

CLUSTERPRO MC
ApplicationMonitor 2.3 for Linux
を 5 ノード以上の RAC 構成で
使用する場合の補足事項

© 2018(Apr) NEC Corporation

多ノード RAC 構成用の設定

はしがき

本書は、CLUSTERPRO MC ApplicationMonitor 2.3 for Linux(以後 ApplicationMonitor と記載します)を 5 ノード以上の RAC 構成で使用する場合の設定について記載したものです。

(1) 本書は、以下の OS に対応します。

- ・ Red Hat Enterprise Linux 6.4～6.9, 7.0～7.4
- ・ Oracle Linux 6.4～6.9, 7.0～7.4

(2) 本書で説明しているすべての機能は、プログラムプロダクトであり、次の表のプロダクト型番およびプロダクト名に対応しています。

| プロダクト型番 | プロダクト名 | プロダクトリリース |
|------------|--|-----------|
| UL4437-702 | CLUSTERPRO MC ApplicationMonitor 2.3 for Linux (1CPU ライセンス) | 2.3 |
| UL4437-712 | CLUSTERPRO MC ApplicationMonitor 2.3 for Linux (1CPU ライセンス)(他社機版) | 2.3 |
| UL4437-722 | CLUSTERPRO MC ApplicationMonitor 2.3 for Linux VM (1 ノードライセンス) | 2.3 |
| UL4437-732 | CLUSTERPRO MC ApplicationMonitor 2.3 for Linux VM (1 ノードライセンス)(他社機版) | 2.3 |
| UL4437-701 | CLUSTERPRO MC ApplicationMonitor CD 2.3 | 2.3 |

(3) 本書は、次の表現を使用します。

- ・ 大かっこ [] 省略可能なパラメータを表します。
- ・ 山かっこ <> 任意の文字列を指定することを表します。

(4) 商標および登録商標

- ✓ Linux は、Linus Torvalds 氏の米国およびその他の国における、登録商標または商標です。
- ✓ CLUSTERPRO は、日本電気株式会社の登録商標です。
- ✓ Oracle と Java は、Oracle Corporation およびその子会社、関連会社の米国およびその他の国における登録商標です。
- ✓ その他、本書に登場する会社名および商品名は各社の商標または登録商標です。
- ✓ なお、本書では®、TM マークを明記しておりません。

目次

| | |
|---------------------------------------|---|
| 1. 多ノード RAC 構成用の設定 | 1 |
| 1.1. SLAVE からの状態通知の待ち合わせ時間の設定 | 2 |
| 1.2. ハートビートの送受信間隔の設定 | 3 |
| 1.3. 設定ファイル (oramond.conf) の記述例 | 5 |
| 1.4. 設定の確認 | 7 |
| 1.5. 設定の配布 | 7 |

1. 多ノード RAC 構成用の設定

5 ノード以上の RAC 構成で ApplicationMonitor を使用する場合、ノード間通信のチューニングを行うため、別冊の『CLUSTERPRO MC ApplicationMonitor 2.3 for Linux ユーザーズガイド』に記載している通常の設定に加え、以下の設定が必要です。

- ◆ MASTER (※1)決定後に SLAVE (※2)からの状態通知を待ち合わせる時間の設定
- ◆ ハートビート(※3)の送受信間隔の設定

(※1) MASTER

各ノード上で動作するモニタ制御デーモン (oramond) の情報を取りまとめ、管理するモニタ制御デーモンを指します。

(※2) SLAVE

MASTER が管理するモニタ制御デーモンを指します。

(※3) ハートビート

各ノード上で動作するモニタ制御デーモン間で行っている状態確認用の通信を指します。

これらの設定についての手順を説明します。

1.1. SLAVE からの状態通知の待ち合わせ時間の設定

ApplicationMonitor では、各ノードの ApplicationMonitor および監視対象の情報を管理するため、起動時に MASTER を選出します。

通常、すべてのノード上でモニタ制御デーモン (oramond) が起動している場合、MASTER 選出後に MASTER が SLAVE の状態を(例 1)のように syslog に出力します。

(例 1)状態通知が規定時間内に行われた場合の syslog

```
oramond[<pid>]: 0:Forming an AM cluster as MASTER.
oramond[<pid>]: 0:AM cluster membership:
oramond[<pid>]: 0: 0 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 1 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 2 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 3 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 4 <NODE_NAME>           ALIVE
```

しかし、多ノード RAC 構成の環境においては、すべてのノード上でモニタ制御デーモンが起動している状態にもかかわらず、管理対象の SLAVE が多いことに起因し、(例 2)のように一部のノードの状態が確認できない場合があります。

(例 2)状態が確認できていない場合の syslog

```
oramond[<pid>]: 0: 0 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 1 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 2 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 3 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 4 <NODE_NAME>           unreachable
```

上記の現象を回避するため、MASTER 決定後に SLAVE からの状態通知を待ち合わせる時間を調整し、設定する必要があります。

設定ファイル (oramond.conf) の全ノードステートメントに以下のパラメータを追記してください。

| パラメータ名 | 意味 | |
|----------------------|---|--------|
| | 省略時のデフォルト値 | 値の範囲 |
| ELECTION_ACK_TIMEOUT | MASTER 決定後に SLAVE からの状態通知を待ち合わせる時間を秒で指定します。 | |
| | 10 秒 | 1~3600 |

1.2. ハートビートの送受信間隔の設定

ApplicationMonitor では、MASTER 選出完了後も各ノードのモニタ制御デーモン (oramond) の状態を定期的に監視するため、規定間隔に 1 回状態確認用の通信を行います。

多ノード RAC 構成の環境においては、管理対象の SLAVE 数によって規定間隔ではハートビートを送信できず、SLAVE の動作するノードの syslog に "Heartbeat timeout." というエラーメッセージが出力される可能性があります。

上記の現象を回避するため、モニタ制御デーモン間で行っている状態確認用の通信の送受信時間を調整し、設定する必要があります。

設定ファイル (oramond.conf) の全ノードステートメントに以下のパラメータを追記してください。

| パラメータ名 | 意味 | |
|--------------------|----------------------------------|--------|
| | 省略時のデフォルト値 | 値の範囲 |
| HEARTBEAT_INTERVAL | モニタ制御デーモン間の状態確認用通信の実施間隔を秒で指定します。 | |
| | 10 秒 | 1~3600 |

1 多ノード RAC 構成用の設定

ELECTION_ACK_TIMEOUT パラメータおよび HEARTBEAT_INTERVAL パラメータを同時に
見直す必要がある場合があります。

通常、すべてのノードで ApplicationMonitor が正常に起動すると、(例 3)のように MASTER および SLAVE の選出が正常に完了したメッセージを syslog に出力します。

(例 3)選出が正常に完了した場合の syslog

```
oramond[<pid>]: 0:Detecting other AM.
oramond[<pid>]: 0:Forming an AM cluster as MASTER.
oramond[<pid>]: 0:AM cluster membership:
oramond[<pid>]: 0: 0 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 1 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 2 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 3 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 4 <NODE_NAME>           ALIVE
oramond[<pid>]: 0:Permitting Oracle access on node #0.
oramond[<pid>]: 0:Permitting Oracle access on node #1.
oramond[<pid>]: 0:Permitting Oracle access on node #2.
oramond[<pid>]: 0:Permitting Oracle access on node #3.
oramond[<pid>]: 0:Permitting Oracle access on node #4.
oramond[<pid>]: 0:Forming complete.      ← MASTER および SLAVE の選出が正常
                                         に完了したメッセージ
```

しかし、多ノード RAC 構成の環境の場合、各ノードからの状態確認メッセージが遅延し、(例 4)のように状態確認途中で再度 MASTER の選出が繰り返し発生してしまう可能性があります。

(例 4)選出が繰り返し発生してしまう場合の syslog

```
oramond[<pid>]: 0:Detecting other AM.
oramond[<pid>]: 0:Forming an AM cluster as MASTER.
oramond[<pid>]: 0:AM cluster membership:
oramond[<pid>]: 0: 0 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 1 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 2 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 3 <NODE_NAME>           ALIVE
oramond[<pid>]: 0: 4 <NODE_NAME>           ALIVE
oramond[<pid>]: 0:Permitting Oracle access on node #0.
oramond[<pid>]: 0:Permitting Oracle access on node #1.
oramond[<pid>]: 0:Permitting Oracle access on node #2.
oramond[<pid>]: 0:Permitting Oracle access on node #3.
                                         ← node#4 の状態確認が時間内に行えないため、
                                         再度 MASTER の選出が発生

oramond[<pid>]: 0:Detecting other AM.
oramond[<pid>]: 0:Forming an AM cluster as MASTER.
oramond[<pid>]: 0:AM cluster membership:
<以下、省略>
```

上記の現象が発生した場合、ELECTION_ACK_TIMEOUT パラメータおよび HEARTBEAT_INTERVAL パラメータの双方を調整し、設定してください。

1.3. 設定ファイル (oramond.conf) の記述例

設定ファイルは、コメント、パラメータ、ステートメントによって構成されたテキストファイルです。

5 ノード RAC 構成のインスタンス監視、単独起動のリスナー監視の設定に、多ノード RAC 構成時の調整パラメータを追加した設定は、以下のように記述します。

```

NODE_NAME = node1
NODE_NAME = node2
NODE_NAME = node3
NODE_NAME = node4
NODE_NAME = node5

MONTYPE = 12CR2

ORACLE_BASE = /u01/app/oracle
ORACLE_HOME = /u01/app/oracle/product/12.2.0/dbhome_1
ORA_NLS = /u01/app/oracle/product/12.2.0/dbhome_1/nls/data
SHLIB_PATH = /u01/app/oracle/product/12.2.0/dbhome_1/lib
NLS_LANG = japanese_japan.ja16euc
GRID_BASE = /u01/app/grid
GRID_HOME = /u01/app/12.2.0/grid
GRID_USER = grid

ORACLE_HANG_CHECK = YES
CHECK_WAIT_EVENT = YES

ELECTION_ACK_TIMEOUT = 40
HEARTBEAT_INTERVAL = 40

InstanceMonitor {
    OWNER_NAME = SYS
}

Node node1 {
    ORACLE_SID = sid1

    ListenerMonitor LISTENER1 {
        MONITOR_CONTROL = NO
        NET_SERVICE_NAME = LSNR1
    }
}

Node node2 {
    ORACLE_SID = sid2

    ListenerMonitor LISTENER2 {
        MONITOR_CONTROL = NO
        NET_SERVICE_NAME = LSNR2
    }
}

Node node3 {
    ORACLE_SID = sid3

    ListenerMonitor LISTENER3 {
        MONITOR_CONTROL = NO
        NET_SERVICE_NAME = LSNR3
    }
}

Node node4 {
    ORACLE_SID = sid4

    ListenerMonitor LISTENER4 {
        MONITOR_CONTROL = NO

```

1 多ノード RAC 構成用の設定

```
        NET_SERVICE_NAME = LSNR4
    }
}
Node node5 {
    ORACLE_SID = sid5

    ListenerMonitor LISTENER5 {
        MONITOR_CONTROL = NO
        NET_SERVICE_NAME = LSNR5
    }
}
```

1.4. 設定の確認

ApplicationMonitor の管理コマンド (oraadmin) の check リクエストを用いて、修正した設定ファイル (oramond.conf) の内容を確認します。

管理コマンドの詳細は、別冊の『CLUSTERPRO MC ApplicationMonitor 2.3 for Linux ユーザーズガイド』を参照してください。

1.5. 設定の配布

ApplicationMonitor の管理コマンド (oraadmin) の apply リクエストを用いて、修正した設定ファイル (oramond.conf) から構成情報ファイル (oramond<構成番号>.bin) を生成し、設定した各ノードに配布します。

なお、本書に記載しているパラメータは、ApplicationMonitor が動作中の場合も変更が可能です。

管理コマンドの詳細は、別冊の『CLUSTERPRO MC ApplicationMonitor 2.3 for Linux ユーザーズガイド』を参照してください。

以上で、ApplicationMonitor を 5 ノード以上の RAC 構成で使用可能となります。

1 多ノード RAC 構成用の設定

CLUSTERPRO
MC ApplicationMonitor 2.3 for Linux
を5ノード以上の RAC 構成で使用する場合の
補足事項

2018年 4月 第1版
日本電気株式会社
東京都港区芝五丁目7番地1号
TEL (03) 3454-1111(代表)

Ⓟ

© NEC Corporation 2012-2018

日本電気株式会社の許可なく複製、改変などを行うことはできません。
本書の内容に関しては将来予告なしに変更することがあります。

保護用紙