemaker CM 35sec English Ver.1 video on YouTube. The video thumbnail features the 'pacemaker SOURCEWARE' logo. The website navigation includes HOME, メーリングリスト, ダウンロード&インストール, マニュアル, デスクトップテーマ・壁紙等, コミュニティ概要, 関連コミュニティ, その他, ニュース, イベント情報, 試み物, WEBラジオ.

Pacemaker 応援キャラクター

高良かな

べーちゃん

高良かよ

ころちゃん

ピアンカ



Linux-HA Japan プロジェクト OF ○○○教室にてデモ展示中！

*OSC2020 Tokyo/Springは展示中止となりました

もくじ

■ Pacemakerとは

■ Linux-HA Japan プロジェクトのこれまでと今後

- 過去: プロジェクトの発端と目的
- 現在: 取り巻く状況の変化
- 未来: 今後の方針

■ 使い方はどう変わる?

■ Pacemaker/HAクラスタの未来

過去: Linux-HA Japanプロジェクトの発端



■ そもそもいつ、なぜ、何の目的で活動を始めたか

■ 設立日:

□ 2007年10月4日

Enterprise Watch 最新ニュース

冗長構成を支えるOSSミドルウェア「Heartbeat」の日本語Webサイトがオープン

日本電信電話株式会社（以下、NTT）、エヌ・ティ・ティ・データ先端技術株式会社（以下、NTT-DI）、VA Linux Systems Japan株式会社（以下、VA Linux）、日本電気株式会社（以下、NEC）の4社は、企業システム向けOSSミドルウェアの普及活動の一環として、Linux上で動作し、サービスの停止時間を短縮化する「Heartbeat」の日本語Webサイトをオープンすると発表した。10月5日から公開する。

Heartbeat動作イメージ

Heartbeatは、プライマリシステムの万一の障害に備えた冗長構成環境において、システムの状態を監視し、障害を検出した場合に、予備のシステムへの切り替えを行 OSSミドルウェア。

今回オープンするWebサイトでは、コミュニティと協力し、プログラムやインストールマニュアル、設定例などをダウンロード提供するほか、メーリングリストを併設。これに参加することで利用の際の疑問やバグ情報などをユーザー間で共有することが可能だ。また、今後のリリーススケジュールや最新のパッチ情報なども掲載するとのこと。

■ URL

(*)

(*)1 <https://enterprise.watch.impress.co.jp/cda/topic/2007/10/04/11299.html>

Copyright (c) 2007 Impress Watch Corporation, an Impress Group company. All rights reserved.

(*)2 <https://www.ntt.co.jp/news/news07/0710/071004a.html>

Copyright(c) 2007 日本電信電話株式会社

News Release 071004a
2007年10月4日 (報道発表資料)

日本電信電話株式会社
エヌ・ティ・ティ・データ先端技術株式会社
VA Linux Systems Japan株式会社
日本電気株式会社

オープンソース・ソフトウェア（OSS）の普及展開に向けた取り組みについて
～企業システム向けOSSミドルウェアの日本語サイト公開～

日本電信電話株式会社（以下、NTT、東京都千代田区、代表取締役社長：三浦惺）と、エヌ・ティ・ティ・データ先端技術株式会社（以下、NTT-DI、東京都江東区、代表取締役社長：山田伸一）と、VA Linux Systems Japan株式会社（以下、VA Linux、東京都中央区、代表取締役社長：上田哲也）と、日本電気株式会社（以下、NEC、東京都港区、代表取締役執行役員社長：矢野薫）は、從来より、Linux(R) OSやデータベース管理システム PostgreSQL^{(*)1}をはじめとする企業システム向けOSSミドルウェア^{(*)2}の普及支援をおこなってまいりましたが、日本における更なる普及展開を目的として、Linux上で動作し、サービスの停止時間を短縮化することを可能とする『Heartbeat（ハートビート）』の日本語サイトを平成19年10月5日（金）にオープンします。

■背景

(*)2

Linux-HA Japan プロジェクト設立の目的



■ Linux-HA 日本語コミュニティの開設

- 利用ノウハウの情報交換(Webサイト・メーリングリスト)
- 日本国でのHAクラスタソフトウェアの知名度向上
- オープンソースでも商用レベルのサービスができる!ことのアピール

■ すぐ使えるバイナリパッケージのリリース

- 開発コミュニティからはソースコードリリースのみ(当時)
- 必要なコンポーネントを集めてビルド・バイナリパッケージを提供
- さらに使いやすくするための運用補助ツールを提供



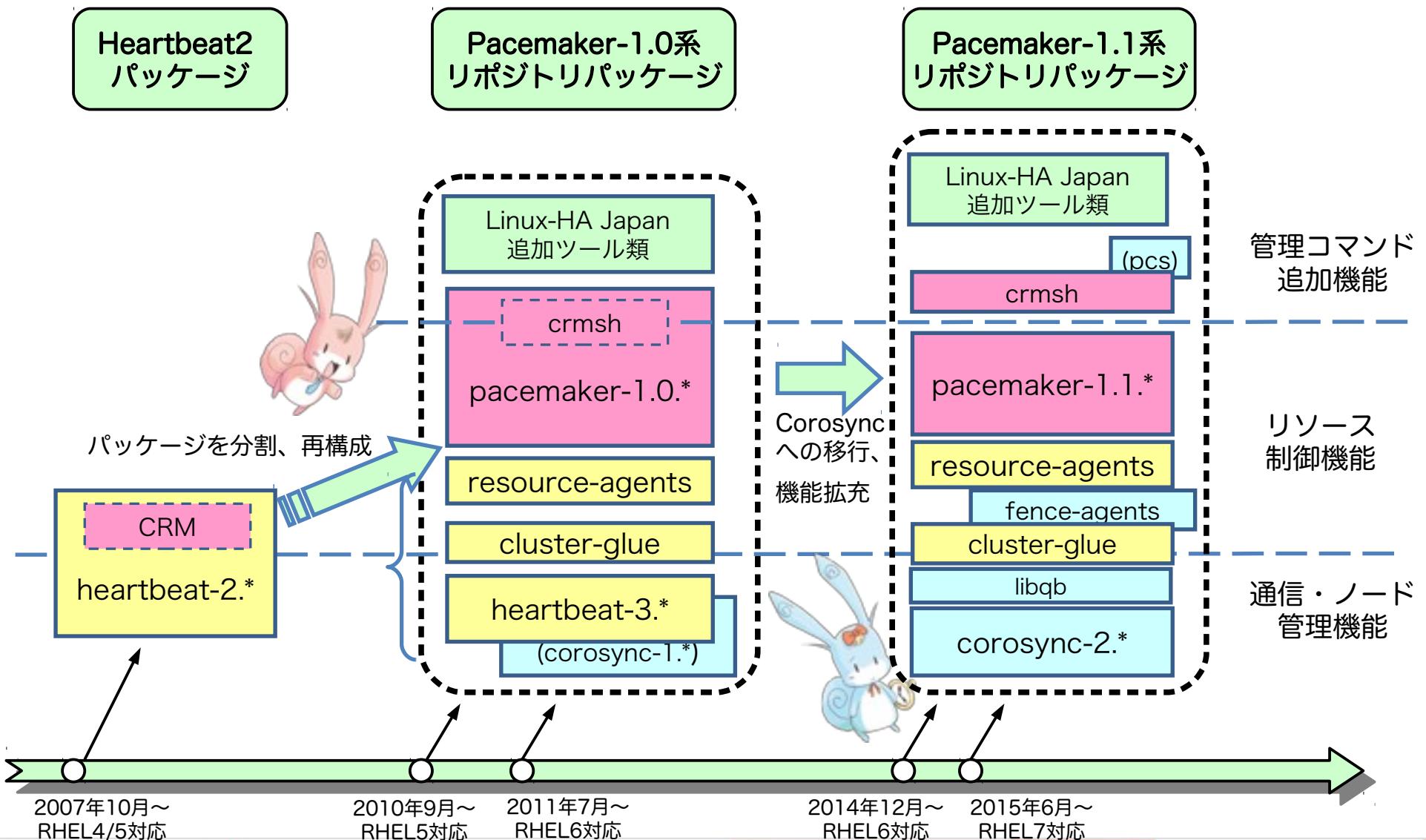
HighAvailability

Heartbeat (当時)

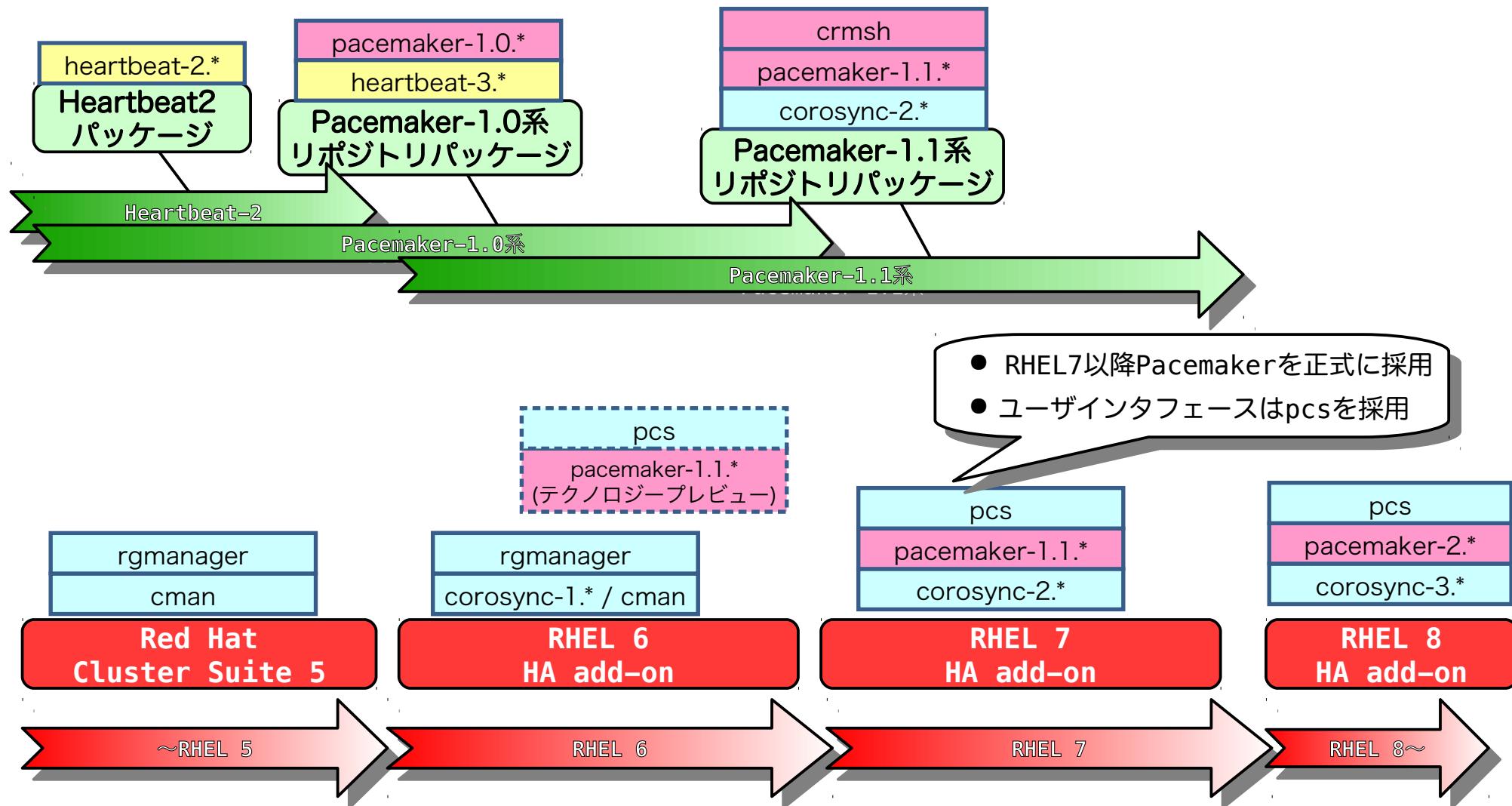


Pacemaker (現在)

Heartbeat / Pacemakerの発展



RHEL HAクラスタ製品との関連



現在: 取り巻く状況の変化

- RHEL7以降 HA add-on にPacemakerを正式採用済み
 - CentOS 7 を含めすでに広く利用されはじめている
- 他の一般的なディストリビューションでも採用済み
 - SLES, Ubuntu
- Linux-HA Japan 当初の目的の一つ
「すぐ使えるバイナリパッケージの提供」はその役目を終えつつある

★変わります！

■ RHEL 8 / CentOS 8 以降は…

- HA add-on 版 Pacemaker パッケージの利用を推奨していきます。
 - 利用手順・設定も Red Hat社のドキュメント・ナレッジに極力沿った利用方法を推奨します。
- Linux-HA Japan からは、追加ツール類のみリリースを行います。
 - ただし内容は RHEL 8 HA add-on の利用条件の範囲に限定し見直します。

★変わりません！

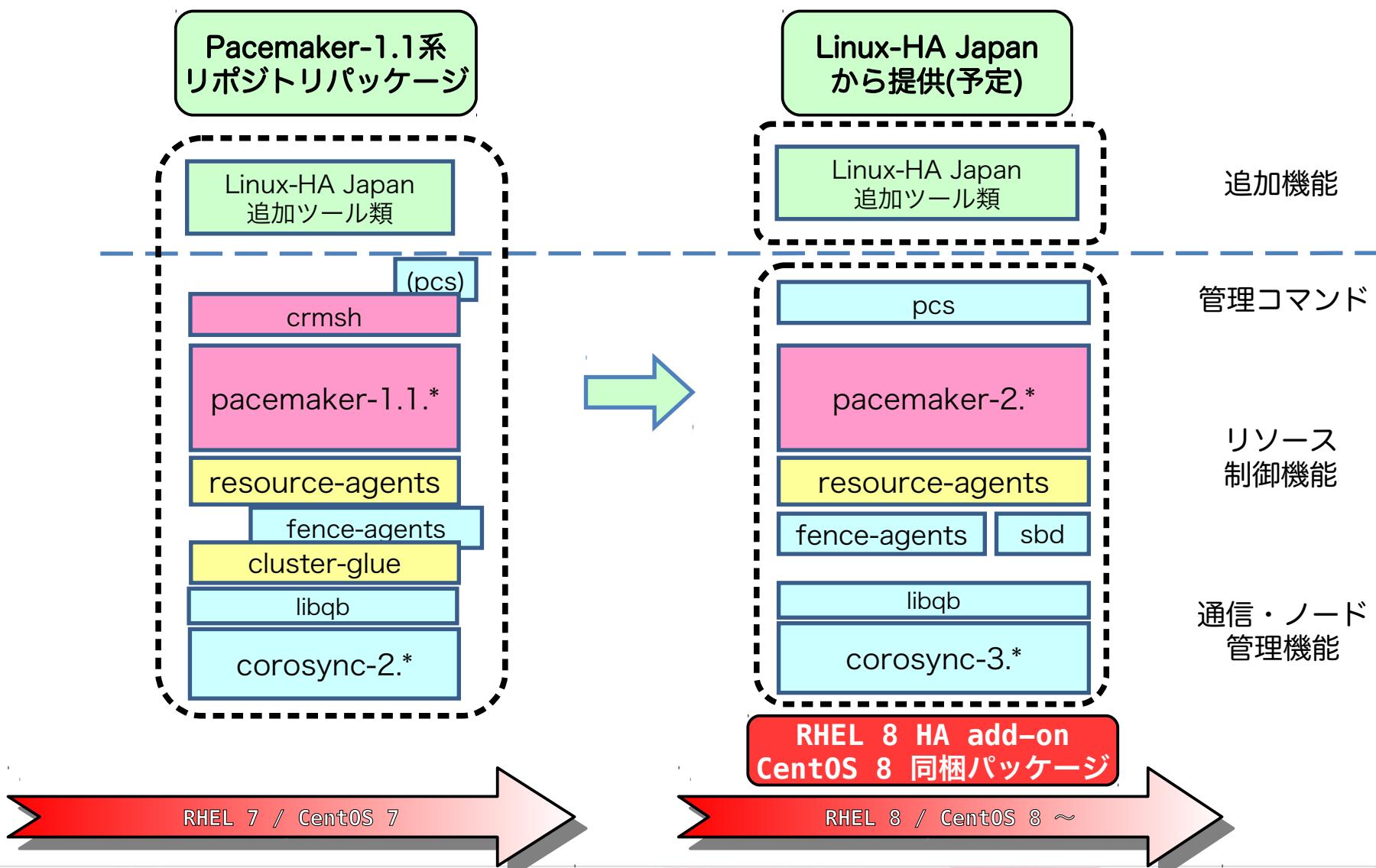
■ 以下は今後も変わりません！

- RHEL 7 / CentOS 7 向けのPacemaker-1.1リポジトリパッケージは、今後も継続してリリースしていきます。
- もちろん、HAクラスタに関する情報交換の場として、Linux-HA Japan コミュニティは引き続き活動を継続していきます！

もくじ

- Pacemakerとは
- Linux-HA Japan プロジェクトのこれまでと今後
- 使い方はどう変わる?
 - コンポーネントの差分
 - pcsによる構築手順例
 - STONITH機能の差分
 - Linux-HA Japan 追加パッケージ内容(予定)
- Pacemaker/HAクラスタの未来

今後の Pacemaker の利用イメージ



コンポーネントの差分

機能	コンポーネント	Pacemaker-1.1系 リポジトリパッケージ	RHEL8.1 HA add-on	主な差異
ユーザインターフェース	crmsh	2.1.9	–	ユーザインターフェースのコンポーネントが大きく異なり、構築時の手順、前提となる設定が大きく異なる。
	pcs	(0.9.167)	0.10.2-4	
リソース管理機能	pacemaker	1.1.21-1	2.0.2-3	前提となる設定が大きく異なるため、故障時の詳細動作にさまざまな差異がある。
通信・ノード管理機能	corosync	2.4.4-2	3.0.2-3	片系切断検知、多重化方式が改善
	kronosnet	–	1.10-1	
	libqb	1.0.5-1	1.0.3-10	
リソースエージェント	resource-agents	4.3.0-1	4.1.1-33	同梱されるリソースエージェントに差異あり
STONITHエージェント	cluster-glue	1.0.12-5	–	STONITH有効化が必須 代替機能への移行が必要
	fence-agents	4.4.0	4.2.1-30	
	sbd	–	1.4.0-15	

crmsh と pcs の違い

作業	Pacemaker-1.1系 リポジトリパッケージ	RHEL 8 HA add-on
ユーザインターフェース	crmsh	pcs
インストール・初期設定	<ul style="list-style-type: none"> 両ノードでパッケージをインストール 両ノードで各種設定ファイルを編集 (corosync.conf等) 	<ul style="list-style-type: none"> 両ノードでパッケージをインストール・初期設定 一方のノードでクラスタ初期設定コマンドを実行 <p> デフォルトで利用する場合設定ファイルの編集は不要</p>
クラスタ起動・終了	<ul style="list-style-type: none"> 両ノードで systemctl コマンドを実行しサービス起動 	<ul style="list-style-type: none"> 一方のノードで pcs コマンドを実行しサービス起動 <p> 一つのpcsコマンドでクラスタ全体の起動・停止が可能</p>
リソース設定	<ul style="list-style-type: none"> crmsh形式の設定ファイルを作成 設定ファイルには全てのリソース設定を記載 crm コマンドでクラスタ全体の設定を反映 	<ul style="list-style-type: none"> 一つの設定項目ごとに一つのpcsコマンドを実行してリソース設定を追加 クラスタ全体の設定は複数のpcsコマンドを実行するシェルスクリプトとして作成 <p> 設定途中は一時的に意図しない起動状態となる場合あり <u>クラスタ全体の一括設定には一時ファイルを作成する手順が必要</u></p> <p> 構築済みのクラスタ設定からpcsコマンドを逆生成することは現状不可</p>

pcsによるインストール・初期設定

(全てのノードで実行)

■インストール

```
# dnf install pcs pacemaker fence-agents-all
```

■クラスタが利用するポートの許可

```
# firewall-cmd --permanent --add-service=high-availability  
# firewall-cmd --add-service=high-availability
```

■クラスタユーザの認証設定

```
# passwd hacluster  
新しいパスワード:
```

■pcsdサービスの有効化

```
# systemctl start pc sd.service  
# systemctl enable pc sd.service
```

(いずれか一つのノードで実行)

■ クラスタの認証設定

```
# pcs cluster auth centos8-1 addr=192.168.0.1 centos8-2 addr=192.168.0.2  
Username: hacluster  
Password:
```

■クラスタの初期設定および起動

```
# pcs cluster setup my_cluster centos8-1 addr=192.168.101.1 addr=192.168.102.1 \  
centos8-2 addr=192.168.101.2 addr=192.168.102.2
```

pcsによるクラスタ起動・停止

(いずれか一つのノードで実行)

- クラスタ全体の起動

```
# pcs cluster start --all
```

(いずれか一つのノードで実行)

- クラスタ全体の停止

```
# pcs cluster stop --all
```

pcsによるリソース設定 (1/2)

■ 構成例: PostgreSQL共有ディスク構成

■ Filesystemリソース設定

```
# pcs resource create filesystem1 ocf:heartbeat:Filesystem \
  device="/dev/mapper/mpatha2" directory="/pgdata" fstype="xfs" force_unmount="safe" \
  op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart \
  stop timeout=60s on-fail=fence
```

■ 仮想IPアドレスリソース設定

```
# pcs resource create ipaddr ocf:heartbeat:IPAddr2 \
  ip="192.168.201.100" nic="eno2" cidr_netmask="24" \
  op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart \
  stop timeout=60s on-fail=fence
```

■ PostgreSQL リソース設定

```
# pcs resource create pgsql ocf:heartbeat:pgsql \
  pgctl="/usr/pgsql-11/bin/pg_ctl" psql="/usr/pgsql-11/bin/psql" \
  pgdata="/pgdata/data" pgdba="postgres" pgport="5432" pgdb="template1" \
  op start timeout=300s on-fail=restart monitor timeout=60s interval=10s on-fail=restart \
  stop timeout=300s on-fail=fence
```

```
# pcs resource group add pgsql-group filesystem1 ipaddr pgsql
```

■ ネットワーク監視(ping)リソース設定

```
# pcs resource create ping ocf:pacemaker:ping \
  name="ping-status" host_list="192.168.201.254" attempts="2" timeout="2" debug="true" \
  op start timeout=60s on-fail=restart monitor timeout=60s interval=10s on-fail=restart \
  stop timeout=60s on-fail=ignore
```

```
# pcs resource clone ping
```

pcsによるリソース設定 (2/2)

■ 構成例: PostgreSQL共有ディスク構成

■ STONITHリソース設定

```
# pcs stonith create fence1-ipmilan fence_ipmilan \
  delay="5" pcmk_host_list="centos8-1" \
  ip="192.168.0.91" username="ipmiuser" password="xxxxxxxx" lanplus="1" \
  op start timeout=60s on-fail=restart monitor timeout=60s interval=3600s on-fail=restart \
  stop timeout=60s on-fail=ignore

# pcs stonith create fence2-ipmilan fence_ipmilan \
  delay="0" pcmk_host_list="centos8-2" \
  ip="192.168.0.92" username="ipmiuser" password="xxxxxxxx" lanplus="1" \
  op start timeout=60s on-fail=restart monitor timeout=60s interval=3600s on-fail=restart \
  stop timeout=60s on-fail=ignore
```

■ リソース動作制御

```
# pcs resource defaults resource-stickiness=200
# pcs resource defaults migration-threshold=1
```

■ リソース配置・順序制約

```
# pcs constraint location pgsql-group prefers centos8-1=200
# pcs constraint location pgsql-group prefers centos8-2=100

# pcs constraint colocation add pgsql-group with ping-clone score=INFINITY
# pcs constraint location pgsql-group rule score=-INFINITY \
  ping-status lt 1 or not_defined ping-status
# pcs constraint order ping-clone then pgsql-group symmetrical=false kind=optional

# pcs constraint location fence1-ipmilan avoids centos8-1
# pcs constraint location fence2-ipmilan avoids centos8-2
```

クラスタ起動後の状態

```
# pcs status --full
Cluster name: my_cluster
Stack: corosync
Current DC: centos8-1 (1) (version 2.0.1-4.el8-0eb7991564) - partition with quorum
Last updated: Tue Sep  3 16:56:42 2019
Last change: Tue Sep  3 16:55:45 2019 by hacluster via crmd on centos8-1

2 nodes configured
7 resources configured

Online: [ centos8-1 (1) centos8-2 (2) ]

Full list of resources:

Resource Group: pgsql-group
    filesystem1      (ocf::heartbeat:Filesystem):     Started centos8-1
    ipaddr          (ocf::heartbeat:IPAddr2):        Started centos8-1
    pgsql           (ocf::heartbeat:pgsql): Started centos8-1
Clone Set: ping-clone [ping]
    ping            (ocf::pacemaker:ping):  Started centos8-2
    ping            (ocf::pacemaker:ping):  Started centos8-1
    Started: [ centos8-1 centos8-2 ]
fence1-ipmilan (stonith:fence_ipmilan):           Started centos8-2
fence2-ipmilan (stonith:fence_ipmilan):           Started centos8-1
```



Node Attributes:

- * Node centos8-1 (1):
 - + ping-status : 1
- * Node centos8-2 (2):
 - + ping-status : 1

Migration Summary:

- * Node centos8-2 (2):
- * Node centos8-1 (1):

Fencing History:

PCSD Status:

- centos8-1: Online
- centos8-2: Online

Daemon Status:

- corosync: active/disabled
- pacemaker: active/disabled
- pcsd: active/enabled

STONITH機能(フェンシング)

- RHEL 8 HA add-on ではSTONITH機能の利用は必須です
 - stonith-enabled=true (デフォルト設定)
- STONITH機能の目的
 - スプリットブレイン対策(排他制御)
 - 制御不能な故障ノードの強制停止



STONITH機能(フェンシング)とは?

- フェンスを立てて隔離すること



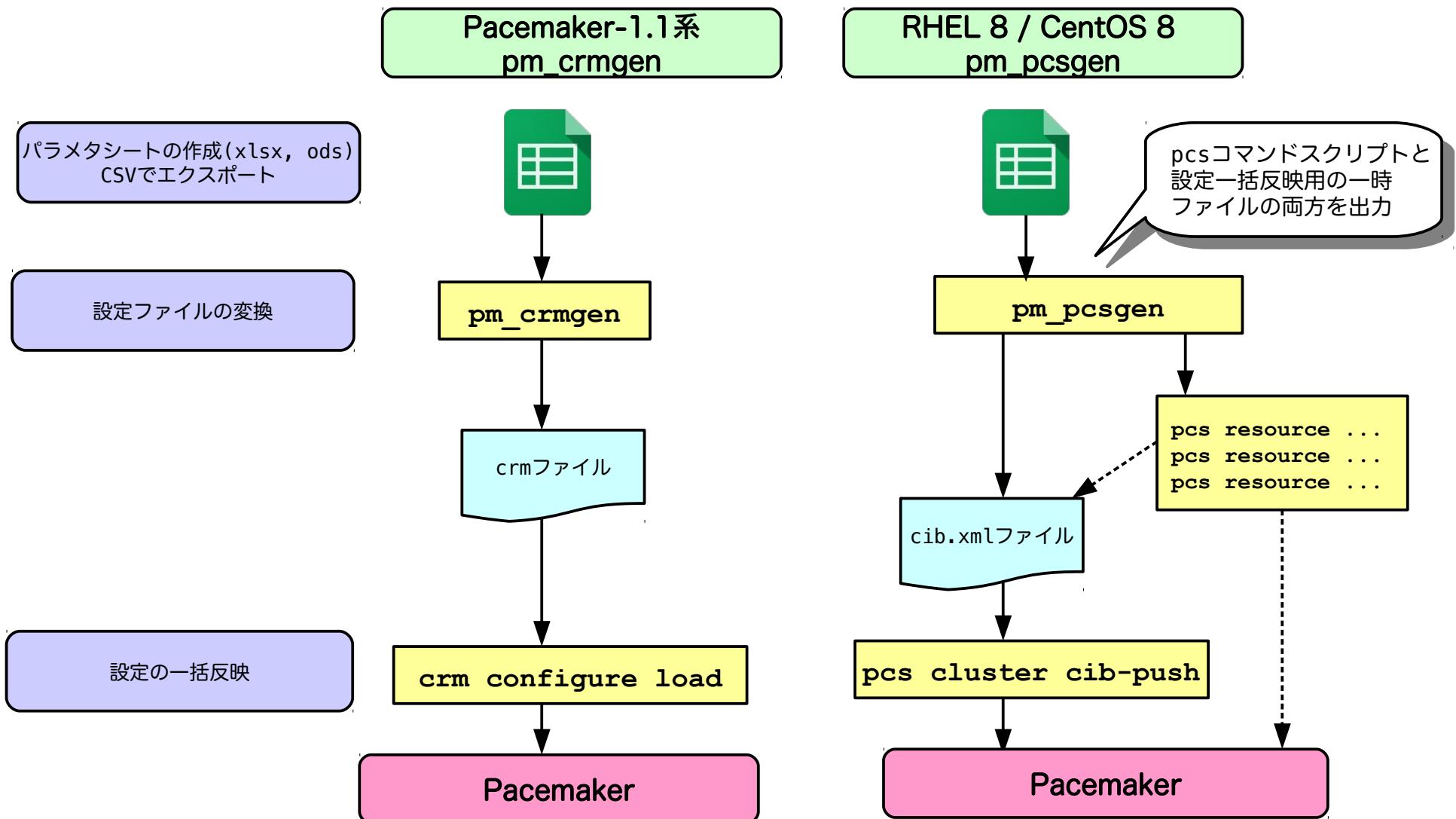
スプリットブレイン対策機能の代替

制御機能	Pacemaker-1.1系 リポジトリパッケージ	RHEL 8 HA add-on	補足
電源制御による 排他制御	external/ipmi	fence_ipmilan	物理環境用。IPMIによる強制電源断 OS側の追加設定(ACPI Soft-Off の無効化)が必要
	external/libvirtd	fence_vmware_rest 他	仮想環境用。VMの強制停止 環境に合わせたエージェントを使用
共有ディスクによる 排他制御	sfex	fence_sbd	sfexよりサービス継続性が向上(stop NG故障、切り替え時間) ハードウェアwatchdogが必須
		fence_scsi	ハードウェアwatchdogが利用不可の環境用(VMware等)
ネットワーク到達性 による排他制御	VIPcheck	sbd(self-fencing) + qdevice	qdeviceノード(もしくは3ノード以上の構成)が必須 ハードウェアwatchdogが必須 (利用条件が大きく異なりそのまま代替とすることは不可)
STONITH相撲ち 防止制御	stonith-helper (standby-waitパラメタ)	delay パラメタ qdevice	優先ノードは固定 standby-wait相当機能(現用系ノード優先)は現在検討中 https://github.com/ClusterLabs/fence-agents/pull/308 qdeviceノード(もしくは3ノード以上の構成)が必須

■ pm_extra_tools (仮)

- pm_pcsgen (仮): pcs設定変換ツール
 - pm_crmgen (Pacemaker-1.1系crm設定変換ツール)のpcs対応版
 - pcsコマンドスクリプトおよび設定一括投入用の一時ファイルを作成
 - Pacemaker-1.1系までとほぼ同様の使用感で利用可能
- pgsql リソースエージェント: PostgreSQL 管理用RA
 - PostgreSQL 12 対応、PG-REX(レプリケーション)対応
 - 開発コミュニティ最新版の修正に追随
- hulft リソースエージェント: HULFT 管理用RA
 - 日本でのニーズが高くPacemaker-1.1系のものをほぼそのまま流用可能
 - HULFTがRHEL 8 正式対応後に検討予定

pm_pcsgen (仮)



CentOS 8 利用時の補足

- CentOS 8 上の Pacemaker は、CentOS 8.1911以降 (RHEL 8.1ベース)で利用できます。
 - CentOS 8.1905 (RHEL 8.0ベース)には Pacemaker は含まれていません。
- Pacemaker のインストールにはオンラインリポジトリを参照する必要があります。
 - ISOメディアには含まれていません。(8.1911 時点)
 - インストールには HighAvailability リポジトリを有効にします。
 - 実行例

```
# dnf install pcs pacemaker fence-agents --enablerepo=HighAvailability
```

もくじ

- Pacemakerとは
 - Linux-HA Japan プロジェクトのこれまでと今後
 - 使い方はどう変わる?
- Pacemaker/HAクラスタの未来

■ クラウド上でのサービス監視

- クラウド機能だけでは対応できない個別のアプリケーション監視
- Multi AZ対応 (AWSでの障害事例)

■ クラウド・コンテナ基盤の冗長化

- OpenStack コントローラノードの冗長化など
- Pacemaker bundle 機能によるコンテナの監視

■ オンプレミスとクラウドのハイブリッド

- 全てのサービスがクラウド化に適しているとは限らない

■ Statefulコンテナの故障対応

- データベースサーバのみ切り出した構成など
- フェンシングの概念の応用

※ 参考資料: OSC2018 Tokyo/Fall セミナー、bundle機能、Statefulコンテナの課題について
「コンテナを止めるな! ~Pacemaker によるコンテナHAクラスタリングと Kubernetes との違いとは」
<http://linux-ha.osdn.jp/wp/archives/4751>

最新動向: HA Cluster Summit 2020 模様



■ HA Cluster Summit とは

- Pacemaker/Corosyncを中心としたHAクラスタ関連の開発者が一堂に会し、現状の振り返りや今後の方向性を議論する開発者会議。2,3年に一度開催。



■ 開催概要

- 期間: 2020年2月5日(水)～2月6日(木)
- 場所: Red Hat 社チェコオフィス
- 参加者: 約45名。主に Red Hat, SUSE, LINBIT他
- URL:
 - http://plan.alteeve.ca/index.php/HA_Cluster_Summit_2020

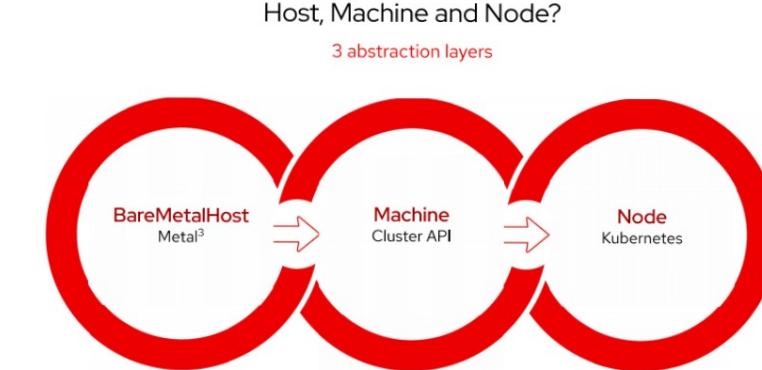
■ 最近の動向

- 参加者が前回の約2倍、HAクラスタ自体の関係者だけでなく、OpenStackやKubernetes、運用監視などの応用についてコミュニティが拡がりつつある。
- Pacemaker本体に大きな機能追加はないが、周辺ツールとの連携(機械処理しやすい出力など)が重要視されはじめている。
- Pacemaker関連コンポーネント(pcs, SBD, Kronosnet等)の細かな改善は継続して行われている。

■ Metal3 (Metal Kubed)

□ 概要

- Kubernetes / OpenShift の物理環境のための管理機能
- Power fencing機能によるHAの改善
Pacemakerの元主要開発者も関与
- Cluster API (Kubernetes SIG)をベアメタル向けに拡張、
Ironicをベースに作成



□ 主な機能

- Health Check
- Power Fencing
 - クラウドでは reboot は意味がないが、物理環境では故障対応として意味がある
- Maintenance mode

ClusterLabs Summit 2020 Red Hat

□ 参照情報

- <https://wiki.clusterlabs.org/wiki/File:CL2020-slides-Mular-Kubernetes.pdf>
- <https://github.com/metal3-io>

■ Linux-HA Japan をこれからもよろしくお願ひします!

