IPv4アドレス枯渇対応タスクフォース
2009年度活動中間報告
アプリケーションWG

2009年10月19日（月）
慶應義塾 協生館 2階
藤原洋 記念ホール

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

1. 主要関係企業・団体の動向について

(1) JISA
・基本的にSIerの集まりであり、連携模索中。会員向けアンケートの実施企画等、具体的提案が必要。

(2)IPA
・OSS推進フォーラム、試験制度等々、一部は関係を持ちつつも、OSC等で講演枠を確保しつつ、現在、連携模索中。
・「枯渇するとどうなる？」、「つながらない」、「おそい」、「セキュリティ」などが訴求しやすい。

(3)IBM
・IPv6関係では、これまで比較的、接点が多くないが、ITソリューションというカテゴリーでは主要プレーヤー。SEC－WGでISS事業部メンバーによる洗い出し作業が始まっている。

(4) Oracle/SUN
・WebLogic等のAppServer系は対応してきているが、DBは、11gR2で、正式にIPv6対応(2009年10月)

(5) JUS／LL
・本TFにて協働していただいている。

(6) Ruby
・7月にヒアリング実施。1.9.1でIPv6Onlyでの対応。Dual Stackでの対応に課題あり。

(7) その他の言語・アプリ
・現状はWebベースの情報収集
・ネットゲーム、リッチコンテンツ配信等、IPv4 Double NAT他で支障が出そうなアプリケーションの調査

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

2．事例:Perl、RubyにおけるIPv6対応状況

①Perl

a. Perl本体については、既に最新版で IPv6対応済み。
b. 一方、CPANのようなライブラリに関しては、まだ誰も手を付けていない状況。
c. Perl開発者の意識としては、v4アドレス枯渇に対して危機意識をもっていない。枯渇しても何らかの形で延命されると考えている。
d. 開発者の環境に IPv6 が無いというのが一番の問題。
e. 開発者に対して情報提供するのであれば、実際にIPv6対応したコードを出すことが必須。

②Ruby

a. 1.9.1では、最初にIPv6で宣言すると、IPv6 Onlyとなり、Dual Stackにならない。
b. 1.9.2で、現在、修正中。
c. サーバーが、特に、UDPの処理で、どうすればよりのか困っている状況。
d. IPアドレスを抽象化する仕組みはあるが、使わない傾向。もっと普及させる必要がある。
e. MLでは2006年8月に出たきり、かなりの期間、IPv6という言葉は見ていない。
f. クライアントはIPv6対応しているだろうから、HomeGWやルータがIPv6を通し始めれば、Ruby開発者も利用者も対応するはず。

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

3. 個別の言語等での対応に関する情報

C/C++等

http://jp.sun.com/practice/software/solaris/jp/wp/wp-ipv6/

Java | Solaris　コミュニティ | パートナー | My Sun | ご購入について　Japan | Worldwide

Sun microsystems　製品　ダウンロード　サービス　ソリューション　サポート　トレーニング　お知らせ　検索

ホーム > 製品情報 > ソフトウェア > Solaris > White Papers >

IPv6とインターネットの将来

IPv6とインターネットの将来

IPv6とインターネットの将来

PDF[261k] | PS[953K]

目次

第1章 はじめに
- インターネットの成長とインターネットプロトコル
- IPのバージョンに依存しないテクノロジー
- IPv6:次世代のプロトコル
- Solaris™オペレーティング環境:Webの基盤

第2章 IPv4とIPv6の比較:概要
- IPv4のバックグラウンド
- 一時的な対策としてのNAT
- 長期的なソリューションとしてのIPv6
- IPv6を使用した将来の計画

第3章 IPv6アプリケーション・プログラム・インタフェース
- IETF仕様
- 基本ソケットAPI
- 高度なソケット機能
- Sunリモートプロシージャ呼び出し

第4章 クライアントコードのIPv6へのポーティング
- ネームサービスとIPv4からIPv6への移行
- IPv4とIPv6の相互運用性
- IPv4とIPv6の命名(Naming)APIのちがい

第5章 サーバコードのIPv6へのポーティング 第6章 アプリケーションのアップグレードを支援するツール
- Socket Scrubber for C/C++
- Socket Scrubberの使用例

第7章 まとめ

第8章 付録
- IPv4からIPv6へのソケットの変更
- IPv4とIPv6のちがい

第9章 参考文献

第1章 はじめに

TCP/IPプロトコルスイートはインターネットの基盤を提供します。今日、企業や個人の間で行われているすべてのデータ通信は、Webブラウザ、電子メール、ファイル転送、リモートログインなどを含めて、事実上、これらのインターネットプロトコルに基づいています。今後数年間にわたって、インターネットは、現在のさまざまな制限を超えてインターネットの成長を可能にする重要な転換期に直面するはずです。この転換期には、インターネットプロトコル(IP)のより新しく、より信頼性の高いバージョン、つまりIPv6への移行も含まれます。さまざまな変更がTCP/IPプロトコルスイートのIP部分に加えられます。

インターネットの成長とインターネットプロトコル

アカウントの数を見るだけでも、インターネットは信じられない速度で成長を続けています。DNS(Domain Name Servers - すべての常設ホストシステムを定義するインターネットサービス)に関するInternet Software Consortiumの年次調査によれば、DNSの中で5,600万を超えるホストが識別されることが明らかになっています。前年の3,800万、1993年7月の200万未満という数字と比較すれば、この調査結果はインターネットの急速な成長を裏付けています。最新の数字は、インターネットが年率で65%を超える成長を続けていることを示しており、しかもこの数字にはDNSによって公式に発表されているシステムしか含まれていません。つまり、プライベートノードはこの調査の対象には含まれていないということです。

http://www.nslabs.jp/socket.rhtml

[Software] [Publications] [Ramble/Memo] [Services]

トップ > 技術メモ (ソフトウェア開発) > 分散オブジェクト・ネットワークプログラミング >

IPv6ソケットプログラミング

(2006.8.5 ページを独立。)

C/C++でのIPv6ソケットプログラミングについて。

ソケットについてはすでに掃いて捨てるほど解説サイトがあるが、IPv6に対応した、しかもポータブルな書き方を紹介しているところは見当たらなかった。

- 覚悟はできてますか? -トンでもなく高価なIPv6
- 報告書「IPv4アドレス枯渇に向けた提言」公開にあたって 2006.4.3

技術的に言えば、IPv6しか喋れないソフトウェア(クライアント)は、IPv4のサービス(サーバ)を利用できない。IPv6のサーバはIPv4しか喋れないクライアントからの接続を受けることができる。(IPv4-mapped address)

まず社会全体のサーバがIPv6に対応し、なおかつIPv4が廃れるまでIPv4アドレスも持つ必要がある。「すべての」クライアントがIPv6に対応するまでIPv4アドレスを捨てるわけにはいかない。

結局、「すべての」ソフトウェアがIPv4/IPv6の両方に対応するまでIPv4アドレスの需要が減らない。かなり無駄、不効率、不経済。本当になんでIPv6をIPv4と互換性を持たせるようにしなかったんだろう。

目次:

1. サーバを作る
2. クライアントを作る
3. IPv6, IPv4を区別するサーバ
4. 非同期 (non-blocking) I/OでIPv4/IPv6両対応

@ サーバを作る

まず、ソケットでIPv4あるいはIPv6クライアントからの接続を受け付けるプログラムを作ってみる。Fedora Core 5 Linuxで試した。

IPv4 onlyのときの典型的なコードは、次のようになる。

```
int sockfd = socket(AF_INET, SOCK_STREAM, 0);

sockaddr_in addr;
memset(&addr, 0, sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = htonl(INADDR_ANY);
addr.sin_port = htons(port);
bind(sockfd, (sockaddr*) &addr, sizeof(addr));
```

AF_INETはIPv4を指定するものだし、sockaddr_inはIPv4のアドレスしか格納できない。これではIPv6で通信はできない。ではどうするか。

ソケットアドレスの生成〜listen()

IPv6ということで、AF_INET6とsockaddr_in6を使ってみるか。それでもいいが、よりよいのは、getaddrinfo()に適切なソケットアドレス情報などを生成させ、それでソケットを生成し、bind()などを行う。

Java系

http://e-class.center.yuge.ac.jp/jdk_docs/ja/technotes/guides/net/ipv6_guide/index.html

ネットワーク IPv6 ユーザーガイド (JDK/JRE 5.0)

このドキュメントでは、次のトピックについて説明します。

- 概要
- サポートされるオペレーティングシステム
- Java での IPv6 の使用
- Java での IPv6 サポートの詳細
  - 特殊な IPv6 アドレス型
  - IPv6 関連のシステムプロパティー
  - デュアルスタックノード
  - Java アプリケーションへの影響
  - IPv6 のネットワークプロパティー

概要

このわずか数年で、IPv6 は業界で (特に、ヨーロッパとアジア太平洋地域の業界で)、幅広く受け入れられるようになりました。拡張性、モビリティー、サービス品質、広大なアドレス空間、自動 構成、セキュリティー、マルチホーム、エニーキャストとマルチキャスト、リナンバリングなどは、IPv6 で実現できる機能の一部です。

J2SE 1.4 が 2002 年 2 月にリリースされてから、Java では Solaris と Linux での IPv6 サポートを開始しました。Windows での IPv6 サポートは、J2SE 5.0 で追加されました。C や C++ などのその他の言語でも IPv6 をサポートしていますが、Java には大きな利点がいくつかあります。

- Java を使用すると、単一のコードベースに投資することで、IPv4 および IPv6 の両方に対応できる
- 既存の Java アプリケーションがすでに IPv6 対応している
- IPv6 への移行が容易

後述するコード例で以上の点を証明し、IPv6 サポートについて追加の情報を説明します。

サポートされるオペレーティングシステム

J2SE リファレンス実装では、次のオペレーティングシステムがサポートされています。

- Solaris 8 以降
- Linux カーネル 2.1.2 以降 (IPv6 のより良いサポートにはカーネル 2.4.0 以降を推奨)
- Windows XP SP1 および Windows 2003

Java での IPv6 の使用

IPv6 を Java で使用するのは簡単かつ透過的で自動的に行われます。その他の多くの言語とは異なり、移植は必要ありません。実際、ソースファイルの再コンパイルさえ必要 ありません。

「The Java Tutorial」からの例を示します。

```
Socket echoSocket = null;
   PrintWriter out = null;
   BufferedReader in = null;

   try {
      echoSocket = new Socket("taranis", 7);
      out = new PrintWriter(echoSocket.getOutputStream(), true);
      in = new BufferedReader(new InputStreamReader(
         echoSocket.getInputStream()));
   } catch (UnknownHostException e) {
      System.err.println("Don't know about host:taranis.");
```

http://w2c.jp/Seminar/2009/february/3_day/

現在位置: ホーム > セミナー > 共催セミナー

日本ウェブ協会 共催セミナー

Ajaxが動かなくなる？「IPv6移行時の問題を考える」

IPアドレス枯渇問題というのをご存知でしょうか。

現在使用されているIPv4というプロトコルは、近い将来にIPアドレスが不足してしまうと言われております。足りなくなったIPアドレスは、IPv6という別のプロコトルを利用せざる得ません。このIPv6への移行はスムースに行われるかどうかは疑問視されており、128桁というアドレスを使用し、ネットワークの構成が現在のIPv4とは全く異なるため、認証系のサービス等に問題が生じるのではないかと言われているのです。あるケースによっては、Ajaxが動かなくなるのでは！と。

参加自由の公開収録セミナーです

社団法人日本インターネットプロバイダ協会との共催で立石聡明様他をお招きしての参加自由の公開収録セミナーです。長野電算の菅沼 真様や、AVIS様など名前でISPを展開されており、既にIPv6によるサービスも始めていらっしゃる講師をお招きします。具体的にサーバにはどうしたらいいのかといった実装についてのお話も聞くことができる内容になっています。

内容としては、以下の構成を予定しています。

1. Pv4枯渇状況とIPv6導入の現状
2. IPv6サービス上の問題点(BフレッツやNGNでの問題)
   および、実装上の課題や導入について

新しいIPアドレスが貰えなければ、ネットワークは広げられません。これは、インターネットの可能性を大きく制限する可能性があり、実際、アジアのIPアドレス枯渇はすでにひっ迫した状況にあり、現在の在庫が枯渇するまで1000日を切ったといわれ、英Ovumのアナリスト Iain Stevensonもまた、「アジアでは2～3年以内にIPアドレスが枯渇するだろう」と述べています。

すでにIPv6にしたいからではなく、遡ばざるを得ない状況が生まれており、IPv6は、もはや夢探しのためのものではなくなっているのです。

現行のIPアドレス43億個に対し、米国などに割り当てられたIPアドレスは70%を占め、これに対し、IT大国と呼ばれるインドでさえ割り当てられたIPアドレスは、わずか200万個。この43億個のうち、未使用のIPアドレスは約10億個で、これは十分な数のように思えますが、ブロードバンド化が進む韓国や、中国、インド、欧州などがすでに割り当て分のIPアドレスを使い果たしつつあるため、日本でもIPアドレス枯渇への懸念が高まっているのです。

IPv6への移行問題は、インターネットプロバイダだけの問題ではなく、サイトを所有する企業や組織、あるいはウェブの制作に携わる方々も理解しておきたい内容です。

今回は内容の収録が主となりますが、会員以外の方でもどなたでも参加でき傍聴は自由です。

この機会をご利用下さい。※会場の収容人数には限りがございますが、傍聴をご希望の方は予めこちらからお気軽にお申込みください。

種別	セミナー
開催日	2009年2月3日 火曜日
予定時間	15時から18時まで(3時間)
	前半：IPv4枯渇状況とIPv6導入の現状

JavaでのIPv6サポート； http://www.lowtech.ii2.cc/j2se16/technotes/guides/net/ipv6_guide/index.html

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

Perl、PHP

http://www.geekpage.jp/programming/perl-network/

http://handsout.jp/slide/982

http://info.2ch.net/wiki/index.php?IPv6

PHPプログラミングのIPv6対応の実際他
http://handsout.jp/slides/search_tag?name=PHP%2BIPv6
PHPでIPv6プログラミング／クライアント編
http://d.hatena.ne.jp/kajidai/20081110/1226254561

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

Ruby、Python

http://www.ruby-lang.org/ja/man/html/ipaddr.html

IPAddr

IP アドレスを扱うためのクラスです。

```
require 'ipaddr'

ipaddr1 = IPAddr.new "3ffe:505:2::1"
p ipaddr1       #=> #<IPAddr: IPv6:3ffe:0505:0002:0000:0000:0000:0000:0001/ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff>

ipaddr2 = IPAddr.new "192.168.2.0/24"
p ipaddr2       #=> #<IPAddr: IPv4:192.168.2.0/255.255.255.0>
```

クラスメソッド

IPAddr.new(*addr = '::', family = Socket::AF_UNSPEC*)

新しい IPAddr オブジェクトを生成します。addr は `address/prefixlen' や `address/mask' などの形式も受け付けます。プリフィックス長やマスクが指定されると、マスクされた IPAddr オブジェクトを返します。IPv6 アドレスの場合は、[ ] で囲まれていてもかまいません。family は自動的に判定されますが、明示的に指定することもできます。

IPAddr.new_ntoh(*addr*)

ネットワークバイトオーダーのバイト列から IPAddr オブジェクトを生成します。

```
require 'ipaddr'
p IPAddr.new_ntoh("\x00\xa9\x25\x00\x01\x31") #=> #<IPAddr: IPv4:192.168.1.1/255.255.255.255>
```

IPAddr.ntop(*addr*)

ネットワークバイトオーダーのバイト列で表現された IP アドレスを人間の読める形式に変換します。

メソッド

&(*ipaddr*)

他の IPAddr オブジェクトとのビットごとの論理積により、新しい IPAddr オブジェクトを生成します。

|(*ipaddr*)

他の IPAddr オブジェクトとのビットごとの論理和により、新しい IPAddr オブジェクトを生成します。

~

ビットごとの論理否定により、新しい IPAddr オブジェクトを返します。

>>(*num*)

ビットごとの右シフト演算により、新しい IPAddr オブジェクトを返します。

<<(*num*)

ビットごとの左シフト演算により、新しい IPAddr オブジェクトを返します。

==(*ipaddr*)

IPAddr オブジェクト同士が等しいかを比較します。

```
require "ipaddr"
p IPAddr.new("192.168.0.1") == IPAddr.new("192.168.0.1/24") #=> false
```

http://www.python.jp/doc/2.4/lib/module-socket.html

Pythonライブラリリファレンス

Previous: 7.1.1 例 Up: 7. オプションのオペレーティングシステムサービス Next: 7.2.1 socket オブジェクト

7.2 socket -- 低レベルネットワークインターフェース

このモジュールは、PythonでBSD *ソケット*インターフェースを利用するために使用します。最近のUNIXシステム、Windows, MacOS, BeOS, OS/2など、多くのプラットフォームで利用可能です。

C言語によるソケットプログラミングの基礎については、以下の資料を参照してください。*An Introductory 4.3BSD Interprocess Communication Tutorial* (Stuart Sechrest), *An Advanced 4.3BSD Interprocess Communication Tutorial* (Samuel J. Leffler他), *UNIX Programmer's Manual, Supplementary Documents 1* (PS1:7章 PS1:8章)。ソケットの詳細については、各プラットフォームのソケット関連システムコールに関するドキュメント(UNIXではマニュアルページ、WindowsではWinSock(またはWinSock2)仕様書)も参照してください。IPv6対応のAPIについては、RFC 2553 *Basic Socket Interface Extensions for IPv6*を参照してください。

Pythonインターフェースは、UNIXのソケット用システムコールとライブラリを、そのままPythonのオブジェクト指向スタイルに変換したものです。各種ソケット関連のシステムコールは、`socket()`関数で生成する *ソケットオブジェクト*のメソッドとして実装されています。メソッドのパラメータはCのインターフェースよりも多少高水準で、例えば`read()`や`write()`メソッドではファイルオブジェクトと同様、受信時のバッファ確保や送信時の出力サイズなどは自動的に処理されます。

ソケットのアドレスは以下のように指定します:単一の文字列は、`AF_UNIX`アドレスファミリを示します。*(host, port)*のペアは`AF_INET`アドレスファミリを示し、*host*は`'darking.cwi.nl'`のようなインターネットドメイン形式または`'100.50.200.5'`のようなIPv4アドレスを文字列で、*port*はポート番号を整数で指定します。`AF_INET6`アドレスファミリは *(host, port, flowinfo, scopeid)*の長さ4のタプルで示し、*flowinfo*と*scopeid*にはそれぞれCの`struct sockaddr_in6`におけるsin6_flowinfoと`sin6_scope_id`の値を指定します。後方互換性のため、`socket`モジュールのメソッドでは`sin6_flowinfo`と`sin6_scope_id`を省略する事ができますが、*scopeid*を省略するとスコープを持ったIPv6アドレスの処理で問題が発生する場合があります。現在サポートされているアドレスファミリは以上です。ソケットオブジェクトで利用する事のできるアドレス形式は、ソケットオブジェクトの作成時に指定したアドレスファミリで決まります。

IPv4アドレスのホストアドレスが空文字列の場合、`INADDR_ANY`として処理されます。また、`'<broadcast>'`の場合は`INADDR_BROADCAST`として処理されます。IPv6では後方互換性のためこの機能は用意されていませんので、IPv6をサポートするPythonプログラムでは利用しないで下さい。

IPv4/v6ソケットの*host*部にホスト名を指定すると、処理結果が一定ではない場合があります。これはPythonはDNSから取得したアドレスのうち最初のアドレスを使用するので、DNSの処理やホストの設定によって異なるIPv4/6アドレスを取得する場合があるためです。常に同じ結果が必要であれば、*host*に数値のアドレスを指定してください。

エラー時には例外が発生します。引数型のエラーやメモリ不足の場合には通常の例外が発生し、ソケットやアドレス関連のエラーの場合は`socket.error`が発生します。

`setblocking()`メソッドで、非ブロッキングモードを使用することができます。また、より汎用的に`settimeout()`メソッドでタイムアウトを指定する事ができます。

`socket`モジュールでは、以下の定数と関数を提供しています。

exception error
この例外は、ソケット関連のエラーが発生した場合に送出されます。例外の値は障害の内容を示す文字列か、または`os.error`と同様な *(errno, string)*のペアとなります。オペレーティングシステムで定義されているエラーコードについては`errno`を参照してください。

exception herror
この例外は、C APIの`gethostbyname_ex()`や`gethostbyaddr()`などで、*h_errno*のようなアドレス関連のエラーが発生した場合に送出されます。

例外の値は *(h_errno, string)*のペアで、ライブラリの呼び出し結果を返します。*string*はC関数`hstrerror()`で取得した、*h_errno*の意味を示す文字列です。

exception gaierror
この例外は`getaddrinfo()`と`getnameinfo()`でアドレス関連のエラーが発生した場合に送出されます。

例外の値は *(error, string)*のペアで、ライブラリの呼び出し結果を返します。*string*はC関数`gai_strerror()`で取得した、*h_errno*の意味を示す文字列です。*error*の値は、このモジュールで定義される`EAI_*`定数の何れかとなります。

exception timeout
この例外は、あらかじめ`settimeout()`を呼び出してタイムアウトを有効にしてあるソケットでタイムアウトが生じた際に送出されます。例外に付属する値は文字列で、その内容は現状では常に``timed out''となります。バージョン 2.3 で 新たに追加された仕様です。

`AF_UNIX`
`AF_INET`
`AF_INET6`

PostgreSQL、MySQL

http://www.postgresql.jp/document/pg745doc/html/datatype-net-types.html

PostgreSQL 7.4.5 文書
第 8 章データ型

前のページ　概要　早送り　次のページ

8.8. ネットワークアドレス型

PostgreSQL ではIPv4アドレス、IPv6アドレス、MACアドレスを格納するデータ型が提供されています。表8-17を参照してください。これらのデータ型は入力値のエラー検査と専用の演算子と関数を補助提供していますので、普通のテキストデータ型の代わりにこちらを使うことをお勧めします。

表 8-17. ネットワークアドレスデータ型

名前	格納サイズ	説明
cidr	12もしくは24バイト	IPv4もしくはIPv6ネットワーク
inet	12もしくは24バイト	IPv4もしくはIPv6ホスト、および、ネットワーク
macaddr	6バイト	MACアドレス

inetもしくはcidrをソートする時、IPv4アドレスは常にIPv6よりも前にソートされます。::10.2.3.4 or ::ffff:10.4.3.2などIPv6アドレス内に埋め込まれた、もしくは、関連付けされたIPv4アドレスも同様です。

8.8.1. inet

inet型はIPv4もしくはIPv6ホストアドレスとオプションでそのホストアドレスが所属するサブネット識別情報を1つのフィールドに保持します。サブネット識別情報はホストアドレスの内何ビットがネットワークアドレス("netmask")を表すかを指定することで表現されます。もしネットマスクが32の場合、IPv4では単一ホストを意味し、サブネットを示しません。IPv6ではアドレス長は128ビットです。そのため128ビットが一意なホストアドレスを指定します。ネットワークのみを使用したい場合はinetではなくcidr型を利用してください。

このデータ型に対する入力書式は、IPv4、IPv6両方ともaddress/yです。yがネットマスクのビット数です。もし/yの部分を省略するとネットマスクはIPv4では32、IPv6では128となり、その値は単一ホストを表現します。ネットマスクが単一ホストを表す場合、その表示時、/yの部分は抑制されます。

8.8.2. cidr

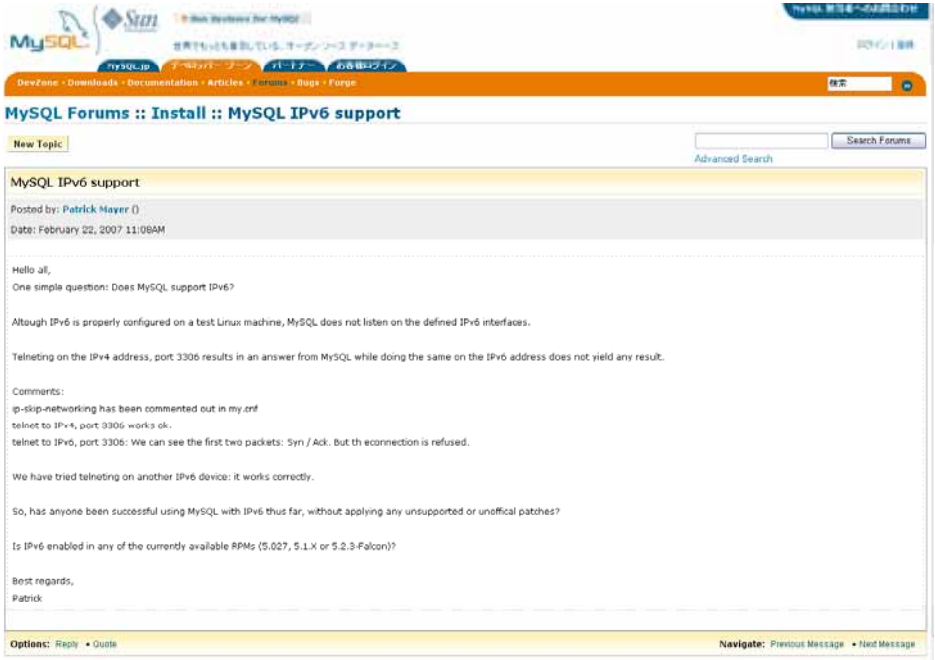
cidrデータ型はIPv4、IPv6ネットワーク仕様を保持します。入出力書式はクラス分けのないインターネットドメインルーティング協定に従います。ネットワークを指定する時の書式はaddress/yで、addressがIPv4もしくはIPv6アドレスで表したネットワークです。yはネットマスクのビット数です。もし/yが省略された場合には、入力時に書き込まれたオクテットすべてが含まれるように大きさが維持されること以外は、従来のクラス付きアドレス番号指定システムに従って計算されます。指定したネットマスクの右側にビットをセットしたネットワークアドレスを指定するとエラーになります。

表8-18に例がいくつかあります。

表 8-18. cidrデータ型入力例

cidr 入力	cidr 出力	abbrev(cidr)
192.168.100.128/25	192.168.100.128/25	192.168.100.128/25
192.168/24	192.168.0.0/24	192.168.0/24
192.168/25	192.168.0.0/25	192.168.0.0/25
192.168.1	192.168.1.0/24	192.168.1/24
192.168	192.168.0.0/24	192.168.0/24
128.1	128.1.0.0/16	128.1/16
128	128.0.0.0/16	128.0/16
128.1.2	128.1.2.0/24	128.1.2/24
10.1.2	10.1.2.0/24	10.1.2/24
10.1	10.1.0.0/16	10.1/16
10	10.0.0.0/8	10/8
10.1.2.3/32	10.1.2.3/32	10.1.2.3/32
2001:4f8:3:ba::/64	2001:4f8:3:ba::/64	2001:4f8:3:ba::/64
2001:4f8:3:ba:2e0:81ff:fe22:d1f1/128	2001:4f8:3:ba:2e0:81ff:fe22:d1f1/128	2001:4f8:3:ba:2e0:81ff:fe22:d1f1
::ffff:1.2.3.0/120	::ffff:1.2.3.0/120	::ffff:1.2.3/120
::ffff:1.2.3.0/128	::ffff:1.2.3.0/128	::ffff:1.2.3.0/128

http://forums.mysql.com/read.php?11,141295



MySQL Forums :: Install :: MySQL IPv6 support

New Topic　Search Forums　Advanced Search

MySQL IPv6 support

Posted by: Patrick Mayer ()
Date: February 22, 2007 11:08AM

Hello all,
One simple question: Does MySQL support IPv6?

Altough IPv6 is properly configured on a test Linux machine, MySQL does not listen on the defined IPv6 interfaces.

Telneting on the IPv4 address, port 3306 results in an answer from MySQL while doing the same on the IPv6 address does not yield any result.

Comments:
ip-skip-networking has been commented out in my.cnf
telnet to IPv4, port 3306 works ok.
telnet to IPv6, port 3306: We can see the first two packets: Syn / Ack. But th econnection is refused.

We have tried telneting on another IPv6 device: it works correctly.

So, has anyone been successful using MySQL with IPv6 thus far, without applying any unsupported or unoffical patches?

Is IPv6 enabled in any of the currently available RPMs (5.027, 5.1.X or 5.2.3-Falcon)?

Best regards,
Patrick

Options: Reply • Quote　Navigate: Previous Message • Next Message

Oracle

http://www.oracle.com/technology/products/oraclenet/index.html

shortcuts | GETTING STARTED | DOWNLOADS | DOCUMENTATION | FORUMS | ARTICLES | SAMPLE CODE | TUTORIALS

Printer View

ORACLE DATABASE 11g

Oracle Net Services

What's New in Oracle Database 11gR1 & 11gR2

The new features for Net Services in Oracle Database 11g Release 2 include:

- Internet Protocol Version 6 (IPv6) Addressing and Connectivity for single-instance database (please refer the IPv6 Statement of Direction below for more information).
- Oracle Restart in single instance environment for restarting the listener, the database and other Oracle components following hardware/software failure or host restart.
- Support for CONNECT_TIMEOUT and TRANSPORT_CONNECT_TIMEOUT for each IP address
- CIDR notation and wildcard support for valid node checking

Technical Information

Oracle OpenWorld 2009

- Oracle Net Services: Best Practices for Database Performance and Scalability (PDF)

Oracle Database 11gR2

- IPv6 Statement of Direction (PDF)

注：Oracle Application Serverは10gで対応済み

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

DB2他

http://edocs.beasys.co.jp/e-docs/wls/docs103/jdbc_drivers/usedriver.html

IP アドレスの使用

WebLogic Type 4 JDBC ドライバでは、IPv4 および IPv6 形式の IP (インターネット プロトコル) アドレスがサポートされます。IPv6 アドレスは、特定のバージョンのデータベースに接続する場合にのみサポートされます (表 2-2 を参照)。また、IPv6 アドレスに接続するためには、Windows では J2SE 5.0 以上、UNIX/Linux では J2SE 1.4 が必要になります。

表 2-2 各 WebLogic Type 4 JDBC ドライバでサポートされる IP アドレス形式

ドライバ	IPv4	IPv6
DB2	サポートされているすべてのバージョン	z/OS 上の DB2 v9.1 Linux、UNIX、Windows 上の DB2 V9.1 以上 iSeries 上の DB2 V5R2 以上
Informix	サポートされているすべてのバージョン	Informix 10 以上
Oracle	サポートされているすべてのバージョン	サポートされていない
Microsoft SQL Server	サポートされているすべてのバージョン	Microsoft SQL Server 2005 以上
Sybase	サポートされているすべてのバージョン	Sybase 12.5.2 以上

ネットワークで名前付きサーバがサポートされている場合は、接続 URL またはデータ ソースに指定されたサーバ名を、IPv4 または IPv6 アドレスに解決できます。たとえば、次の URL のサーバ名 DB2Server を、いずれかのタイプのアドレスに解決できます。

```
jdbc:bea:db2://DB2Server:50000;DatabaseName=jdbc;User=test;
Password=secret
```

また、接続 URL のサーバ名部分に、IPv4 または IPv6 形式のアドレスを指定することもできます。たとえば、次の接続 URL では、IPv4 形式でサーバを指定しています。

```
jdbc:bea:db2://123.456.78.90:50000;DatabaseName=jdbc;User=test;
Password=secret
```

ServerName データ ソース プロパティを使用して、いずれかの形式でアドレスを指定することも可能です。次のデータ ソース定義では、IPv6 形式でサーバ名を指定しています。

```
DB2DataSource mds = new DB2DataSource();
mds.setDescription("My DB2DataSource");
mds.setServerName("[ABCD:EF01:2345:6789:ABCD:EF01:2345:6789]");
mds.setPortNumber(50000);
...
```

注意：接続 URL やデータ ソースで IPv6 アドレスを指定する場合は、そのアドレスを括弧で囲む必要があります。

WebLogic Type 4 JDBC ドライバでは、通常の IPv6 形式に加え、圧縮したアドレスや IPv4 と IPv6 を組み合わせたアドレスなど、IPv6 の代替形式もサポートされます。たとえば、次の接続 URL では IPv6 形式でサーバを指定していますが、連続するゼロのビットを圧縮する構文が使用されています。

```
jdbc:bea:db2://[2001:DB8:0:0:8:800:200C:417A]:50000;DatabaseName=jdbc;
User=test;Password=secret
```

次の接続 URL でも IPv6 形式でサーバを指定していますが、IPv4 と IPv6 を組み合わせたアドレスが使用されています。

```
jdbc:bea:db2://[0000:0000:0000:0000:0000:FFFF:123.456.78.90]:50000;
DatabaseName=jdbc;User=test;Password=secret
```

IPv6 の詳細については、次のページを参照してください。

http://tools.ietf.org/html/rfc4291#section-2.2

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

4．その他の関連情報URL

http://www.oracle.co.jp/showcase/2005/ondemand_tv/top.html
OndemandTVがOracleで構築してv6対応

http://otndnld.oracle.co.jp/products/ias/pdf/ohs_sod.pdf
Oracle Application Server Release2は、Apache2.0に対応しているからIPv6対応との発表

http://otndnld.oracle.co.jp/document/products/as10g/101340/doc_cd_ps/relnotes.1013/B50870-02/install.htm
OracleAS10gR3は、v6対応していない？？

http://blog.goo.ne.jp/xmldtp/e/368e5a1769cadbc345d66189185103f7
OracleのINET型は両方対応？

http://www.citrix.co.jp/company/press/releases/20081204.html
CitrixのNetScalerでIpv6対応。OracleE-businessSuiteもOK？

http://blogs.oracle.com/stevenChan/2008/05/ipv6_certified_with_oracle_ebu.html
IPv6のReverse ProxyでOK？

http://download.oracle.com/docs/cd/E13161_01/tuxedo/docs10gr3/relnotes/relnotes.html
C/C++、COBOL向けのTuxedoR3はIPv6対応！

http://win6.jp/
加藤淳也さんのサイト

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan

5. その他について

活動ML； v4exh-app@mri.co.jp

現状参加；

江崎さん (v6pc/WIDE)
荒野さん (v6pc/インテックネットコア)
松山さん (jus/創夢)
菅沼さん (JAIPA/電算)
向井さん (JANOG/KDDI)
佐藤さん (IAj)
澤部さん (v6pc/MRI)
津國さん (v6pc/MRI)
中村 (v6pc/MRI)

情報のご提供を！

Copyright (C) 2009 Task Force on IPv4 Address Exhaustion, Japan