サーバの負荷分散と冗長化

松岡　孝

共通基盤技術支援室　情報通信技術系

概要

Linux に装備されている Linux Virtual Server（LVS）という技術を利用すると，通常は一台のサーバで処理する負荷を，複数のサーバを用いて分散処理させる事が可能である．さらに LVS を行うサーバを冗長化することによって，より可用性の高いシステムを構築することができる．

本稿では，これらを実現するためのミドルウェアである KeepAlived を用いたサーバの構築と実装について紹介する．

1　負荷分散とは

負荷分散とは，特定のサーバに処理が集中しないよう，他のサーバになるべく均一に処理を分散させることである．この技術を使うと，利用者からの要求に素早く応答できたり，分散先のサーバがたとえ一台停止しようとも，残りのサーバでサービスの継続が可能である

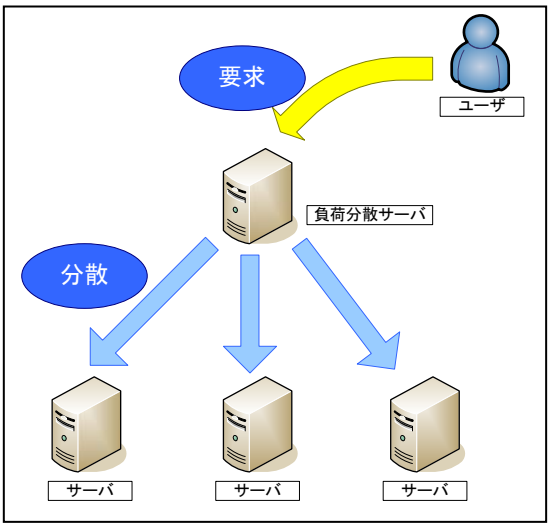

図 1. 負荷分散の例

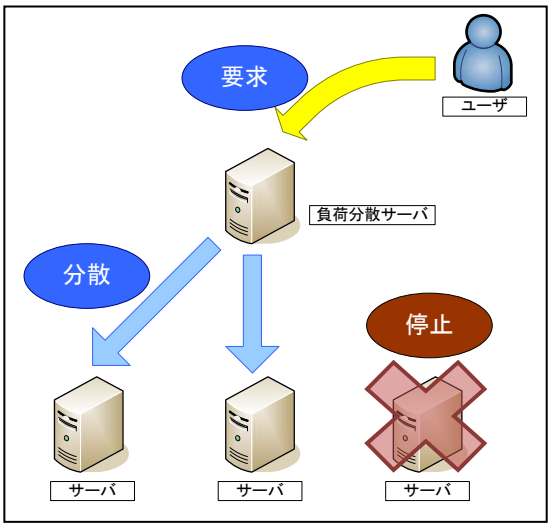

図 2.サーバ一部停止時

2　サーバ冗長化とは

サーバ冗長化とは，サーバの障害による停止に備えて，障害発生後もサービスが継続できるように，サーバを複数配置して運用する構成のことである．サーバを冗長化するとシステムの停止が発生しないため，通常の一台構成のサーバよりサービス停止率が著しく下がり，またメンテナンス性も向上する．

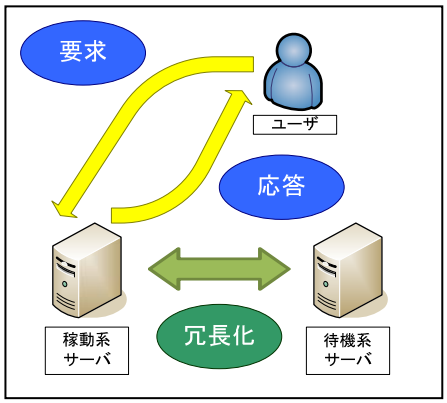

図 3. サーバ冗長化の例

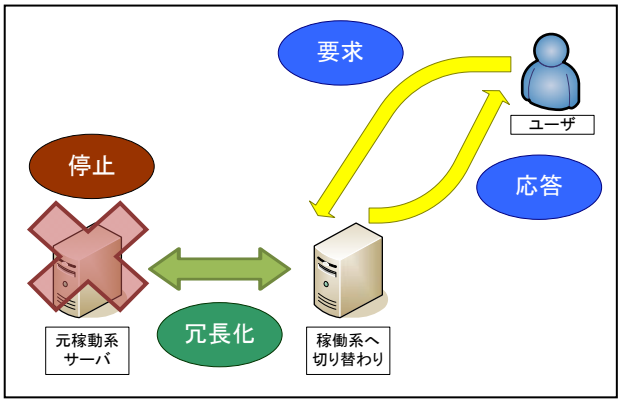

図 4. 冗長化後の障害発生時

3 KeepAlived について

Linux には，元々Linux Virtual Server（LVS）という負荷分散のための機能が備わっている．KeepAlived はこの LVS の制御を行うミドルウェアである．KeepAlived は，LVS だけでは実現できないサーバ冗長化の機能も備えており，負荷分散を行うサーバも二重化することが可能である．

4 KeepAlived の実装

4.1 KeepAlived によるサーバ冗長化

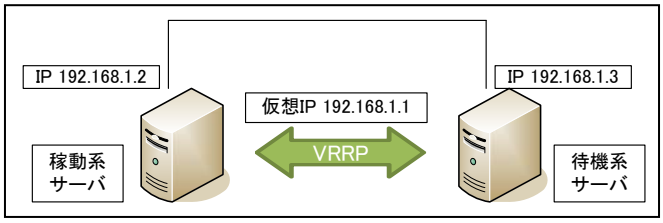

図 5. KeepAlived による冗長化例

あるサーバを KeepAlived で冗長化する例を紹介する．冗長化機能を有効にすると，サーバ間で VRRP による死活監視が行われ，両者間では仮想 IP が利用できるようになる．利用者がサーバへアクセスする際は，この仮想 IP を宛先にすると，そのタイミングで稼動系として動作しているサーバへ到達できる．万が一，稼動系のサーバが停止したとしても，待機系は稼動系へと昇格し仮想 IP も引継がれるため，利用者にサーバ停止を意識させることがない．

4.2 KeepAlived による負荷分散

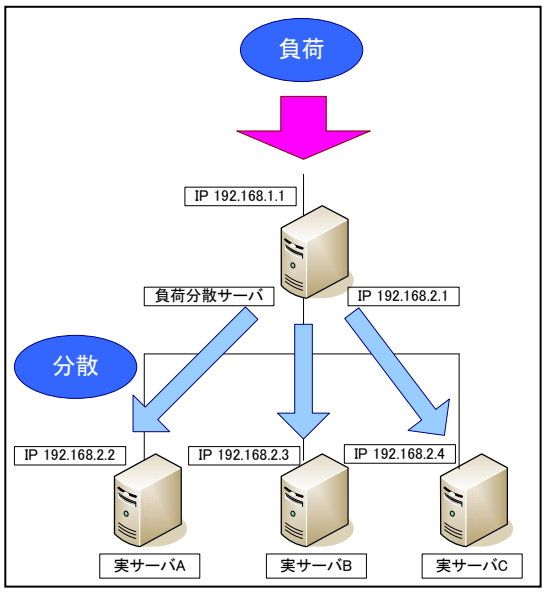

図 6. KeepAlived による負荷分散例

KeepAlived の負荷分散機能を利用すると，KeepAlived が動作しているサーバで受けた負荷を，他の実サーバへ振り分けることができる．なお実サーバ側ではゲートウェイの指定を負荷分散サーバにする必要がある．これにより分散された処理の戻り先が負荷分散サーバになり，この負荷分散サーバから利用者に要求の応答を返す．

また KeepAlived はヘルスチェック機能も備えており，表 1

表 1. ヘルスチェックの方式

TCP_CHECK	実サーバの特定の TCP ポートの応答を確認する
HTTP_GET / SSL_GET	実サーバに HTTP か HTTPS で接続しファイルを取得する
SMTP_CHECK	実サーバに SMTP 接続しコマンドの応答を確認する
MISC_CHECK	別途スクリプトを実行し応答を確認する

に記す方式で，実サーバの状態監視が行える．

ヘルスチェック機能により無応答と判断された実サーバは，自動的に負荷の分散先から切り離され，復旧した際には，再び負荷の分散先へと自動的に追加される．これにより，利用者からは実サーバの停止と復旧を意識させることがない．なお表 2 で記すように負荷分散の方式には各種あり，用途によって適宜使い分けることができる．

表 2. 負荷分散の方式

方式	意味	動作
rr	Round Robin ラウンドロビン	実サーバに均等に分散する．
wrr	Weighted Round Robin 重み付けに応じたラウンドロビン	実サーバに設定された重み付け順に分散する．
lc	Least Connection 最小接続数	接続数が最も少ない実サーバに分散する．
wlc	Weighted Least Connection 重み付けに応じた最小接続数	重み付けを考慮した上で接続数が最も少ない実サーバに分散する．
lblc	Locality Based Least Connection 宛先に応じた最小接続数	通常はプロキシサービスに用いる．同一の宛先へは同一の実サーバに分散する．ただし実サーバへの接続数が多い場合は他の実サーバへ分散する．
lblcr	Locality Based Least Connection with Replication レプリケーション付きの lblc	lblc と同じ動作をし，他の実サーバへ分載する際もセッションのマッピングを維持する．
dh	Destination Hashing 宛先ハッシュ	同一の宛先へは同一の実サーバに分散する．
sh	Source Hashing 送信元ハッシュ	同一の送信元からは同一の実サーバに分散する．
sed	Shortest Expected Delay 応答速度の予測	最も応答が速いと予測される実サーバに分散する．
nq	Never Queue アイドルサーバを優先	アイドルの実サーバがあれば，そこへ分散する．ない場合は sed と同じ動作をする．

4.3　負荷分散と冗長化を組み合わせる

負荷分散の機能を利用すると，たとえ一台の実サーバが停止してもサービス停止には至らない．しかしこの負荷分散サーバがシングルポイントとして存在し続ける．もし負荷分散サーバが停止した場合にはサービス停止へと至ってしまう．これを防ぐために負荷分散サーバを冗長化する例を紹介する．

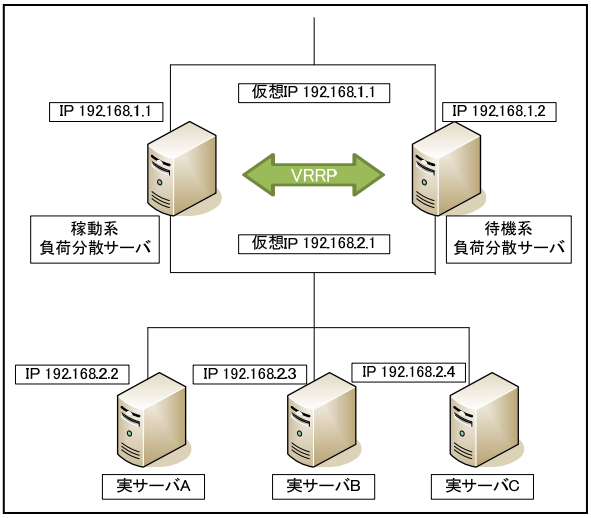

図 7. 負荷分散と冗長化の組合せ例

図 7 がその例である．負荷分散サーバを冗長化することにより，懸念であったシングルポイントが無くなっている．この構成では，負荷分散サーバの間でネットワークインターフェイスの上位側と下位側に 2 つの仮想 IP を持つことになる．利用者はサーバへアクセスする際，上位側の仮想 IP192.168.1.1 を宛先にすることで，稼動系として動作しているサーバへ到達できる．また実サーバ側のゲートウェイを下位側の仮想 IP192.168.2.1 に指定することで，たとえ稼動系の負荷分散サーバが停止したとしても，待機系にゲートウェイの仮想 IP が引継がれ，分散された処理の戻り先が確保できる．この組合せにより稼動系の負荷分散サーバが停止したとしてもサービスの停止には至らない．

5　おわりに

本稿ではサーバの負荷分散と冗長化の実装例を紹介した. Linux には元々LVS という負荷分散機能は搭載されているが，設定項目が複雑で，さらにヘルスチェックや冗長化の機能は含まれていない．KeepAlived というミドルウェアを使用すれば，比較的容易に負荷分散と冗長化を組み合わせたサーバの構築が可能になる．Web サーバ等の並列処理が可能なサービスについては，特にこの KeepAlived を利用した構成を導入しやすいので，ぜひ参考にしてもらいたい.

最後に日頃，何かとお世話なっている情報連携統括本部の構成員ならびに，全学技術センター情報通信系の諸氏に感謝する.

参考文献

[1] デージーネット, “Linux アドバンストネットワークサーバ構築ガイド HA サーバ構築編”, 秀和システム, 2005 年 12 月

[2] “Linux 仮想サーバー - Red Hat Customer Portal “, （https://access.redhat.com/knowledge/docs/ja-JP/Red_Hat_Enterprise_Linux/5/html/Cluster_Suite_Overview/s1-lvs-overview-CSO.html）