検証におけるPacemaker ログ解析ノウハウ

～Pacemakerの思考を追ってみよう～

OSC2020 Osaka

2020/1/25

Linux-HA Japan

西原　健

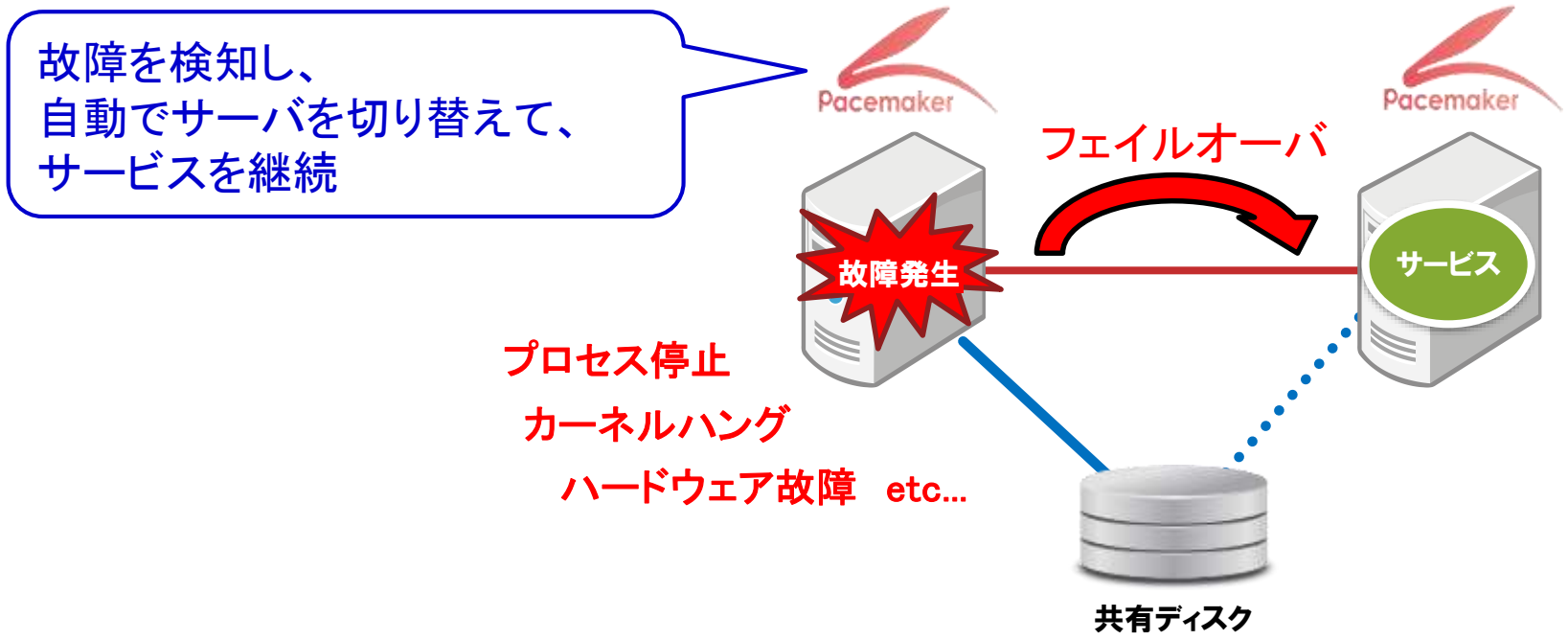
本日の内容

- はじめに
- 検証で想定外の動作が起こったら？
- Pacemakerの思考
- ログから思考を追ってみよう
- まとめ
- コミュニティ紹介
- 予告
- Appendix

はじめに

Pacemakerとは

Pacemakerはオープンソースの**HA**クラスタソフトです。

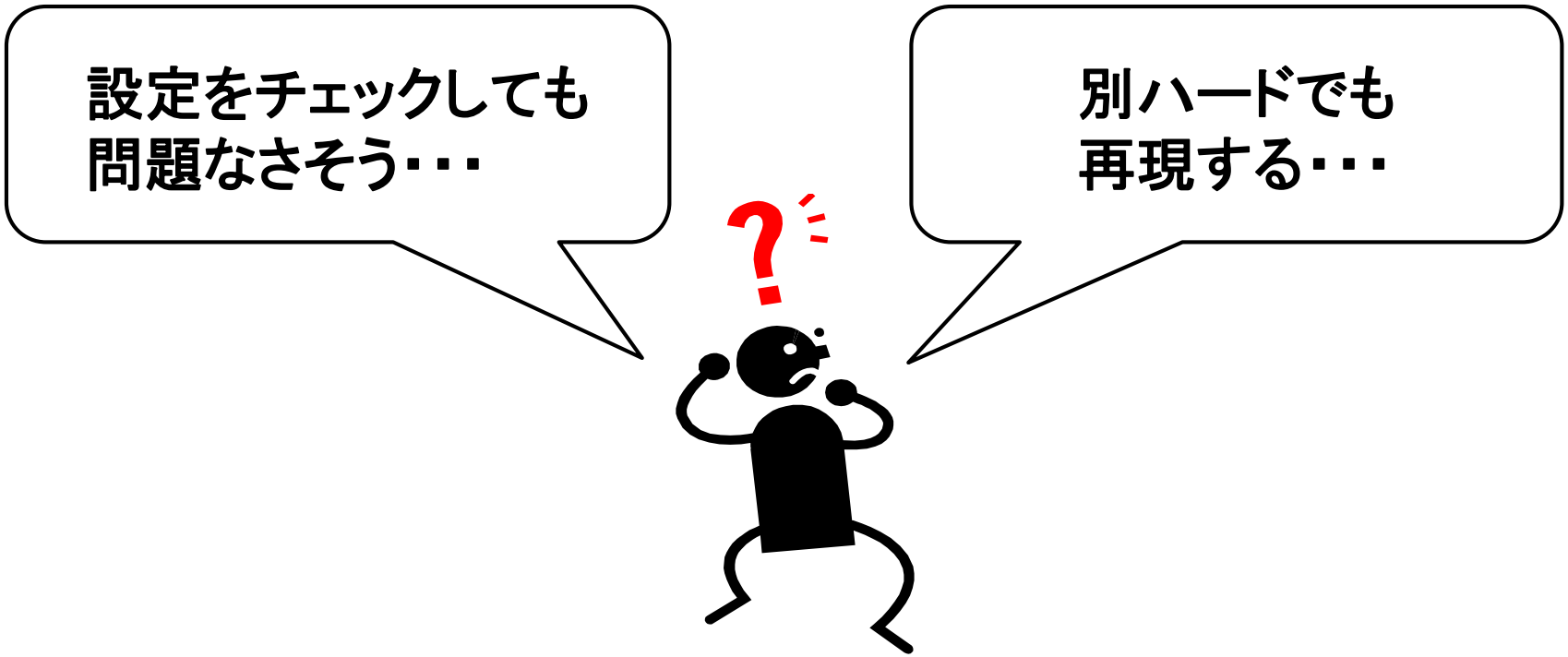


高可用なシステムの構築にあたっては、起こりうる故障について、Pacemakerが想定通り動作するか事前に検証することが重要です。

検証で想定外の動作が起こったら？

想定外の動作が起こった場合

対策の検討のため、原因を調査することになりますが、
簡単にはその原因が分からないケースもあります。

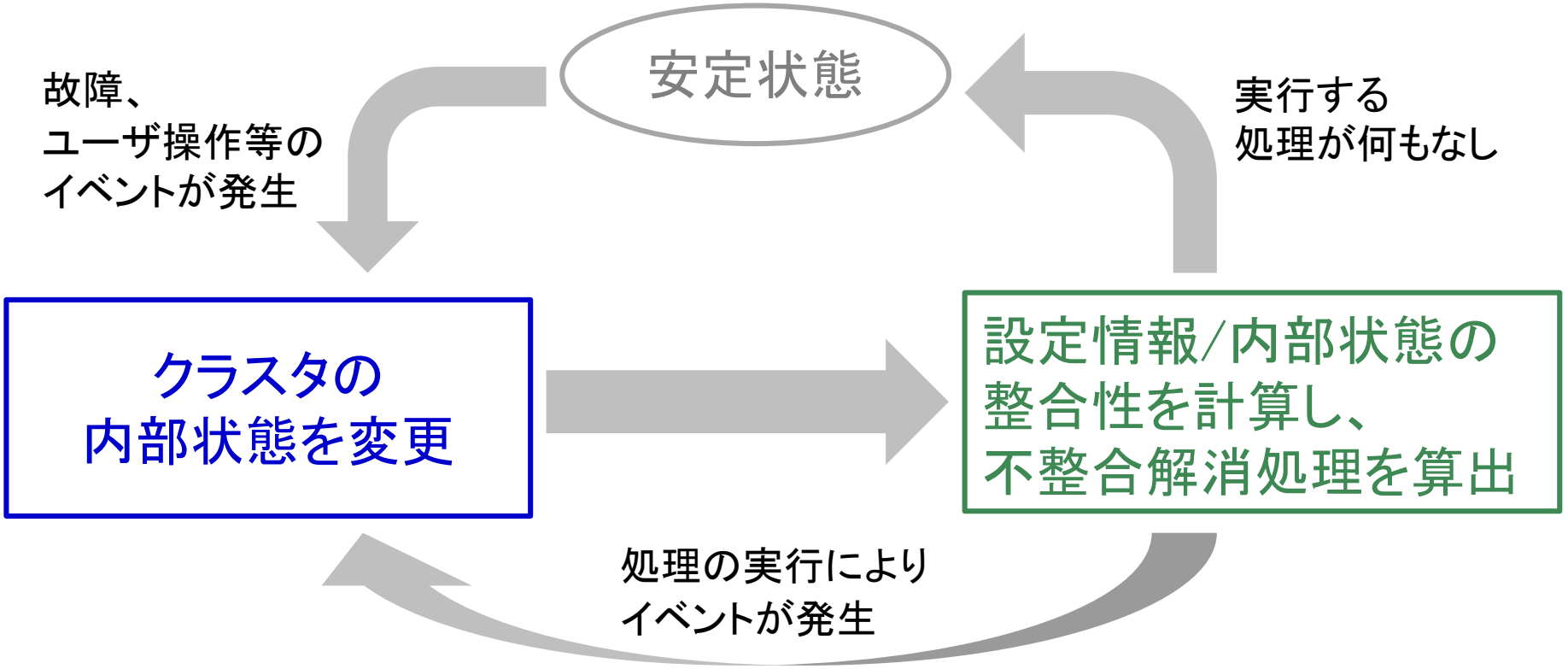


すぐに原因の見当が付かない場合は、なぜその動作に至ったか、
ログからPacemakerの思考を追ってみるのが有用です。

Pacemakerの思考

Pacemakerの思考

Pacemakerは、故障等が発生しても
「内部状態の変更」「整合性の計算」を繰り返し、
最終的に設定と整合性の取れた安定状態へと遷移させるよう思考します。



Pacemakerの思考　＝　「内部状態の変更」＋「整合性の計算」

ログ確認のポイント

Pacemakerのログは量が多いので、
必要なプロセスにポイントを絞って確認することが重要です。

Pacemakerの思考　＝

「内部状態の変更」　＋　「整合性の計算」

cib プロセス

Pacemakerの設定情報/内部状態は
「cib」というxmlに格納されており
cibプロセスによって更新されます。

pengine プロセス

cibの更新を受け、
pengineは整合性を計算し、
実行すべき処理を算出します。
※DCノードのみが実行します

想定外の動作が発生した際は、

1. DCのpengineプロセスのログを追い、想定外の動作を計算したログを探す
2. 想定外の動作が計算された際のcibの内容を確認し、計算の原因を特定
3. 原因となるcibの変更がどこで起きたのか、cibプロセスのログを追う

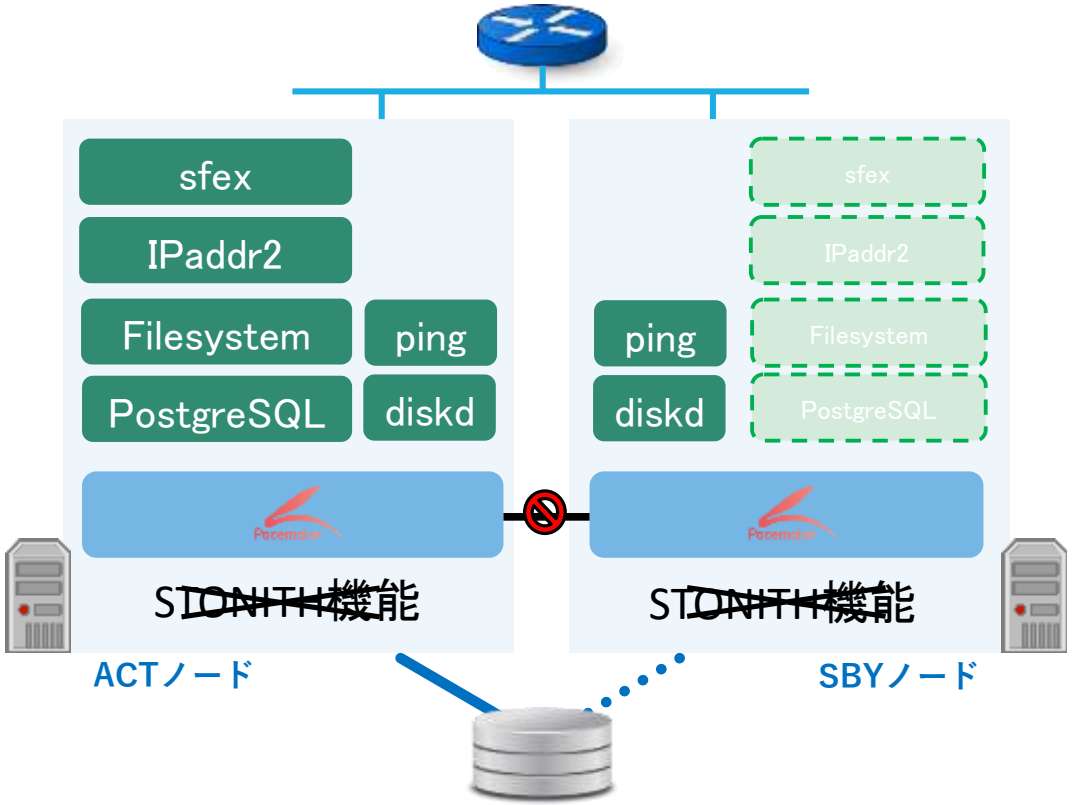
ことで原因に迫ることができます。

ログから思考を追ってみよう

お題

ハートビートLANの断/回復の試験の際の想定外動作

リソース構成



検証時の動作

ハートビートLAN断

SBYノードがsfexを開始

SBYノードでsfexが起動失敗

ハートビートLAN回復

両サーバで
サービス停止

ACTノードがリソースを停止

ACTノードがリソースを再開

ACTノードで想定外のリソース停止が　2019/11/13(水) 15:05:09　に発生

1. DCのpengineプロセスのログを追い、想定外の動作を計算したログを探す
2. 想定外の動作が計算された際のcibの内容を確認し、計算の原因を特定
3. 原因となるcibの変更がどこで起きたのか、cibプロセスのログを追う

①想定外の動作を計算したpengineプロセスのログを特定します。

DC（今回はSBYノード）のログの中から

- 想定外のリソース停止が発生した時刻で
- Stop処理を算出している

pengineプロセスのログが出力されている部分を探します。

```
(snip)
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmPostgreSQL cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmSfex         ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmFilesystem   ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmVIP          ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmPostgreSQL   ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmPing:0#011(Started Node-01)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmPing:1#011(Started Node-02)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd1:0#011(Started Node-01)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd1:1#011(Started Node-02)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd2:0#011(Started Node-01)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd2:1#011(Started Node-02)
(snip)
```

②そのpengineのログをパートごとに分割して内容を確認します

pengineのログは主に4つのパートで構成されています。

```
Nov 13 15:05:09 node02 pengine[2198]:  notice: On loss of CCM Quorum: Ignore
Nov 13 15:05:09 node02 pengine[2198]:    info: Node Node-01 is online
Nov 13 15:05:09 node02 pengine[2198]:    info: Node Node-02 is online
Nov 13 15:05:09 node02 pengine[2198]: warning: Processing failed start of prmSfex on Node-02: unknown error
Nov 13 15:05:09 node02 pengine[2198]:    info: Node 167772517 is already processed
Nov 13 15:05:09 node02 pengine[2198]:    info: Node 167772518 is already processed
Nov 13 15:05:09 node02 pengine[2198]:    info: Node 167772517 is already processed
Nov 13 15:05:09 node02 pengine[2198]:    info: Node 167772518 is already processed
Nov 13 15:05:09 node02 pengine[2198]:    info:  Resource Group: grpPostgreSQL
Nov 13 15:05:09 node02 pengine[2198]:    info:      prmSfex#011(ocf::heartbeat:sfex):#011Started Node-01
Nov 13 15:05:09 node02 pengine[2198]:    info:      prmFilesystem#011(ocf::heartbeat:Filesystem):#011Started Node-01
Nov 13 15:05:09 node02 pengine[2198]:    info:      prmVIP#011(ocf::heartbeat:IPaddr2):#011Started Node-01
Nov 13 15:05:09 node02 pengine[2198]:    info:      prmPostgreSQL#011(ocf::heartbeat:pgsql):#011Started Node-01
Nov 13 15:05:09 node02 pengine[2198]:    info:  Clone Set: clnPing [prmPing]
Nov 13 15:05:09 node02 pengine[2198]:    info:      Started: [ Node-01 Node-02 ]
Nov 13 15:05:09 node02 pengine[2198]:    info:  Clone Set: clnDiskd1 [prmDiskd1]
Nov 13 15:05:09 node02 pengine[2198]:    info:      Started: [ Node-01 Node-02 ]
Nov 13 15:05:09 node02 pengine[2198]:    info:  Clone Set: clnDiskd2 [prmDiskd2]
Nov 13 15:05:09 node02 pengine[2198]:    info:      Started: [ Node-01 Node-02 ]
Nov 13 15:05:09 node02 pengine[2198]:    info: prmSfex has failed INFINITY times on Node-02
Nov 13 15:05:09 node02 pengine[2198]: warning: Forcing prmSfex away from Node-02 after 1000000 failures (max=1)
Nov 13 15:05:09 node02 pengine[2198]:    info: prmSfex: Rolling back scores from prmFilesystem
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmSfex cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:    info: prmFilesystem: Rolling back scores from prmVIP
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmFilesystem cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:    info: prmVIP: Rolling back scores from prmPostgreSQL
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmVIP cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmPostgreSQL cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmSfex        ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmFilesystem    ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmVIP         ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:  notice:  * Stop      prmPostgreSQL    ( Node-01 )   due to node availability
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmPing:0#011(Started Node-01)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmPing:1#011(Started Node-02)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd1:0#011(Started Node-01)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd1:1#011(Started Node-02)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd2:0#011(Started Node-01)
Nov 13 15:05:09 node02 pengine[2198]:    info: Leave   prmDiskd2:1#011(Started Node-02)
Nov 13 15:05:09 node02 pengine[2198]:  notice: Calculated transition 9, saving inputs in /var/lib/pacemaker/pengine/pe-input-550.bz2
```

直前のリソース状態のパート

思考内容のパート

算出された処理のパート

計算に使用したcibのパート

思考内容パートを見ると、
「sfex/Filesystem/VIP/PostgreSQLはどのノードでも起動させてはいけない」
と判断したことが分かります。

思考内容のパート

```
Nov 13 15:05:09 node02 pengine[2198]:    info: prmSfex has failed INFINITY times on Node-02
Nov 13 15:05:09 node02 pengine[2198]: warning: Forcing prmSfex away from Node-02 after 1000000 failures (max=1)
Nov 13 15:05:09 node02 pengine[2198]:    info: prmSfex: Rolling back scores from prmFilesystem
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmSfex cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:    info: prmFilesystem: Rolling back scores from prmVIP
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmFilesystem cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:    info: prmVIP: Rolling back scores from prmPostgreSQL
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmVIP cannot run anywhere
Nov 13 15:05:09 node02 pengine[2198]:    info: Resource prmPostgreSQL cannot run anywhere
```

そして、その判断の根拠となるcibは
/var/lib/pacemaker/pengine/pe-input-550.bz2　に格納されていることも分かります。

計算に使用したcibのパート

```
Nov 13 15:05:09 node02 pengine[2198]:  notice: Calculated transition 9, saving inputs in /var/lib/pacemaker/pengine/pe-input-550.bz2
```

1. DCのpengineプロセスのログを追い、想定外の動作を計算したログを探す
2. 想定外の動作が計算された際のcibの内容を確認し、計算の原因を特定
3. 原因となるcibの変更がどこで起きたのか、cibプロセスのログを追う

③pengineプロセスが計算に使用したcibの内容を確認し、なぜ「sfex/Filesystem/VIP/PostgreSQLはどのノードでも起動させてはいけない」と判断したのか調査します。

cibは主に2つのパートに分かれています。

第1層	第2層	・・・	説明
<cib>	<configuration>		crmファイルで投入した設定情報部分 (リソース設定、制約設定等) 設定情報なので意図的に設定変更しない限り 構築時から変わらない部分
	<status>		属性値やリソース操作履歴などの内部状態部分 内部状態なので故障等のイベントにより 運用中に動的に変動する部分

<configuration>の内容はすでに既知の情報ですので、調査対象は<status>部分となります。

<status>の主要なパートは下記のようになります。

第1層	第2層	第3層	・・・	説明
<status>	<node_state>	<transient_attributes>		リソースの属性情報です。 pingやdiskdの設定する 属性情報などが格納されています。
		<lrm>		リソースの操作履歴です。 主にpengineの 「直前のリソース状態のパート」 を生成するために使われます。

<lrm>の内容は主にpengineの「直前のリソース状態のパート」に関わる部分ですので、
<status>の中でも調査対象は<transient_attributes>部分となります。

```
<status>
  <node_state id="167772517" uname="Node-01" in_ccm="true" crmd="online" crm-debug-origin="do_state_transition" join="member" expected="member">
   <transient_attributes id="167772517">
    <instance_attributes id="status-167772517">
     <nvpair id="status-167772517-ringnumber_0" name="ringnumber_0" value="10.0.1.101 is UP"/>
     <nvpair id="status-167772517-ringnumber_1" name="ringnumber_1" value="10.0.2.101 is UP"/>
    </instance_attributes>
   </transient_attributes>
   <lrm id="167772517">
    （中略）
   </lrm>
  </node_state>
<node_state id="167772518" uname="Node-02" in_ccm="true" crmd="online" crm-debug-origin="do_state_transition" join="member" expected="member">
   <transient_attributes id="167772518">
    <instance_attributes id="status-167772518">
     <nvpair id="status-167772518-ringnumber_0" name="ringnumber_0" value="10.0.1.102 is UP"/>
     <nvpair id="status-167772518-ringnumber_1" name="ringnumber_1" value="10.0.2.102 is UP"/>
     <nvpair id="status-167772518-diskcheck_status_internal" name="diskcheck_status_internal" value="normal"/>
     <nvpair id="status-167772518-diskcheck_status" name="diskcheck_status" value="normal"/>
     <nvpair id="status-167772518-default_ping_set" name="default_ping_set" value="100"/>
     <nvpair id="status-167772518-fail-count-prmSfex.start_0" name="fail-count-prmSfex#start_0" value="INFINITY"/>
     <nvpair id="status-167772518-last-failure-prmSfex.start_0" name="last-failure-prmSfex#start_0" value="1573624977"/>
    </instance_attributes>
   </transient_attributes>
   <lrm id="167772518">
    （中略）
   </lrm>
  </node_state>
 </status>
```

属性値がringnumberしか存在していない

ACTノードの属性情報

SBYノードの属性情報

ACTノードの内部状態を見ると、設定でリソース起動の前提としている
diskdやpingの属性情報が存在していないことが分かります。

1. DCのpengineプロセスのログを追い、想定外の動作を計算したログを探す
2. 想定外の動作が計算された際のcibの内容を確認し、計算の原因を特定
3. 原因となるcibの変更がどこで起きたのか、cibプロセスのログを追う

④cibプロセスがいつACTノードのdiskd/pingの属性情報を消去したのか確認します。

cibプロセスによるcib変更のログから
ACTノードのdiskd/pingの属性情報に関わる部分のみピックアップして追います。

```
Nov 13 15:01:47 node02 cib[2194]:    info: -- /cib/status/node_state[@id='167772517']/transient_attributes[@id='167772517']
```

ACTノードの属性値がすべて消去

cibのxmlのタグの追加/削除/変更について
右記の記号でログ出力されます

++	追加
--	削除
+	変更

調査した状況を時系列で整理することで、今回の事象の原因が分かります。

原因

ハートビートLANの回復後、ACTノードの属性情報が再設定される前に
SBYノード(DC)が整合性計算を行ってしまった。

原因分析から下記のような対策例を考えることができます。

- 暫定対処
  - ハートビートLAN断時には手動でSBYノードを保守者が落とす、というワークアラウンドを実施する。

- 根本対処
  - STONITHの導入を検討する
  - Pacemakerの仕様上正しい動作なのか分析結果を開発コミュニティに共有し、バグならばコード修正を検討してもらう。

※なお、今回の事象はLinux-HA Japanが10/2にリリースしたPacemaker-1.1.21-1.1ではコード修正が行われた結果、大幅に発生頻度が低下しました。

まとめ

まとめ

Pacemakerの思考　＝

「内部状態の変更」＋「整合性の計算」

cib プロセス

pengine プロセス

Pacemakerのログは大量ですが、ポイントを絞れば決して怖くありません。
cib と pengine を味方につけて検証ログの解析を楽しみましょう！

コミュニティ紹介

Linux-HA Japanのご紹介

Linux-HA Japan URL

http://linux-ha.osdn.jp/

https://ja.osdn.net/projects/linux-ha/

Pacemaker関連の最新情報を
日本語で発信

Pacemakerのダウンロードも
こちらからどうぞ
（インストールが楽なリポジトリパッケージを公開しています）

日本におけるHAクラスタについての活発な意見交換の場として
「Linux-HA Japan日本語メーリングリスト」も開設しています

Linux-HA-Japan MLでは、Pacemaker、Corosync
DRBDなど、HAクラスタに関連する話題は歓迎！

- **ML登録用URL**

  http://linux-ha.osdn.jp/
  の「メーリングリスト」をクリック

- **MLアドレス**

linux-ha-japan@lists.osdn.me

※スパム防止のために、登録者以外の投稿は許可制です

予告

開発コミュニティ(ClusterLabs)では現在
Pacemaker-2.0系の開発が進んでいます。
RHEL 8 のHA Add-Onでも採用されています。

Linux-HA Japanでは、RHEL 8/CentOS 8 以降OS同梱のPacemakerをベースに、クラスタ界隈をより盛り上げていこうと準備しています。

詳細は OSC2020 Tokyo/Spring にて！！！！

ご清聴ありがとうございました

Appendix(趣味の世界)

crm_simulateのご紹介

Pacemakerの思考は全てcibの情報がもとになっているため、
cibファイルさえあればいつでも思考を分析することが可能です。

計算に使用したcibのパート

```
Nov 13 15:05:09 node02 pengine[2198]:  notice: Calculated transition 9, saving inputs in /var/lib/pacemaker/pengine/pe-input-550.bz2
```

PacemakerにはcibのファイルをInputすることで
思考をシミュレーションするコマンドが用意されています。

crm_simulate -Sx <cibファイル> -VVVV	infoログを出力
crm_simulate -Sx <cibファイル> -VVVVV	debugログを出力
crm_simulate -Sx <cibファイル> -VVVVVV	traceログを出力

通常の運用環境ではなかなか出力できないdebugやtraceのログも
cibファイルさえあればcrm_simulateを使ってログ出力させることができます。

traceログを出力

今回のセミナーで使用したcibファイルを使ってtraceログを出力してみました。

```
trace:   set_crm_log_level:       New log level: 8
trace:   validate_with:           Validating with: /usr/share/pacemaker/pacemaker-2.10.rng (type=2)
info:    validate_with_relaxng:   Creating RNG parser context
trace:   write_xml_stream:        Writing XML out to /var/lib/pacemaker/cib/shadow.30562
（中略）
notice:  unpack_config:           On loss of CCM Quorum: Ignore
trace:   unpack_config:           Orphan resources are stopped
trace:   unpack_config:           Orphan resource actions are stopped
trace:   unpack_config:           Stopped resources are removed from the status section: false
trace:   unpack_config:           Maintenance mode: false
(中略)
debug:   group_rsc_location:      Processing rsc_location rsc_location-grpPostgreSQL-1-rule-3 for grpPostgreSQL
trace:   native_rsc_location:     Applying rsc_location-grpPostgreSQL-1-rule-3 (Unknown) to grpPostgreSQL
trace:   native_rsc_location:     Node-02 + Node-02: 100 + 0
trace:   merge_weights:           0 + 100 = 100
trace:   native_rsc_location:     grpPostgreSQL + Node-01 : 0
trace:   native_rsc_location:     grpPostgreSQL + Node-02 : 100
trace:   native_rsc_location:     Applying rsc_location-grpPostgreSQL-1-rule-3 (Unknown) to prmSfex
trace:   native_rsc_location:     Node-02 + Node-02: 100 + 0
trace:   merge_weights:           0 + 100 = 100
(中略)
trace:   crm_clear_bit:           Bit 0x00000001 cleared by cib_file_signoff:708
trace:   crm_exit:                cleaning up libxml
info:    crm_xml_cleanup:         Cleaning up memory from libxml2
trace:   crm_exit:                exit 0
```

12639行

膨大ですが、Pacemakerの思考をたどる上での究極のログと言えます。
関数名から思考とPacemakerのコードを結びつけて分析すれば、
Pacemakerの内部動作についてより理解を深めることができる、かも？

traceログで皆さんも深淵を覗こう！