【P4D】 Heartbeat機能によるP4Dサーバの監視

概要

Helix Core2020.1より追加されたHeartbeat機能を使用し、稼働中のP4Dサーバを監視することが可能になりました。

Heartbeat機能と以下で紹介する3つのトリガを組み合わせることで、監視ステータスに応じた動作を事前に定義することができます。

ここでは、Heartbeat機能の詳細・設定方法、トリガの設定事例について紹介しています。

説明

1. Heartbeat機能

P4DサーバのHeartbeat機能は、"p4 heartbeat"コマンドが実行されることで動作します。
このコマンドは、以下の5つの構成可能変数で設定されたデフォルト値を使用して実行されます。

構成可能変数の説明

・net.heartbeat.interval (ms): 2000
　ターゲットサーバにHeartbeatを送信する間隔。

・net.heartbeat.wait (ms): 2000
　ターゲットサーバからの応答を待機する時間。
　この時間が経過しても最初の応答が返ってこない場合は、"heartbeat-missing"トリガが起動。

・net.heartbeat.missing.interval (ms): 2000
　応答が返ってこなかった場合にターゲットサーバにHeartbeatを送信する間隔。

・net.heartbeat.missing.wait (ms): 4000
　応答が返ってこなかった場合にターゲットサーバからの応答を待機する時間。
　要求がタイムアウトになる前にターゲットサーバからの応答が返ってきた場合は、
　"heartbeat-resumed"トリガが起動。

・net.heartbeat.missing.count: 5
　Heartbeatが停止していると判断するための、ハートビート応答が連続して返ってこなかった回数。
　ターゲットサーバからの応答が連続して返ってこなかった回数がこの値に達すると、
　"heartbeat-dead"トリガが起動。

デフォルト設定では、2秒間隔でHeartbeat通信がターゲットとなるP4Dサーバに送信され、2秒後にターゲットサーバから通信の返却がない場合は、ターゲットサーバが通信に問題が発生している可能性があると判断します。
しかし、4秒後までにターゲットサーバから通信が戻ってきた場合は、問題が解消したと判断します。

2. Heartbeatの設定方法

P4Dサーバの起動とともに、Heartbeat機能(p4 heartbeat)コマンドが実行されるように構成可能変数"startup.N"を使用します。
構成可能変数を設定するためには、"p4 configure set"コマンドを使用します。

設定コマンド例
p4 -u super -p server:port configure set standby#startup.4=heartbeat

※ super：スーパ権限ユーザを指定
※ server:port：対象P4Dサーバを指定

Heartbeat機能に関連する構成可能変数を個別に変更する場合は、同様に"p4 configure set"コマンドを使用します。

設定コマンド例
p4 -u super -p server:port configure set net.heartbeat.interval=3000

※ super：スーパ権限ユーザを指定
※ server:port：対象P4Dサーバを指定

結果例：ターゲットサーバから応答が届かなかった際のメッセージ

```
Perforce server error:
        2020/11/24 21:14:14 pid 2174 service3@18800 background 'heartbeat'
        Heartbeat missing: access to 192.168.56.14:18800 server failed.
        Heartbeat response exceeded maximum wait duration of 2000 milliseconds.
```

3. トリガの設定

Heartbeat機能には、次の３つのトリガが準備されています。

・heartbeat-missing：
　ターゲットサーバからの応答が指定した待機時間を超えた場合、このトリガが起動します。

・heartbeat-resumed：
　要求がタイムアウトになる前にターゲットサーバからの応答が返ってきた場合にこのトリガが起動します。

・heartbeat-dead：
　ターゲットサーバからの応答が連続して返ってこなかった回数が指定値を超えるとこのトリガが起動します。

この3つのトリガをトリガテーブルに設定することで、各条件に合わせた動作を定義することができます。

ここでは、OSのloggerコマンドを用いてOSのシステムログへメッセージを出力する例を紹介します。
①例で使用するシェルスクリプトは、以下のとおりです。

```
hm.sh
#!/usr/bin/bash

TARGET=$1
logger "$TARGETとの通信が切断された可能性があります。"

exit

hr.sh
#!/usr/bin/bash

TARGET=$1
logger "$TARGETとの通信が回復しました。"

exit

hd.sh
#!/usr/bin/bash

TARGET=$1
logger "$TARGETが停止しています。フェイルオーバーを実施してください。"

exit
```

注意

これらのシェルスクリプトは、ターゲットとなるP4Dサーバを監視するP4Dサーバに配置します。
ここでは、/opt/perforce配下に配置した例で説明します。

②トリガテーブルを定義します。
　トリガテーブルの定義は、ターゲットサーバ(マスタサーバ)にて実施します。

トリガテーブル設定コマンド例

```
p4 -u super -p server:port triggers

Triggers:
        hm heartbeat-missing heartbeat "/opt/perforce/hm.sh  %targetport%"
        hr heartbeat-resumed heartbeat "/opt/perforce/hr.sh  %targetport%"
        hd heartbeat-dead heartbeat "/opt/perforce/hd.sh  %targetport%"
```

実際に、トリガが実行されると以下のメッセージがOSのシステムログへ出力されます。

ログ例：ターゲットから応答がない場合
Dec 22 21:37:18 standby root: 192.168.56.14:18800との通信が切断された可能性があります。

ログ例：ターゲットから応答が回復した場合
Dec 22 21:37:24 standby root: 192.168.56.14:18800との通信が回復しました。

ログ例：ターゲットからの応答が定義された回数以上ない場合
Dec 22 21:41:27 standby root: 192.168.56.14:18800が停止しています。フェイルオーバーを実施してください。

ここでは簡単なスクリプトを使用していますが、スクリプト内でフェイルオーバー等を実施することで、より安定性の高いP4Dサーバの運用をしていただくことができます。