



Red Hat Advanced Server での Caché ECP クラスタ

Version 5.1
2006-03-14

Red Hat Advanced Server での Caché EOP クラス

Caché Version 5.1 2006-03-14

Copyright © 2006 InterSystems Corporation.

All rights reserved.

このドキュメントは、Sun Microsystems、RenderX Inc.、アドビ システムズ および ワールドワイド・ウェブ・コンソーシアム (www.w3c.org) のツールと情報を使用して、Adobe Portable Document Format (PDF) で作成およびフォーマットされました。主要ドキュメント開発ツールは、InterSystemsが構築したCaché と Javaを使用した特別目的のXML処理アプリケーションです。



Caché 製品とロゴは InterSystems Corporation の登録商標です。



Ensemble 製品とロゴは InterSystems Corporation の登録商標です。



InterSystems という名前とロゴは InterSystems Corporation の登録商標です

このドキュメントは、インターシステムズ社(住所: One Memorial Drive, Cambridge, MA 02142)あるいはその子会社が所有する企業秘密および秘密情報を含んでおり、インターシステムズ社の製品を稼動および維持するためにのみ提供される。この発行物のいかなる部分も他の目的のために使用してはならない。また、インターシステムズ社の書面による事前の同意がない限り、本発行物を、いかなる形式、いかなる手段で、その全てまたは一部を、再発行、複製、開示、送付、検索可能なシステムへの保存、あるいは人またはコンピュータ言語への翻訳はしてはならない。

かかるプログラムと関連ドキュメントについて書かれているインターシステムズ社の標準ライセンス契約に記載されている範囲を除き、ここに記載された本ドキュメントとソフトウェアプログラムの複製、使用、廃棄は禁じられている。インターシステムズ社は、ソフトウェアライセンス契約に記載されている事項以外にかかるソフトウェアプログラムに関する説明と保証をするものではない。さらに、かかるソフトウェアに関する、あるいはかかるソフトウェアの使用から起こるいかなる損失、損害に対するインターシステムズ社の責任は、ソフトウェアライセンス契約にある事項に制限される。

前述は、そのコンピュータソフトウェアの使用およびそれによって起こるインターシステムズ社の責任の範囲、制限に関する一般的な概略である。完全な参照情報は、インターシステムズ社の標準ライセンス契約に記載され、そのコピーは要望によって入手することができる。

インターシステムズ社は、本ドキュメントにある誤りに対する責任を放棄する。また、インターシステムズ社は、独自の裁量にて事前通知なしに、本ドキュメントに記載された製品および実行に対する代替と変更を行う権利を有する。

Caché および InterSystems Caché、Caché SQL、Caché ObjectScript および Caché Object は、インターシステムズ社の商標です。

ここで使われている他の全てのブランドまたは製品名は、各社および各組織の商標または登録商標です。

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください:

InterSystems ワールドワイド カスタマサポート

Tel: +1 617 621-0700

Fax: +1 617 374-9391

Email: support@InterSystems.com

目次

Red Hat Advanced Server での Caché ECP クラスタ.....	1
1 インストール前の計画	1
2 Caché に対するクラスタ・サービスの構成	2
2.1 Caché クラスタ・サービスの定義	3
2.2 Caché のインストール	4
3 第 2 ノードの構成	5
4 クラスタ・サービスへの Caché の追加	5
4.1 Linux 用の Caché 初期化ファイル	7
5 アップグレード時の Caché レジストリの保守	9

Red Hat Advanced Server での Caché ECP クラスタ

Caché ECP クラスタは、特定の ECP データ・サーバから別の ECP データ・サーバへのフェイルオーバーを実現する高可用性機能で、障害が発生したサーバの検出にはオペレーティング・システム・レベルでのクラスタ化が使用されます。Caché ECP クラスタ・テクノロジーはテスト済みで、Red Hat Linux Advanced Server バージョン 2.1 でサポートされます。このドキュメントでは、クラスタの構成方法の詳細を説明します。また、このドキュメントは以下のセクションで構成されます。

- ・ [インストール前の計画](#)
- ・ [Caché に対するクラスタ・サービスの構成](#)
- ・ [第 2 ノードの構成](#)
- ・ [クラスタ・サービスへの Caché の追加](#)
- ・ [アップグレード時の Caché レジストリの保守](#)

Red Hat Advanced Server でのクラスタ構成の詳細は、“Red Hat Cluster Manager: Red Hat Cluster Managerインストール・アドミニストレーション・ガイド”を参照してください。

1 インストール前の計画

このセクションでは、クラスタ・システムの構成に必要な要件について、その概要を説明します。以降のセクションでは、クラスタの各種構成要素のインストール、定義、および構成方法について説明します。クラスタ・システムのセットアップ・プロセスを計画するには、最初に構成をホット・スタンバイ構成またはアクティブ・アクティブ構成のどちらにするかを決定する必要があります。ホット・スタンバイ構成では、一度に 1 つのノードのみが Caché を実行します。

アクティブ・アクティブ構成では、各ノードが専用の Caché インスタンスを実行します。ノードが同じデータベースに直接アクセスすることではなく、各データベースは最初の Caché 構成またはもう一方の Caché 構成に割り当てられます。ユーザは ECP によって Caché インスタンスをネットワーク化し、ネームスペース定義によって両方のノードから同じデータを投影できます。

アクティブ・アクティブ構成では、以下の作業が必要です。

- ・ 各 Caché インスタンスに一意の IP アドレスを割り当てる。
- ・ 各 Caché インスタンスに一意の既定ポート番号を割り当てる。

- 各 Caché インスタンスに、インスタンスがインストールされデータベースが配置されている 1 つまたは複数のディスク・パーティションを割り当てる。Caché インスタンスはパーティションを共有できません。

以下の作業は、両方の構成タイプに必要です。

- 各種カーネル・パラメータの設定値を計算および更新し、Caché インストールのサポートに必要な変更を加える。また、両ノードが要件をサポートすることを確認する。アクティブ・アクティブ・クラスタを構成する場合は、1 つのノードが両方の Caché インスタンスを実行する場合に備えて、そのパラメータ値を両方に追加する必要があります。

詳細は、“Caché インストール・ガイド”の付録の“UNIX と Linux 用のシステム・パラメータの計算”を参照してください。

- アクティブな各 Caché インスタンスに仮想 IP アドレスを割り当てる。
- アクティブな各 Caché インスタンスにそれぞれ固有のパーティションが十分にあることを確認する。
- Red Hat Cluster Manager がインストール済みで、クラスタ・サービスが実行されていることを確認する (service cluster start コマンドの説明は、“Red Hat Cluster Manager: Red Hat Cluster Managerインストール・アドミニストレーション・ガイド”の“クラスタの管理”の章を参照)。
- Caché 構成名に Red Hat サービス名を選択する。これは必須ではありませんが、同じ名前を使用した方がクラスタ構成が簡単になります。

2 Caché に対するクラスタ・サービスの構成

クラスタ・システムを準備し、第 1 のクラスタ・ノードを構成するには、以下の手順を実行します。

- “Red Hat Cluster Manager: Red Hat Cluster Managerインストール・アドミニストレーション・ガイド”の“ディスクのパーティション設定”のセクションに説明されている手順に従って、fdisk ユーティリティを使用し、パーティションを作成します。
- 各クラスタ・ノードにパーティションのマウント・ポイントを作成します。
- 最初に作業を行うノードを選択します。
- Caché クラスタ・サービスを定義します。
- 最初のノードに Caché をインストールします。

2.1 Cache クラスターサービスの定義

cluadmin ユーティリティを使用して、Cache クラスターサービスを定義します。詳細は、“Red Hat Cluster Manager: Red Hat Cluster Managerインストール・アドミニストレーション・ガイド”の“cluadmin ユーティリティの使用”のセクションを参照してください。

1. Cache 用の仮想 IP アドレスとストレージを指定します (スターアップ・スクリプトではなく)。

各構成に優先ノードを割り当てる際は、relocate サブコマンドに no を指定します。yes を指定すると、サービスの起動時に、サービスが自動的に優先ノードに再配置されます。yes を指定しても Cache に問題は生じませんが、再配置プロセスを手動で制御できる no の方が推奨されます。

ノード間でサービスを移動するには、service relocate コマンドを使用します。詳細は、“Red Hat Cluster Manager: Red Hat Cluster Managerインストール・アドミニストレーション・ガイド”の“サービスの再配置”のセクションを参照してください。

service show config コマンドによる以下の出力は、1 つのディスク・パーティションが割り当てられた仮想 IP アドレスが 192.9.202.197 の cacheha1 という名前のサービスの構成を示しています。

```
cluadmin>service show config cacheha1

name: cacheha1                -->Name picked for this service
preferred node: lx4
relocate: no
user script: None              -->Did not specify this yet
monitor interval: 0           -->Specify 0 here
IP address 0: 192.9.202.197    -->Virtual IP address assigned
  netmask 0: 255.255.0.0
  broadcast 0: 192.9.255.255
device 0: /dev/sdc4            -->Partition name
  mount point, device 0: /storage2 -->Mount point for the partition
  mount fstype, device 0: ext3
  mount options, device 0: rw
  force unmount, device 0: yes
  samba share, device 0: None
```

2. 必要なすべてのサービスを定義し、ストレージを割り当てます。
3. service show state コマンドを使用して、サービスとそれらの現在の状態を一覧表示します。無効なサービスがある場合は、service enable コマンドを使用して有効化します。もう一方のノードで実行されるサービスがある場合は、service relocate コマンドを使用して、現在のノードに移動します。

詳細は、“Red Hat Cluster Manager: Red Hat Cluster Managerインストール・アドミニストレーション・ガイド”の“サービスの設定と管理”の章を参照してください。

2.2 Caché のインストール

第 1 ノードに Caché をインストールします。インストール・プロセスで構成名を尋ねられたときは、クラスタ・サービスに適用したサービス名を使用します。

インストールが完了したら、Caché 構成マネージャを使用して、以下の構成変更を行います。複数の Caché インスタンスをインストールする場合は、すべてのインストールの完了時ではなく、それぞれのインストールの完了後にこの手順を実行してください。

1. ポート番号の変更 – [詳細] タブの [一般] を展開します。[デフォルト・ポート番号] を、1972 から Caché のこのインスタンスに使用する一意のポート番号に変更します。
2. ライセンス・マネージャの更新 – [詳細] タブの [ライセンス] および [ライセンス・マネージャ] を展開します。

[マネージャ番号 1] の [名前/IP アドレス] を、現行ノードの名前 (または IP アドレス) で置換します。仮想 IP アドレスではなく、実 IP アドレス (または DNS 名) を使用してください。

第 2 のライセンス・マネージャを追加し ([ライセンス・マネージャ] をハイライト表示して [追加] をクリック)、もう一方のクラスタ・メンバの IP アドレスまたは DNS 名を入力します。ここでも、クラスタの仮想 IP アドレスではなく、実名または実 IP アドレスを使用してください。

3. ECP の構成 – アクティブ・アクティブ・クラスタをセットアップし、Caché インスタンス間で ECP を使用する場合は、この時点または以降の段階で ECP を構成できます。[ECP] タブの [エンタプライズ・キャッシュ・プロトコル・サーバとして稼働] チェック・ボックスにチェックを付けます。[追加] をクリックして、もう一方の Caché インスタンスをこのインスタンスに対するサーバとして定義します。明確にするために、ECP サーバの名前にはクラスタ構成名を選択します。この名前は、構成マネージャの [データベースの作成] ウィザードでリモート・ノードを参照する目的で使用されます。[ホスト名] には、実 IP アドレスや実 DNS 名ではなく、クラスタ構成に割り当てた仮想 IP アドレスを使用します。
4. ECP サーバおよびクライアントの最大数の増加 – ネットワークに他の ECP サーバやクライアントがある場合は、各自のシステムに合わせて最大数の設定を増やします。[詳細] タブの [ネットワーク] を展開します。[ECP サーバ] の [ECP クライアント最大数] および [ECP クライアント] の [ECPサーバの最大数] の設定値をそれぞれ増やします。必要に応じて、ECP アクセス制御を有効にすることもできます。

ECP の構成の詳細は、“Caché 分散データ管理ガイド” の “分散システムの構成” の章を参照してください。

現在実行中の Caché インスタンスを停止し、現行ノードで Caché のすべての追加インスタンスをインストールおよび構成する手順を繰り返し実行します。

第 1 のクラスタ・ノードの構成が完了しました。手順を続行する前に、シェル・プロンプトで `ccontrol list` または `ccontrol all` を実行して、第 2 ノードの構成に必要な情報を収集してください。

3 第 2 ノードの構成

cluadmin の service relocate コマンドを使用して、もう一方のクラスタ・ノードにサービス (定義したストレージ) を移動します。

次のどちらかの方法で、このノードの Caché インスタンスを定義します。

- インストール手順を再度実行して、もう一方のノードで使用した構成名と同じ名前を指定します。同じディレクトリにインストールします。

Caché で使用する Web サーバを構成する場合は、これが最善の方法です。Web サーバの構成は各ノードに対して “ローカル” であるためです。Caché を再インストールしても、構成ファイル (`cache.opf`) に対して行った変更は影響されません。

- もう一方のノードの Web サーバの構成が Caché で不要な場合は、`ccontrol install` コマンドを使用してローカル・ノードに Caché インスタンスを登録します。

```
ccontrol create $cfgname directory=$tgtdir versionid=$ver
```

例えば、第 1 ノードでの `ccontrol all` が、次の内容を出力したとします。

	Configuration	Version ID	Port	Directory
dn	CACHEHA1	5.0.1.543	1973	/store1/c50ha
db	CACHEHA2	5.0.1.543	1972	/store2/c50ha2

この場合は次を実行します。

```
ccontrol create cacheha1 directory="/store1/c50ha" versionid="5.0.1.543"
ccontrol create cacheha2 directory="/store1/c50ha" versionid="5.0.1.543"
```

`ccontrol start` を使用して構成を開始し、問題なく動作することを確認したら `ccontrol stop` で構成を停止します。

4 クラスタ・サービスへの Caché の追加

サービス・フェイルオーバーのプロセスとして Caché を開始および停止するスクリプトは、Caché に付属していません。メイン・スクリプトの例は、[Linux 用の Caché 初期化ファイル](#) のセクションで紹介されています。メインの初期化スクリプトを作成したら、以下の手順を実行して、クラスタ・サービスに Caché を追加します。

- インストール済みの各 Caché インスタンスの `/etc/rc.d/init.d` にスクリプトを作成します。つまり、定義済みの各 Caché クラスタ・サービスに対応するインスタンスごとにスクリプトを作成します。以下の 2 例のどちらかをモデルにして、スクリプトを作成します。

例 1 :

```
#!/bin/ksh
/etc/rc.d/init.d/cache $1 cachehal failover
exit ?$
```

例 2 :

```
#!/bin/ksh
/usr/local/etc/cachesys/cache-init $1 cachehal failover
exit ?$
```

cachehal は各自の構成名に置換します。スクリプト名は、“/etc/rc.d/init.d/cache-**<config>**” という形式になります (この例では、/etc/rc.d/init.d/cache-cachehal)。

2. cluadmin ユーティリティの service modify コマンドを使用して、各サービス用のスクリプトの場所を更新します。監視間隔の設定は 0 のままにします。

このスクリプトは status コマンドをサポートしますが、監視間隔が 0 以外に設定されているときに Caché が応答不能になると、もう一方のクラスタ・メンバに自動的にフェイルオーバーされ、問題の診断に必要な情報を収集できなくなります。

3. サービスの起動先とするノードにサービスを再配置します。そのノードがこの時点でアクティブな場合は (つまり、スクリプトを追加する前に、サービスがこのノード上のストレージの可用性を制御している場合は)、service disable および service enable コマンドを使用します。それ以外の場合は、service relocate コマンドを使用します。

service relocate および service disable コマンドは、Caché の正常な終了を試行する stop パラメータを使用してスクリプトを呼び出します。

この時点で、Caché はフェイルオーバー・クラスタ・サービスの構成要素となっています。以下の手順を実行して、クラスタをテストします。

1. 第 1 ノードの電源を切ります。
2. 停止ノードで実行されていた Caché バージョンが第 2 ノードで開始されていることを確認します。
3. 障害が発生したマシンをオンに戻します。Caché はそのまま第 2 ノードで実行されているはずですが (自動的にフェイルバックされることはありません)。
4. 第 2 ノードの電源を切ります。両方の Caché インスタンスが第 1 ノードに移行するはずですが。
5. 第 2 ノードをオンに戻します。
6. service relocate コマンドを使用して、一方の Caché インスタンスを第 2 ノードに戻します。
7. Caché 構成マネージャを、クラスタの仮想 IP アドレスを使用する 2 つの Caché インスタンスに接続します。タイトル・バーのノード名を調べてどちらのノードに接続したかを特定し、cache.cpf ファイルへのパス名を調べてそれがどちらの構成かを特定します。

4.1 Linux 用の Caché 初期化ファイル

このセクションでは、Linux Red Hat Advanced Server での Caché の開始および停止に使用するスクリプトの例を紹介します。このファイルを `/etc/rc.d/init.c/cache` または `/usr/local/etc/cachesys/cache-init` として保存し、保護を 755 に設定します。以下にスクリプト例を示します。

```
#!/bin/ksh
# cache
#
# Cache "System V init" script for Linux systems
#
# Copyright (c) 2003 by InterSystems.
# Cambridge, Massachusetts, U.S.A. All rights reserved.
# Confidential, unpublished property of InterSystems.
# -----
# Maintenance
# 04/01/2003 This script is born.
# -----
# This script is put in the init.d directory and is used by
# the HA failover package to start a Cache configuration when
# the node that was "serving" it failed.
#
# Three arguments should be specified:
# hacache start <config name> failover
# where <config name> is the name of the configuration that
# is displayed by "/usr/bin/ccontrol all" in the 2nd column.
#
# This script can be used to start Cache if Cache is currently
# down (meaning it is down on both nodes). However Cache must have
# been shut down cleanly, it cannot have crashed (eg. there must
# not be a cache.ids file in the cachesys/mgr directory).
# In the future when we are capable of detecting which node created
# the cache.ids file this script will be extended so that it can
# also restart Cache at boot time following a crash. At the moment
# if this script is called without the failover flag and the
# cache.ids file exists, it will display a message and refuse to start
# Cache. Do not use the failover flag to override this behaviour unless
# you know Cache is not running on the other node.
#
# It is very dangerous to call this script and specify the failover
# flag outside of the failover scripts. In an HA environment where
# multiple nodes can see the attached storage simultaneously (eg. NFS
# mounted file systems) it is possible to start Cache from the same
# directory on both nodes; Cache does not currently prevent this.
# If this occurs the results will be disastrous and both nodes will
# have to be shut down, database degradation may need to be repaired,
# and so on.
```

```

#
if [ "$2" = "" ]
then
type="xxxx" #invalid option, forces usage message
else
config=$2 #cache configuration to play with
state=$3 #failover or "nothing"
#
basdir=`/usr/bin/ccontrol list $config | grep -i directory | awk {'print $2'} `
localnode=`uname -a | awk '{print $2}' `
if [ "${basdir}" = "" ]
then
echo "Configuration $config not found"
exit 1
fi
type=$1
fi
#
#See how we were called.
case "$type" in
(start)
# Start daemons.
if [[((-e ${basdir}/mgr/cache.ids) &&("${state}" != "failover" ))]]
then
echo "${basdir}/mgr/cache.ids exists and startup is not failover"
echo "Cache configuration $config not started on $localnode"
exit 1
fi
echo "Starting Cache-HA config $config on $localnode"
ccontrol start $config quietly
status=$?
case $status in
(1)
echo "...Failed to start"
exit 1
;;
(0)
echo "...Started"
exit 0
esac
;;
(stop)
# Stop daemons.
echo "Stopping Cache-HA config $config on $localnode"
ccontrol stop $config quietly
status=$?
case $status in
(1)
echo "Cache configuration $config failed to stop"
exit 1
;;
(0)
echo "Cache configuration $config stopped"
exit 0
esac
;;
(status)
FIELDWIDTH=2
state=`/usr/bin/ccontrol all | grep -i $config | awk {'print $1'} `
if [ "$state" = "up" ]
then
exit 0 #cache is up
fi
exit 1 #cache is down or we can't tell
;;
(restart)
$0 stop $2 $3 || :
$0 start $2 $3

```

```
;;
(*)
echo "Usage: $0 {start|stop|status|restart} <config> [failover|null]"
exit 1
esac
exit 0
```

5 アップグレード時の Caché レジストリの保守

どちらかのクラスター・メンバ上で実行されている Caché によって、フェイルオーバー・クラスターの Caché をアップグレードできます。ただし、アップグレードを実行していないノードでは、Caché が保守するレジストリ (ccontrol all および ccontrol list で表示される) が正しいバージョン ID を表示しません。バージョン ID は ccontrol update コマンドを使用して手動で更新します。構文は、以下のとおりです。

```
ccontrol update $cfgname versionid=$ver
```

例えば、構成 cacheha1 の現在のバージョン ID を 5.0.1.579 に設定する場合、次のコマンドを実行します。

```
ccontrol update cacheha1 versionid="5.0.1.579"
```

