

Edge V7 概要

Agenda

- Edge Componentsの概要とロードマップ
- Load Balancerの実装
- ULBとLLBの差異
- ULBとLLBの選択

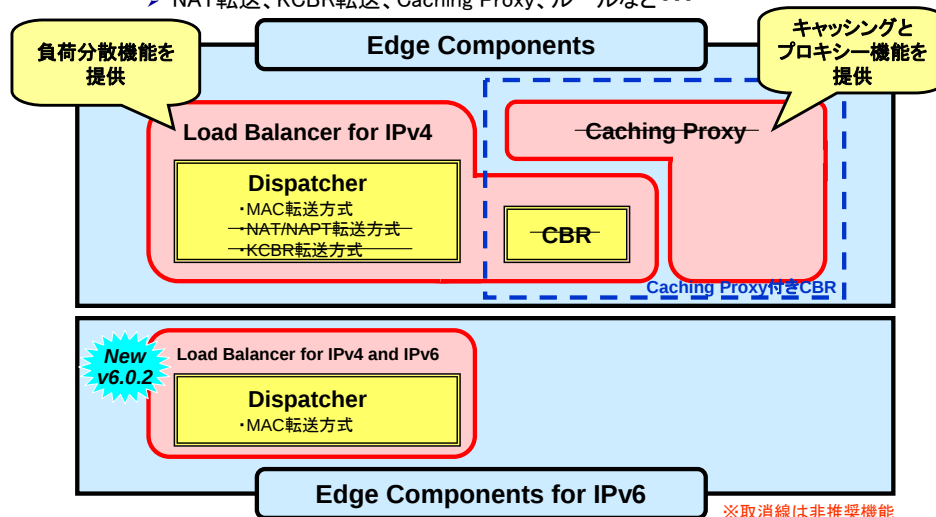
Edge Componentsの概要とロードマップ

Edge Components の概要と非推奨機能

■ Edge Componentsのソフトウェア構成

◆ WAS V6.1より多くの機能が非推奨(廃止予定)に

➢ NAT転送、KCBR転送、Caching Proxy、ルールなど...



4

Edge Componentsは、WAS NDが提供するソフトウェア負荷分散サーバーです。V7のEdge Componentsには、以下の3つのコンポーネントが含まれます。

- Load Balancer for IPv4
- Load Balancer for IPv4 and IPv6
- Caching Proxy

Load Balancerは負荷分散機能を提供するソフトウェアであり、Caching Proxyはキャッシング機能をもつプロキシ・サーバー・ソフトウェアになります。詳しく見ると、Load BalancerにはDispatcherとCBRなどのサブコンポーネントがあります。CBRはCaching Proxyのプラグインとして動作し、Caching Proxyに負荷分散機能を提供します。

WAS V6.1では、NAT転送やCBR転送、Caching Proxyといったいくつかの主要な機能が非推奨(廃止予定)となりました。図中の取消線が非推奨機能に該当します。非推奨とは廃止予定を意味しており、IBMの基本的な方針としては、非推奨になったリリースから数えて、次の2メジャー・リリースまたは満3年(の、いずれか長い方)は、その機能を製品から除去しません。しかし、例外的にこの方針より早く削除されるケースもあります。

V6.0.2からは、Load Balancer for IPv4 and IPv6が登場しました。これは、IPv6に対応した新しいLoad Balancerであり、IPv4でも使用可能です。詳細は後述しますが、従来のLoad Balancerとは異なりユーザー・スペースでの実装となっていることから、ULB (User Space Load Balancer)と呼ばれることもあります。また、単にIPv6版とよばれることもあります。

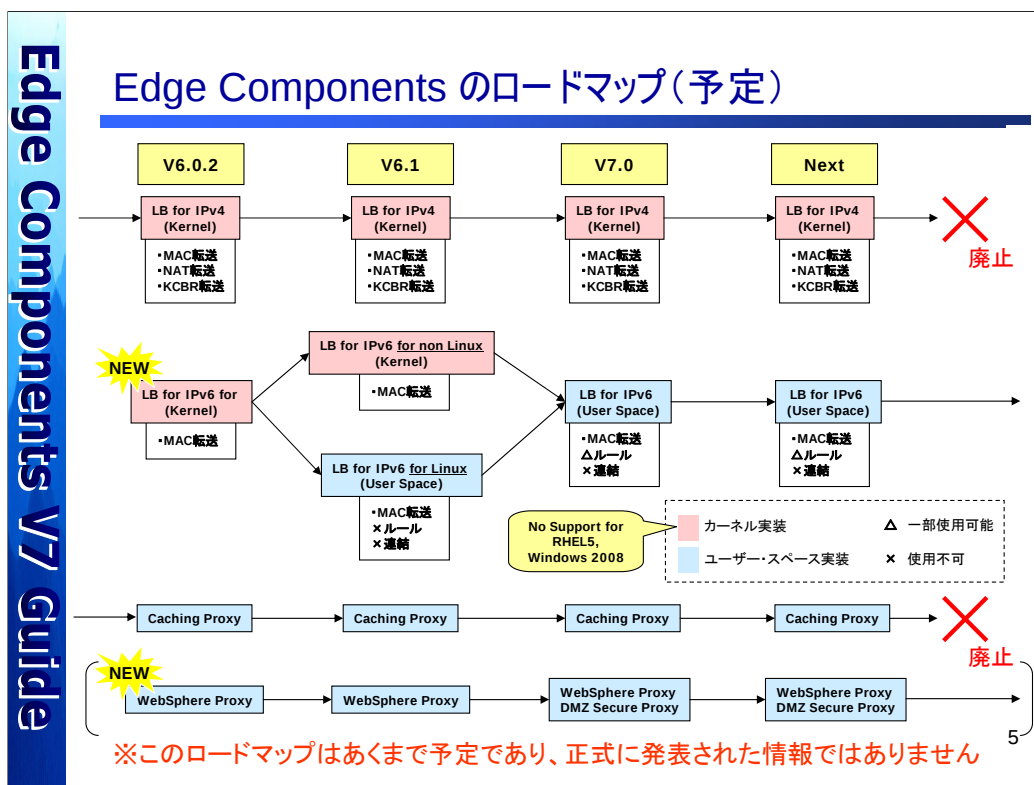
Load Balancer for IPv4およびCaching Proxyは将来のリリースで廃止となるため、今後はLoad Balancer for IPv4 and IPv6のMAC転送を使用した構成への移行が推奨となります。

非推奨のフィーチャー

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/rmig_depfeat.html

Edge Components V6.1での非推奨機能のお知らせとマイグレーション・アクション

Notes:///492563D1002D25CF/589C1169D7E304E349256F57001DCE59/FB83CF70A62CE132492571DC00158965



ここでは、Edge Componentsのこれまでのリリースの製品の構成と、将来の予定をまとめました。

カーネル実装の従来のLoad Balancerは今後廃止となりますので、Load Balancer for IPv4 and IPv6への移行が推奨となります。

Caching Proxyも同様に廃止となりますので、他の製品への移行やトポロジー変更が必要となります。Caching Proxyの代替として使用できるIBMの製品としては、WASで利用可能なWebSphereプロキシー・サーバーがあります。また、mod_proxyを使用してIHSをプロキシー・サーバーとして構成する事も可能です。

プロキシー・サーバーのセットアップ

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tjpx_setups.html

Apache モジュール mod_proxy

http://httpd.apache.org/docs/2.2/ja/mod/mod_proxy.html

WebSphere Proxy Server とは

■ WebSphere Proxy Serverとは

- ◆ Javaベースのプロキシ
- ◆ WAS管理コンソール上で管理が可能
- ◆ 基本機能
 - WLM
 - Cache機能
 - SSL通信・復号機能
 - Webサーバー・プラグインとの連携
 - セッション・アフィニティー



■ DMZ Secure Proxy Serverとは

- ◆ DMZ(非武装地帯)へ配置するために、セキュリティを強化したProxy Server
- ◆ 専用のインストーラーと専用のプロファイル (Secure Proxyプロファイル)
- ◆ JDKではなくJREを使用
- ◆ Webコンテナなど、不要なコンポーネントは含まない
- ◆ セキュリティーレベルが設定可能

6

WebSphere Proxy Serverとは、WAS ND V6.0.2から登場したJavaベースの新しいプロキシ・サーバーです。Caching Proxyとは異なり、WASの管理コンソール上での管理が可能です。

Proxy Serverの基本機能は、WLM、Cache機能、SSL通信・復号機能、Webサーバー・プラグインとの連携、セッション・アフィニティーです。

•WLM

ODC(オンデマンド構成)と呼ばれる機能により、同一セル内のサーバーに対して追加構成をすることなくプロキシを構成できます。特定のアプリケーション・サーバーやWebサーバーに対して割り振る場合は、汎用サーバー・クラスターを構成して下さい。

•Cache機能

静的コンテンツ、動的コンテンツ、SSLコンテンツのキャッシュが可能です。

•SSL通信・復号機能

SSLの復号が可能です。クライアント→Proxy間をSSL通信、Proxy Server以降は通常のHTTP通信を行うことも、クライアント→Proxy ServerをSSL通信にし、Proxy ServerでSSLを復号してURLのチェックを行った後、再び暗号化を行い、後方へSSL通信を行うことも可能です。

•Webサーバー・プラグインとの連携

Webサーバー・プラグインから要求をProxy Serverに割り振ることが可能です。

•セッション・アフィニティーの維持

Proxy ServerにてURIやCookieによるアフィニティー制御が可能です。

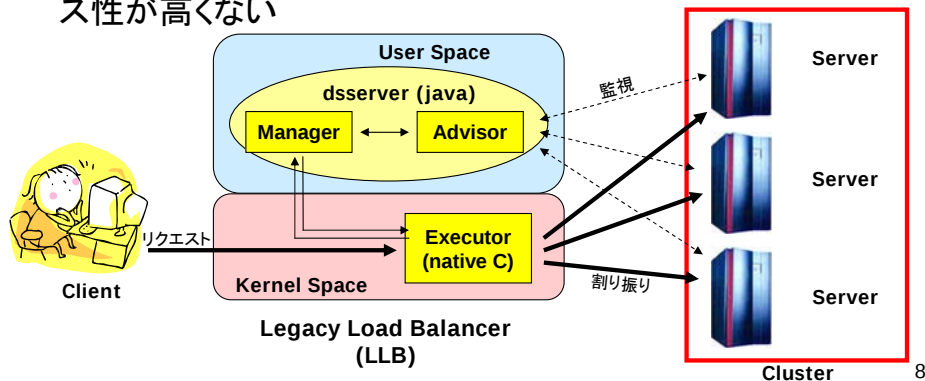
Secure Proxy Serverとは、WASV7から登場したDMZ(非武装地帯)へ配置するためにセキュリティを強化したProxy Serverです。具体的には、必要最低限のポートの開放、デジタル署名されたJARのみをロード、well knownポートへは非特権ユーザーとして接続等によりセキュリティが強化されています。

また、Proxy ServerはWASV7.0 NDに同梱されていますが、Secure Proxy Serverは「Secure Java Proxy Server installation image」に含められます。そして、Secure Proxy Serverを導入しても管理コンソールが導入されませんので、設定変更するためにはwsadminコマンド(ローカル、Flexible Management経由)もしくは、別ノードでConfiguration-Onlyプロファイルを作成し、そのノード上の管理コンソールから設定したProxyの構成情報を、wsadminコマンドにてimport/exportする必要があります。

Load Balancerの実装

Load Balancer for IPv4 (LLB)の実装

- カーネル・スペースで負荷分散処理を行う
 - ◆ 従来からのLoad Balancerということで、Legacy Load Balancer (LLB) と呼ばれることがある
- カーネルがバージョンアップするたびに新しいカーネルに対応したバージョンをFixとして適用する必要がある、メンテナンス性が高い

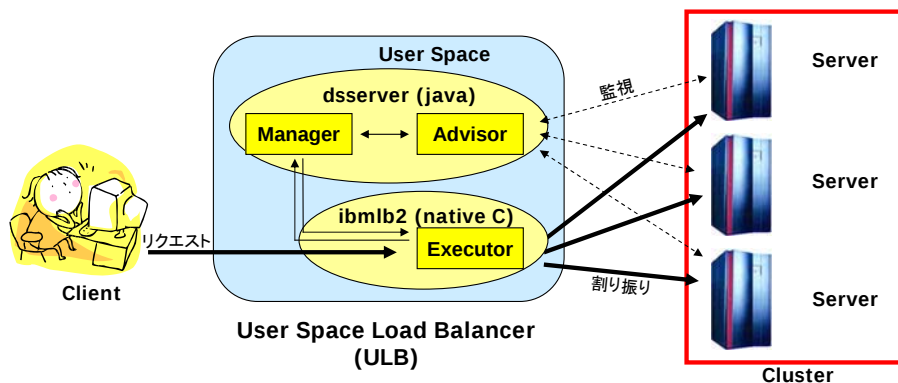


新しいLoad BalancerであるLoad Balancer for IPv4 and IPv6について説明する前に、従来のLoad BalancerであるLoad Balancer for IPv4の実装について説明します。従来のLoad Balancerでは、Executorはカーネル・スペースで動作します。カーネル上で動作しているため、Executorはプロセスとして確認することができません。カーネル上で動作することにより、リクエストを高速に処理することができましたが、この方法では、カーネルがバージョンアップするたびに新しいカーネルに対応したバージョンをFixとして適用する必要がある、メンテナンス性が低下するという問題がありました。

ManagerとAdvisorについては、dsserverという一つのjavaプロセス上で動作します。管理者はdscontrolコマンドまたはGUIを使用し、このdsserverに対して操作を行います。

Load Balancer for IPv4 and IPv6 (ULB)の実装

- ユーザー・スペースで負荷分散処理を行う実装
 - ◆ User Space Load Balancer (ULB)と呼ばれることがある
 - V6.1ではLinux上で動作するIPv4-IPv6版がULB
 - V7.0ではすべてのプラットフォームのIPv4-IPv6版がULB
- カーネル・レベルに依存しないため、メンテナンス性が向上



V6.1のLinux版およびV7.0のすべてのプラットフォームのLoad Balancer for IPv4 and IPv6では、Executorがユーザー・スペースで動作するようになりました。これにより、Executorがカーネル・レベルに依存することがなくなり、メンテナンス性が向上しました。

Executorはibmlb2という一つのプロセスとして確認することができます。ManagerとAdvisorについては従来のLoad Balancerと同様にdsserverというjavaプロセス上で動作します。新しい実装のLoad BalancerをUser Space Load Balancer (ULB)、従来のLoad BalancerをLegacy Load Balancer (LLB) またはKernel Load Balancerと呼ぶことがあります。

より正確には、V7.0のAIX版では、まだExecutorの一部がカーネルで動作しており、完全にユーザー・スペースで稼働しているわけではありません。

ULB と LLB の差異

Edge V7 の ULB と LLB の主な差異

		LLB	ULB
①	転送方式	MAC、NAT、KCBR、CP+CBR	MAC転送、カプセル化転送
②	区切り文字	:	@
③	アフィニティー方式	IP Sticky、Cookie	IP Sticky
④	ルール	10種類	2種類
⑤	dscontrol server downコマンド	使用可能	使用不可
⑥	同一筐体内への転送	可能	不可
⑦	テイクオーバー時の接続/アフィニティー情報の引き継ぎ	引き継ぎ可能	引き継ぎ不可
⑧	goActiveスクリプトによるクラスター・アドレスの設定	必須	自動
⑨	serverDownスクリプトの挙動	従来通り	変更
⑩	インストールディレクトリ	<EDGE_ROOT>/lb	<EDGE_ROOT>/ulb
⑪	ワイルドカード・クラスター/ポート	使用可能	使用不可
⑫	ICMPの転送	しない	する
⑬	TCPリセットの送信	可能	不可

ULBとLLBでは実装が異なるだけでなく、機能面でもいくつかの違いがあります。ここでは主な違いを表にまとめました。以降は、これらの違いを詳しく見ていきます。

なお、LLBのV5→V6→V7の差異については、以下のリンク先をご参照ください。

Edge ComponentsのV5.1からV6.0/6.1/7.0へのバージョンアップに関する既知の考慮点 (WAS-08-040)

<http://www-06.ibm.com/jp/domino01/mkt/cnpages1.nsf/page/default-0004BF4D>

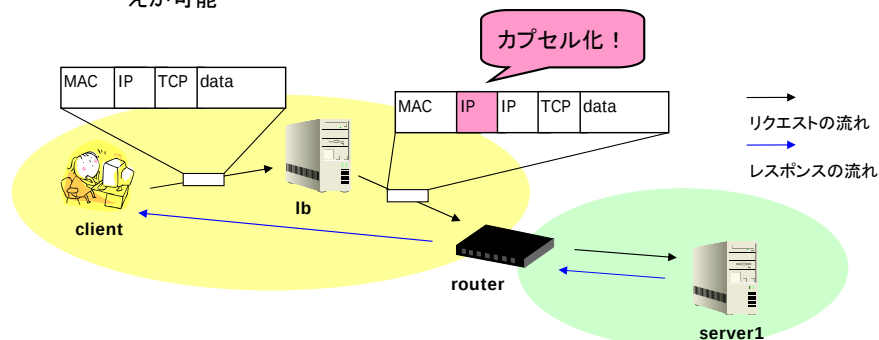
① 転送方式

■ LLB

- ◆ MAC、NAT、KCBR、CP+CBR転送が使用可能

■ ULB

- ◆ MAC転送のみが使用可能
- ◆ カプセル化転送が使用可能
 - IPパケットをIPIPまたはGREでカプセル化することで、MAC転送でのセグメント越えが可能



12

LLBでは、MAC転送、NAT転送、KCBR転送が使用可能です。またCaching ProxyとCBRを組み合わせることでCP+CBR転送を行うことができます。NAT転送ではセグメントを越えた割り振りが可能です。KCBR転送やCP+CBR転送ではURLやCookieをみての割り振りが可能です。

ULBでは、使用可能な転送方式はMAC転送のみとなります。MAC転送では、Load Balancerと割り振り先サーバーは同じネットワーク・セグメントに配置する必要があります。

ULBでは実装がカーネルからユーザー・スペースになったことで、OSレベルのトンネリングを利用できるようになりました。これを利用したのがカプセル化転送です。カプセル化転送ではIPパケットをIP in IPまたはGREといったトンネリング・プロトコルでカプセル化することで、MAC転送でもセグメントを越えた割り振りを可能になりました。

V6.1ではLBマシンと割り振り先サーバーの両方でIPTunnellingの定義が必要でしたが、V7.0ではLB側はdscontrolコマンドで設定できるようになりました。

なお、割り振り先サーバーがWindowsの場合、OSがIP in IPやGREをサポートしていないため、カプセル化転送を行うことはできません。

カプセル化転送を使用したネットワーク・セグメント間のトラフィックの転送

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.edge.doc/lb/info/ae/tprf_encapsulation.html

② 区切り文字

■ LLB

- ◆ クラスター、ポート、サーバーの区切りにはコロン「:」を使う

```
dscontrol server add cluster:port:server
```

■ ULB

- ◆ IPv6アドレッシングでは、アドレス表記に「:」が使われるため、区切り文字としては「:」のかわりに「@」を使う

```
dscontrol server add cluster@port@server
```

```
[IPv6アドレスの例] 3ae3:90a0:bd05:01d2:288a:1fc0:0001:10ee
```

13

LLBではdscontrolコマンドでクラスター、ポート、サーバーの区切りには「:」が使われていました。

ULBではIPv6に対応したため、IPv6アドレス表記で使用する「:」ではなく、「@」が区切り文字として使用されます。

③ アフィニティー方式

■ LLB

◆ IP Sticky

- すべての転送方式で使用可能
 - stickytimeを非ゼロ値(デフォルト:ゼロ)にし、stickyオプションをyes(デフォルト:yes)に設定する

```
dscontrol port add cluster1:80 stickytime 1800
dscontrol server add cluster1:80:server1 sticky yes
```

- 類縁性アドレスマスク(stickymask)が使用可能
 - stickytimeを非ゼロ値(デフォルト:ゼロ)にし、stickymaskを設定(デフォルト:32、設定可能な値:8,16,24,32)する

◆ Cookie Affinity

- KCBR転送方式ではPassive Cookieが使用可能
- CP+CBBR転送方式ではPassive/Active Cookieが使用可能

■ ULB

◆ IP Stickyのみ

- dscontrol portコマンドのオプションselectionalgorithmで設定
 - connection
 - ラウンドロビンでの割り振り
 - affinity
 - クライアントのIPに基づくアフィニティが有効となり、アフィニティのタイムアウトはstaletimeoutと同一になる
 - conn+affin
 - クライアントのIPに基づくアフィニティが有効となり、タイムアウトをstickytimeで設定する

```
dscontrol port add cluster1@80 selectionalgorithm conn+affin stickytime 1800
dscontrol server add cluster1@80:server1
```

- 類縁性アドレスマスク(stickymask)は使用不可能

14

負荷分散装置では、同じクライアントからのリクエストを同じサーバーに割り振るために、セッション・アフィニティーの機能が必要とされる場合があります。

LLBではセッション・アフィニティーを実現する機能として、全ての転送方式でIP Stickyが使用可能です。IP Stickyを有効にすると、LBがクライアントのIPアドレス毎にどのサーバーに割り振ったかを記憶し、Sticky有効時間内は同じサーバーへ割り振りを行います。LLBでIP Stickyを有効にするには、デフォルトがゼロであるstickytimeを非ゼロ値に設定します。stickytimeはexecutor、cluster、portの3つのレベルで設定可能であり、executor<cluster<portの順で優先順位が高くなります。dscontrol serverコマンドのstickyオプションを使用して、クラスター内の特定のサーバーだけIP Stickyを無効にすることも可能です。また、LLBでは共通サブネット・アドレスを基に、クライアントをグループ化してStickyさせるための機能として、類縁性アドレス・マスク(stickymask)が使用可能です。

また、LLBでは、KCBR転送やCP+CBBR転送方式ではCookieを使用することも可能です。

ULBではIP Stickyのみがアフィニティー機能として使用することができます。IP Stickyの設定はportコマンドのselectionalgorithmオプションで設定します。このオプションはポートを定義する際(すなわちdscontrol port addコマンド)のみ指定可能であり、dscontrol port setコマンドで変更することはできません。デフォルトのconnectionではIP Stickyは有効とならず、affinityまたはconn+affinを設定することで、IP Stickyが有効になりますが、これら2つではタイムアウトの扱いが異なります。

connectionでは、タイムアウトはstaletimeoutと同一となります。Load Balancerはstaletimeoutに設定された時間の間パケットのやりとりがない接続を接続情報テーブルから除去し、その接続の後続のパケットは破棄します。LLBでのstaletimeoutのデフォルトは300秒でしたが、ULBでは6400秒に変更されています。

一方conn+affinでは、LLBと同様にstaletimeoutとstickytimeを別々に設定することができます。またdscontrol serverコマンドのstickyオプションを使用することができるのはconn+affinを設定したときのみとなります。

ULBではdscontrol serverコマンドのstickyオプションは、7.0.0.12以降で使用可能です。

crossportアフィニティーはULBでもLLBと同様に使用可能です。

ULBでは類縁性アドレスマスク(stickymask)は使用不可能です。

④ ルール

■ LLB

- ◆ ip、time、active、true、contentなど、10種類のルール・タイプが使用可能

■ ULB

- ◆ 使用可能なルール・タイプはtrueとactiveのみ
 - timeに関してはdscontrol manager quiesceのdailyオプションで代替可能な場合も
 - 23時から5時59分までの割り振りを停止したい場合は以下の用に指定

```
dscontrol manager quiesce server1 daily 23 5
```

15

Load Balancerでは、デフォルトでは重み付きラウンド・ロビン方式での割り振りが行われますが、ルール機能を使用して、割り振りを制御することが可能です。

LLBではip、time、connection、active、port、service、reservedbandwidth、sharedbandwidth、true、contentの10種類のルール・タイプが使用可能です。ただし、これらのうち実際に使用されているのは、ip、time、active、true、contentの5種類であり、それ以外のルール・タイプが使用されている事例はほとんどありません。

type ipでは、クライアントのIPアドレスに基づく振り分けルールを定義することができます。type timeでは時間ベースの割り振りが可能であり、例えば夜間に割り振りを停止するといった設定が可能です。type activeでは活動中の接続数に基づく割り振りが可能であり、例えばアクセスが増えた場合にSorryサーバーへの割り振りを行うといったことができます。type trueはどんな条件にもマッチするルールであり、通常時に振り分けを片寄せする場合などに使用します。type contentはKCBR転送またはCP+CBR転送方式でのみ使用可能であり、HTTPリクエストのURLやCookieに基づいた割り振りを行うことができます。

ULBで使用可能なルール・タイプはtrueとactiveのみとなりました。

一方でULBではdscontrol manager quiesceコマンドに新しくdailyオプションが使用できるようになりましたので、たとえば夜間に振り分けを停止したい場合など、従来のtype timeの代替として使用できる場合もあります。

⑤ dscontrol server downコマンド

■ LLB

- ◆ dscontrol server downコマンドを利用して、特定のサーバーへの割り振りを即時停止することが可能
 - dscontrol manager quiesceコマンドを使用して、特定のサーバーをGracefullに割り振り停止することも可能

■ ULB

- ◆ dscontrol server downコマンドが削除された
 - 代わりにdscontrol manager quiesceコマンドを使用する
 - dscontrol manager quiesceからはnowオプションがなくなったが、nowを指定しなくても、アフィニティーが有効でも新規のリクエストの割り振りは行われない

16

LLBでは、dscontrol server downコマンドを使用して、特定のサーバーを明示的にダウン状態にすることができました。ダウン状態に設定されたサーバーは即座に割り振りが停止されます。よく似たコマンドとしてdscontrol manager quiesceコマンドがあり、こちらでは活動中の接続はそのままに、新規の割り振りのみを停止することができます。LLBのdscontrol manager quiesceコマンドではIP Stickyが有効なリクエストはそのまま割り振りが継続します。IP Stickyが有効な場合も新規リクエストの割り振りを停止する場合、nowオプションを使用します。

ULBでは、dscontrol server downコマンドが削除されました。代替としてはdscontrol manager quiesceコマンドを使用してください。ULBではdscontrol manager quiesceにnowオプションがなく、IP Stickyが有効な場合でも新規リクエストの割り振りは行われません。

⑥ 同一筐体内への割り振り(collocation)

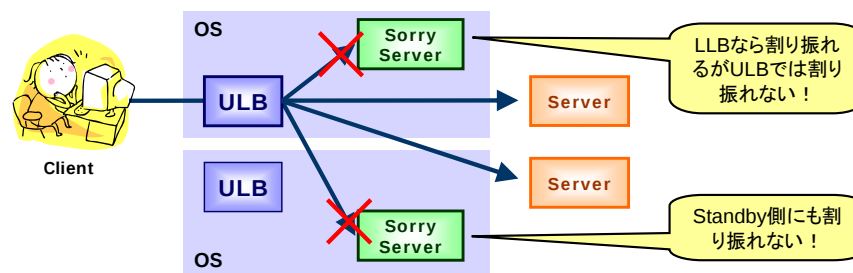
要注意

■ LLB

- ◆ LBと同一筐体内に割り振り先サーバーを配置する構成が可能
 - 例えば、SorryサーバーをLBと同一筐体内に配置することが可能

■ ULB

- ◆ 同一筐体内への割り振りができない
 - Sorryサーバーが必要な場合、別筐体の専用のSorryサーバーが必要
 - Standby機にSorryサーバーを配置することも不可能
 - 通常時の割り振り先サーバー内にSorryサーバーを定義することも不可能



17

LBと同一筐体内に割り振り先サーバーを構成する事をcollocationといいます。日本語マニュアルでは「連結」と訳されています。ここで、同一筐体とは同一OSを指します。

LLBでは同一筐体内に割り振りを行うことが可能であり、SorryサーバーをLBと同一筐体内に構成するといったことが可能でした。

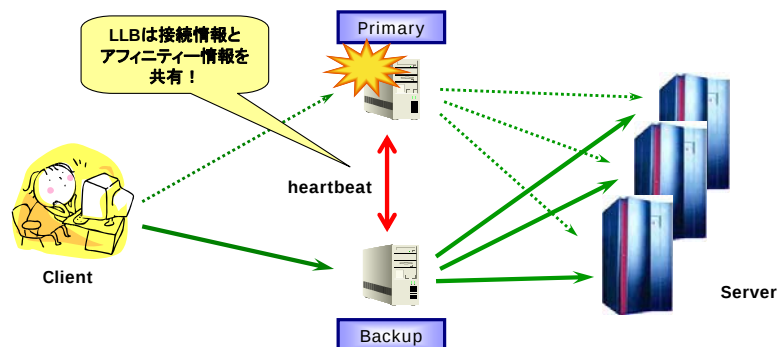
ULBではソフトウェアの実装上、この同居構成ができなくなりました。そのため、Sorryサーバーが必要な場合、必ず、Sorryサーバー専用のマシンを別に用意する必要があります。アクティブ側のLBから、スタンバイ側のLBに同居するSorryサーバーに割り振るということもできません。

また、通常時の割り振り先サーバーと同一筐体上にSorryサーバーを構成するということも、MAC転送の制約からできませんのでご注意ください。これは、MAC転送では割り振り先サーバーは必ずクラスター・アドレスをListenする必要があり、一つの筐体上で同じクラスター・アドレスをListenして複数のWebサーバー・プロセスを起動することができないためです。

⑦ テイクオーバー時の接続/アフィニティー情報の引き継ぎ

要注意

- LLB
 - ◆ ハートビートで、接続情報とアフィニティー情報を共有
- ULB
 - ◆ 接続情報を共有しない
 - ◆ アフィニティー情報を共有しない



18

LLBでは、ハートビートを通して、HAパートナー間で接続情報とアフィニティー情報を共有しています。接続情報を共有しているため、テイクオーバーが発生しても、発生時にアクティブであった接続をバックアップ機が引き継ぐことができ、クライアントからは障害が発生したことがわかりません。また、アフィニティー情報を共有しているため、テイクオーバーのあともセッション・アフィニティーが正しく行われます。ただし、ハートビートが0.5秒間隔であるために、この短い時間に更新された一部の情報が引き継がれない場合もあります。

ULBでは、接続情報およびアフィニティー情報を共有していません。接続情報を共有していないため、テイクオーバーが発生すると、接続中だったクライアントからの以降の packets がバックアップ機では処理できず無視されるため、結果クライアントではタイムアウトエラーが発生します。また、アフィニティー情報を共有していないため、テイクオーバーが発生すると、バックアップ機は、リクエストをセッション・オブジェクトがないサーバーに割り振ってしまう可能性があり、そのため、クライアントはログイン画面からやり直すことになってしまう場合があります。

そのため、確実なセッション・アフィニティーが必要な場合、ULBではIHS:WASを1:Nで構成してLBではなくIHSがアフィニティーを行う構成にするか、WAS側でセッション・パーシスタンスを構成する必要があります。

⑧ クラスター・アドレスの設定方法

■ LLB

- ◆ スタンドアローン構成でのクラスター・アドレスの設定
 - dscontrol executor configureコマンドで手動設定する
 - goIdle/goInOpスクリプトにdscontrol executor configure/unconfigureコマンドを記述して自動設定することも可能
- ◆ HA構成でのクラスター・アドレスの設定
 - goActive/goStandby/goInOpスクリプトにdscontrol executor configure/unconfigureコマンドを記述して自動設定する
- ◆ 設定されたクラスター・アドレスはifconfig/ipconfigといったOSのコマンドで確認可能

■ ULB

- ◆ スタンドアローン構成およびHA構成でのクラスター・アドレスの設定
 - クラスターを構成すると自動的にロードバランスが開始されるため、設定する必要はない
 - dscontrol executor configure/unconfigureコマンドは使用できない
 - goActiveといったスクリプトを用意する必要はない
 - ✓ 通知やログ書き出しのために使用する事は可能だが、スクリプトからアドレスの設定を行ってはいけない
- ◆ 設定されたクラスター・アドレスはifconfigといったコマンドでは確認できない
 - dscontrol cluster addをすると、LBがarpリクエストに応答できるようになる
 - テイクオーバー発生時には、gratuitous arpが発行される

19

LLBでは、クラスター・アドレスはOS上に明示的に定義する必要がありました。そのためのコマンドとしてdscontrol executor configureコマンドが使用可能であり、スタンドアローン構成ではこのコマンドを手動実行することで、HA構成ではgoActive/goStandby/goInOpスクリプトにコマンドを記述することでクラスター・アドレスの構成を行います。

ULBでは、クラスター・アドレスをOS上に明示的に構成する必要はありません。クラスターを定義すると、自動的にクラスター・アドレス宛のリクエストのロードバランスが開始されます。

⑨ serverDownスクリプトの挙動

■ LLB

- ◆ ポート内の最後のサーバーがダウンした際には、serverDownは呼ばれず、managerAlertのみが呼ばれる
- ◆ ポートの全滅状態から一つのサーバーが復活した際には、managerClearが呼ばれ、未だダウンしているサーバーに対してserverDownコマンドが呼ばれる

■ ULB

- ◆ ポート内の最後のサーバーがダウンした際には、serverDownが呼ばれ、同時にmanagerAlertが呼ばれる
- ◆ ポートの全滅状態から一つのサーバーが復活した際には、serverUpが呼ばれ、同時にmanagerClearが呼ばれる

20

Load Balancerでは、割り振り先サーバーの障害を検知した場合に、ログの書き出しや通知に使用できるスクリプトとして、serverDown/serverUp/managerAlert/managerClearの4つのスクリプトがありました。

LLBでのスクリプトの挙動には若干の癖があり、スクリプトの使用に当たってはその挙動をよく理解する必要があります。

ULBではスクリプトの挙動が変更となり、LLBに比べると直感的にわかりやすいものとなりました。

新規にULB環境を構築する、あるいは新規にスクリプトの利用を開始する場合には、ULBのスクリプトの挙動を理解した上で、スクリプトを利用いただければ問題ありません。LLBを使用している環境からV7.0のLoad Balancer for IPv4 and IPv6への移行を行う場合、スクリプトの挙動の変更に注意を払う必要があります。スクリプト内のロジックを確認し、挙動の変更が影響を与えないか確認して下さい。特に、スクリプトからdscontrolコマンドを発行して振り分けの変更を行うなど、複雑なロジックが組み込まれている場合には、今回のスクリプトの挙動変更は大きな影響がありますのでご注意ください。

【注意事項 - Edge V7.0】Load Balancer for IPv4 and IPv6(ULB)でのserverUp/serverDownスクリプトの挙動について (WAS-09-045)

<http://www-06.ibm.com/jp/domino01/mkt/cnpages1.nsf/page/default-0004CADF>

⑩ インストール・ディレクトリ

■ LLB

- ◆ <EDGE_ROOT>/lb
 - UNIX、Linux
 - /opt/ibm/edge/lb
 - Windows
 - C:\Program Files\ibm\edge\lb
- ◆ dsserver/dscontrol
 - UNIX、Linux
 - /usr/binに配置される
 - Windows
 - C:\WINDOWS\System32に配置される

■ ULB

- ◆ <EDGE_ROOT>/ulb
 - UNIX、Linux
 - /opt/ibm/edge/ulb
 - Windows
 - C:\Program Files\ibm\edge\ulb
- ◆ dsserver/dscontrol
 - UNIX、Linux
 - /usr/binに配置されないので、<EDGE_ROOT>/ulb/binに配置されたスクリプトを直接実行する必要があります
 - Windows
 - C:\WINDOWS\System32には配置されませんが、Path環境変数に<EDGE_ROOT>/ulb/binが設定される

21

LLBとULBではインストール・ディレクトリ名が変更になりました。また、インストールされるパッケージ名や種類も変わっています。

また、LLBではUNIXとLinuxでは/usr/binディレクトリにdsserver.shとdscontrol.shスクリプトが、WindowsではC:\WINDOWS\System32ディレクトリにdsserver.cmdとdscontrol.cmdスクリプトが配置されていました。ULBではこれらPathの通ったディレクトリにスクリプトが配置されません。Windowsでは、インストール時にPath環境変数に<EDGE_ROOT>/ulb/binが設定されますが、UNIXとLinuxでは設定されませんので、<EDGE_ROOT>/ulb/binに配置されているスクリプトを直接実行するか、このディレクトリにPathを通す必要があります。

⑪ ワイルドカード・クラスターおよびワイルドカード・ポート

■ LLB

- ◆ ワイルドカード・クラスターが使用可能
 - ポート/サーバー構成が複数のクラスターで同一の場合などに使用する
- ◆ ワイルドカード・ポートが使用可の
 - Passive FTPのサポートのためにはワイルドカード・ポートの構成が必要
 - ポート21を構成した場合は自動的にワイルドカード・ポートが構成される。
 - 明示的に構成されていないポートへの要求を確実に破棄するためにも使用可能

■ ULB

- ◆ ワイルドカード・クラスター/ポートが使用不可
 - Passive FTPをサポートできない

22

LLBではワイルドカード・クラスターとワイルドカード・ポートが使用可能でした。

ULBではワイルドカード・クラスターとワイルドカード・ポートが使用する事ができなくなりました。このため、LLBでは可能であったPassive FTPのロードバランスがサポートされなくなりましたので注意が必要です。

⑫ ICMPメッセージの転送

■ LLB

◆ ICMPメッセージの転送および処理をサポートしない

- クラスター・アドレスに対してPing (ICMP Echo Request)を行うと、Load Balancer 自身がPingに応答する

■ ULB

◆ ICMPメッセージの転送や処理をサポート

- クラスター・アドレスに対してPingを行った場合、Load BalancerはICMPメッセージをバックエンドサーバに転送し、バックエンドサーバがPingに応答する

23

LLBはICMPの転送をサポートしていません。そのため、クラスター・アドレスに対してPingを行うと、LB自身が応答を返すという挙動をします。

ULBではICMPの転送および処理がサポートされました。どのような指針で転送や処理を行うかは以下のリンク先をご確認下さい。

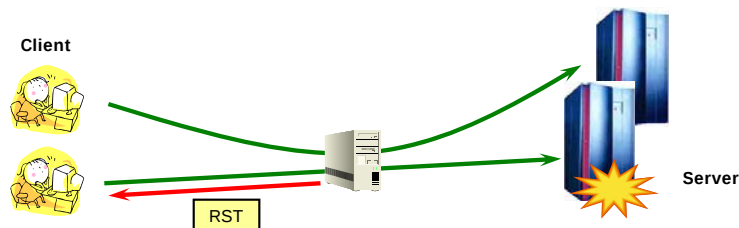
ICMP 転送およびメッセージングのサポート

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.edge.doc/lb/info/ae/cadm_icmp.html

⑬ TCP リセットの送信

■ LLB

- ◆ ヘルスチェックにより割り振り先サーバーの障害を検知した際に、接続中のクライアントにTCPリセットを送信するオプションを選択可能
 - クライアントは割り振り先サーバーのダウンを即時に知ることができる



■ ULB

- ◆ TCPリセットを送信するオプションがない
- ◆ Load Balancerは障害検知後もパケットの転送を継続する
 - 割り振り先サーバーのプロセスダウンであれば、クライアントが次にパケットを送信したタイミングでサーバーからRSTが返される
 - OSダウンやネットワーク障害の場合、クライアントは障害を知ることができず、TCPの再送タイムアウトまで待つことになる

24

LLBでは、ヘルスチェックにより割り振り先サーバーの障害を検知した際に、接続中のクライアントにTCPリセットを送信するオプションを選択可能でした。デフォルトでは送信しません。このオプションを選択することで、クライアントが割り振り先サーバーのダウンを即時に知ることが可能です。特にLDAPの割り振りなどで、常に接続が張りつぱなしになるようなケースでは、有用なオプションです。

ULBではこのオプションを選択することができません。

ULB と LLB の選択

ULB と LLB の選択

- 基本的にはULBを推奨
- LLBを選択するケース
 - ◆ 既存環境のバージョンアップであり、変更を極力少なくしたい場合
 - ◆ 非推奨機能を使用する必要がある場合
 - 例えばNAT転送やKCBR転送など
 - 非推奨機能もサポートは通常通り提供される
 - ◆ ULBとLLBの差異が問題となる場合
 - 同一筐体内に割り振れないことや、接続情報を共有していないことが問題となる場合
- LLBを選択した場合、いずれは構成の変更が必要になる
 - ◆ サポートは5+3年間
 - ◆ 次のバージョンへバージョンアップしたとするとそのバージョンのサポート期間中はサポートが受けられる
 - ◆ その次のバージョンではLLBがないかもしれないため、それまでに構成変更が必要
- そもそもOSの制限からLLBが選択できない場合もある
 - ◆ LLBはWindows 2008やRHEL5などには非対応
 - ◆ システム要件はノート部分のリンク先を参照

26

V7でLoad Balancerを使用する場合、ULBを選択するか、LLBを選択するかが重要な考慮事項となります。

基本的には将来のバージョンアップに備え、ULBが推奨となります。LLBを選択した場合は、将来のバージョンアップ時に構成の変更が必要になります。

LLBを選択するケースとしては、例えば、既存環境のバージョンアップであり、変更を極力少なくしたい場合があります。また、ULBでは使用できないNATやKCBR転送を使用する必要があるケースもLLBを選択せざるを得ません。また、ULBとLLBの間の細かい差異が問題となり、LLBを選択する必要があるケースもあります。

Support Policy Change: Announcing the new "5 plus 3" support policy

<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg21256700>

WebSphere product lifecycle dates

<http://www-01.ibm.com/software/websphere/support/lifecycle/>

WebSphere Application Server detailed system requirements

<http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27006921>