

---

**CLUSTERPRO MC**  
**ApplicationMonitor 1.0 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**

2013年4月  
第2版

---

## 備考

(1) 本書は、以下のオペレーティングシステムに対応しています。

- Red Hat Enterprise Linux 5.7, 5.8, 6.1, 6.2, 6.3
- Oracle Linux 6.1, 6.2, 6.3

ただし、OS がバージョンアップする際に、本書の内容が変更される場合があります。

(2) 本書で説明している全ての機能は、プログラムプロダクトであり、次の表のプロダクト型番及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
UL4437-102	CLUSTERPRO MC ApplicationMonitor 1.0 for Linux (1CPU ライセンス)	1.0
UL4437-112	CLUSTERPRO MC ApplicationMonitor 1.0 for Linux (1CPU ライセンス)(他社機版)	1.0
UL4437-122	CLUSTERPRO MC ApplicationMonitor 1.0 for Linux VM (1 ノードライセンス)	1.0
UL4437-132	CLUSTERPRO MC ApplicationMonitor 1.0 for Linux VM (1 ノードライセンス)(他社機版)	1.0
UL4437-101	CLUSTERPRO MC ApplicationMonitor CD 1.0	1.0

(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)

- Linux は、Linus Torvalds 氏の米国及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 は、米国 Oracle Corporation の登録商標です。

その他、本書に登場する会社名、製品名は一般に各社の登録商標または商標です。

© 2012-2013 NEC Corporation

## 目 次

ApplicationMonitor の多ノード RAC 構成用設定 .....	1
1. ELECTION_ACK_TIMEOUT の設定 .....	2
2. HEARTBEAT_INTERVAL の設定 .....	3
3. 設定ファイルの記述例 .....	5
4. AM の設定確認 .....	6
5. AM の設定配布 .....	6



## ApplicationMonitor の多ノード RAC 構成用設定

5 ノード以上の RAC 構成で、ApplicationMonitor(以下、AM と略す)を利用する場合、ノード間通信のチューニングを行うため、マニュアル『CLUSTERPRO MC ApplicationMonitor 1.0 for Linux 利用の手引き』に記載されている通常の設定に加え、以下の設定が必要です。

- ・ MASTER (注 1) 決定後に SLAVE からの状態通知を待ち合わせる時間 (**ELECTION\_ACK\_TIMEOUT**) の設定
- ・ ハートビート(注 2)の送受信間隔(**HEARTBEAT\_INTERVAL**)の設定

### (注 1) MASTER と SLAVE

各ノード上で動作するモニタ制御部 (oramond) の情報を取りまとめ、管理しているモニタ制御部を MASTER と呼び、その MASTER に管理されるモニタ制御部を SLAVE と呼びます。

### (注 2) ハートビート

各ノード上で動作する、モニタ制御部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、順に説明します。

### 1. ELECTION\_ACK\_TIMEOUT の設定

AMでは、各ノードのAM及び監視対象の情報を管理するため、起動時にMASTERを選出する処理を行います。

通常、全てのノード上でモニタ制御部が起動している場合、MASTER選出後、MASTERは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構成のシステムにおいては、**全てのノード上でモニタ制御部が起動している状態にも関わらず**、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
```

上記現象を回避するため、ELECTION\_ACK\_TIMEOUTパラメータを調整し、設定する必要があります。

MASTER決定後にSLAVEからの状態通知を待ち合わせる時間を設定するために、設定ファイル(oramond.conf)の全ノードステートメントに次のパラメータを追記してください。

パラメータ名	意味
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。指定できる範囲は 1 ~ 3600 です。省略時は 10 が設定されます。

## 2. HEARTBEAT\_INTERVAL の設定

AMでは、MASTER選出完了後も、各ノードのモニタ制御部の状態を定期的に監視するため、規定間隔(10秒)に一回状態確認用の通信を行います。

多ノードRAC構成のシステムにおいては、管理対象のSLAVE数により、規定間隔で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."エラーが出力される可能性があります。

上記現象を回避するため、HEARTBEAT\_INTERVALパラメータを調整し、設定する必要があります。

各ノード上で動作する、モニタ制御部間で行っている状態確認用の通信の送受信時間を設定するために、設定ファイル(oramond.conf)の全ノードステートメントに次のパラメータを追記してください。

パラメータ名	意味
HEARTBEAT_INTERVAL	モニタ制御部間の状態確認用通信の実施間隔を秒で指定します。指定できる範囲は1~3600です。省略時は10が設定されます。

<補足>1.及び2.のパラメータの同時見直しが必要なケース

通常、AMの全てが正常に起動すると、(例3)のようにMASTER及びSLAVEの選出が正しく完了したメッセージがsyslogに出力されます。

(例3)選出が正しく完了する場合の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.    <- 正常に選出が行えた時のメッセージ
```

しかし、多ノードRAC構成の場合、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の選出が始まってしまう現象が発生する可能性があります。

(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.
                    <- #4 の状態確認が時間内に行えないため、再度 MASTER 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>
```

上記現象が発生した場合、ELECTION\_ACK\_TIMEOUTパラメータ及びHEARTBEAT\_INTERVALパラメータの双方を調整し、設定してください。

### 3. 設定ファイルの記述例

設定ファイルは、コメント、パラメータ、パラメータ群を囲むステートメントにより構成されたテキストファイルです。ダブルクォーテーション(")で囲まれた文字列以外は全て半角英数文字で記述します。

AM(5 ノード RAC 構成)のインスタンス監視、リスナ監視(単独起動)の設定に多ノード RAC 構成時の調整パラメータ(任意パラメータ含む)を加えた設定は、以下のように記述します(例 5)。

#### (例 5)5 ノード RAC 構成簡易設定例

```

NODE_NAME = node1
NODE_NAME = node2
NODE_NAME = node3
NODE_NAME = node4
NODE_NAME = node5
MONTYPE = 11GR2

ORACLE_BASE = /oracle
ORACLE_HOME = /oracle/product/11.2.0
ORA_NLS = /oracle/product/11.2.0/nls/data
SHLIB_PATH = /oracle/product/11.2.0/lib
NLS_LANG = japanese_japan.ja16euc

GRID_BASE = /grid
GRID_HOME = /11.2.0/grid
GRID_USER = grid

ELECTION_ACK_TIMEOUT = 40
HEARTBEAT_INTERVAL = 40

Node node1 {
  ORACLE_SID = c1db1
  ListenerMonitor LSNR_RIP1 {
    MONITOR_CONTROL = NO
    NET_SERVICE_NAME = RIP1
  }
}

Node node2 {
  ORACLE_SID = c1db2
  ListenerMonitor LSNR_RIP2 {
    MONITOR_CONTROL = NO
    NET_SERVICE_NAME = RIP2
  }
}

Node node3 {
  ORACLE_SID = c1db3
  ListenerMonitor LSNR_RIP3 {
    MONITOR_CONTROL = NO
    NET_SERVICE_NAME = RIP3
  }
}

Node node4 {
  ORACLE_SID = c1db4
  ListenerMonitor LSNR_RIP4 {
    MONITOR_CONTROL = NO
    NET_SERVICE_NAME = RIP4
  }
}

Node node5 {
  ORACLE_SID = c1db5
  ListenerMonitor LSNR_RIP5 {
    MONITOR_CONTROL = NO
    NET_SERVICE_NAME = RIP5
  }
}

```

## ApplicationMonitor の多ノード RAC 構成用設定

### 4. AM の設定確認

AM の管理コマンド“oraadmin”を使って、修正した設定ファイルの内容をチェックします。コマンドの用法については、マニュアル『CLUSTERPRO MC ApplicationMonitor 1.0 for Linux 利用の手引き』を参照してください。

### 5. AM の設定配布

AM の管理コマンド“oraadmin”を使って、修正した設定ファイルの内容を設定ノードに配布します。コマンドの用法については、マニュアル『CLUSTERPRO MC ApplicationMonitor 1.0 for Linux 利用の手引き』を参照してください。

なお、本補足ガイドに記載されているパラメータは、動的変更可能パラメータですので、AM 動作中の変更も可能となります。

以上で、AM を 5 ノード以上の RAC 構成で使用可能となります。