

平成 22 年度
ネットワーク位置情報システム
構築・運用に関するガイドライン

第 0.10 版

2011 年 3 月 22 日

変更履歴

版	変更日	変更者	変更頁	変更内容
0.10	2011 年 3 月 22 日			IETF ALTO-06 対応初版

目 次

1	概要.....	1
1.1	本ガイドラインの目的	1
1.2	対象読者	1
1.3	記述について	1
2	推奨スペック	2
2.1	ハードウェア推奨スペック	2
2.2	ソフトウェア推奨スペック	2
3	構築手順.....	3
3.1	環境設定	3
3.1.1	基本的な環境の設定	3
3.2	SYSLOG の設定	3
3.3	MySQL の設定	4
3.4	APACHE HTTPD の設定.....	4
3.5	MEMCACHED のインストール.....	4
3.6	PERL の CPAN モジュールのインストール	4
3.7	ヒントサーバソースコードのインストール	5
3.8	ヒントサーバのユーティリティツールのインストール	5
4	データベースへの初期データ投入.....	6
4.1.1	地域情報データベース(GeoIP)の登録.....	6
4.1.2	BGP フルルート情報の取得と整形.....	6
4.1.3	サブネットと PID のマッピング.....	6
4.1.4	AS リンク情報の抽出.....	7
4.1.5	link_map テーブルへの投入.....	7
4.1.6	link_map テーブルから最短経路の計算と cost_map を生成	8
5	運用手順.....	9

1 概要

1.1 本ガイドラインの目的

本ガイドラインは、平成 22 年度「トラヒックの集中回避のためのネットワーク制御に関する実証実験」において開発された「ネットワーク位置情報システム（以下、ヒントサーバという）」の構築・運用の実績をもとに、同様なサービスを利用したい事業者等が容易に構築・運用ができるように記したものである。

なお、本ガイドラインはあくまで実証実験を目的としたものであり、商用利用のための考慮を含んでいないことに留意すること。商用利用のためには、別途、セキュリティ対策、負荷対策、可用性に対する対策等を講じる必要がある。

1.2 対象読者

本ガイドラインは、ヒントサーバの構築・運用を望む一般的な技術水準の Linux 技術者を対象として記載する。

1.3 記述について

本書内では以下のように記述する。

- `[]` 省略可能な部分を示す。
- `{A|B}` A or B の選択肢を示す。
- 背景色の部分はシェルにて実行するコマンドラインおよびその応答画面を示す。
- `#` で始まる行は root 権限において実行することを示す。

2 推奨スペック

2.1 ハードウェア推奨スペック

ハードウェアの推奨スペックを表 2.1-1 に示す。同様に、実際に平成 22 年度の実証実験において使用されたヒントサーバのスペックを参考情報として記した。

表 2.1-1 ハードウェア推奨スペック

項目	推奨スペック	参考情報 (H22 年度ヒントサーバ 実際の仕様)
アーキテクチャ	IA32-64bit 互換機	IA-32-64bit 互換機
CPU	Xeon マルチコア以上推奨	Xeon 5570/2.93GHz
メインメモリ	8GB 以上	32GB
ハードディスク	300GB 以上	2.7TB
ネットワークインターフェース	100Base-TX 以上	1000Base-TX

2.2 ソフトウェア推奨スペック

ソフトウェアの推奨スペックを表 2.2-1 に示す。同様に、実際に平成 22 年度の実証実験において使用されたヒントサーバのスペックを参考情報として記した。

表 2.2-1 ソフトウェア推奨スペック

項目	推奨スペック	参考情報 (H22 年度ヒントサーバ 実際の仕様)
OS	Linux CentOS 5.5 (64bit) 以上	Linux CentOS 5.5 (64bit) kernel 2.6.18
httpd	Apache 2.2 以上	Apache 2.2.3
RDBMS	MySQL 5.1 以上	MySQL 5.1.50
メモリキャッシュ	memcached 1.4.5 以上	memcached 1.4.5

Perl	Perl 5.8 以上	Perl 5.8.8
PHP	PHP 5.2 以上	PHP 5.2.14

3 構築手順

以下の構築手順は、CentOS 5.5 (64bit)を前提とする。

3.1 環境設定

3.1.1 基本的な環境の設定

Linux パッケージ管理システムである yum を更新しておく(yum update)。

他に、iptables のファイアウォール設定、および ssh のセキュリティ設定をしておくことが望ましい。表 3.1-1 に記載のポート以外は基本的に不要であるので閉じる。また ssh は標準ポート(TCP22)ではなく、独自のポート番号に設定しておくことが望ましい(不正攻撃を抑止するため)。

表 3.1-1 ヒントサーバが使用するポート

種別	ポート番号	用途
TCP/UDP	53	DNS
UDP	123	ntp
TCP	80 (変更が望ましい)	HTTP
TCP	22 (変更が望ましい)	ssh

また、必須ではないが、DNS の多用による速度低下を抑制するため、DNS キャッシュ(dnsmasq デモン)を有効にしておくといよい。dnsmasq は Linux CentOS5.5 には標準でインストールされているがデフォルトではオフになっている。

3.2 syslog の設定

ヒントサーバは syslog を利用してログを出力するので、syslog の設定が必要である。以下のように設定する。

```
# vi /etc/syslog.conf
```

以下の行を追加する。

```
local2.* /var/log/hintsvr/peerselection.log
```

変更を反映する。

```
# /sbin/service syslog restart
```

また、あわせて logrotate の設定を必要に応じて変更しておくが良い。logrotate は標準では 4 週間分のログしか保存しない設定になっているため注意する。

3.3 MySQL の設定

MySQL に別紙 1 の SQL 文 (geoloc_create.sql) を投入してデータベースを作成する。

```
# mysql -p -u root mysql
mysql> create database geoloc;
mysql> exit
# mysql -p -u root geoloc < geoloc_create.sql
```

3.4 Apache httpd の設定

Apache をインストールしたら、Perl モジュール(mod_perl)をインストールし、Perl スクリプトを利用可能にしておく。

ヒントサーバは比較的負荷の高いサービスであるため、環境に応じてパフォーマンスのチューニングをしておくことが望ましい。

3.5 memcached のインストール

オープンソースのメモリキャッシュデーモンである memcached を使用する。

<http://memcached.googlecode.com/> を参照しインストールする。

3.6 Perl の CPAN モジュールのインストール

ヒントサーバのソースコードは現在 Perl で記述されており、Perl のライブラリである CPAN ライブラリを使用している。いくつか標準ではインストールされていないモジュールに依存しているため、表 3.6-1 の追加モジュールをインストールする。

表 3.6-1 必要な CPAN ライブラリ追加モジュール

No.	モジュール名	内容
1	DBI	データベース接続インターフェース。
2	DBD::mysql	MySQL 接続インターフェース。
3	Cache::memcached::Fast	memcached 接続インターフェース。
4	NetAddr::IP	IP アドレス処理ユーティリティモジュール。
5	JSON::XS	JSON モジュール。

3.7 ヒントサーバソースコードのインストール

別途開示するヒントサーバのソースコードのうち、http サービス用スクリプト (Perl スクリプト) を入手し、インストールする。

インストール先は通常、`/var/www/html/hintsvr/` である。Apache の `httpd.conf` を適切に設定して Perl スクリプトが動作するようにしておくこと。

3.8 ヒントサーバのユーティリティツールのインストール

別途開示するヒントサーバのソースコードのうち、データ加工等に使用するバッチユーティリティツールを別途インストールする。インストール先は通常、`/usr/local/hintsvr/` としているが、必要に応じて変更しても良い。インストールしたらユーティリティツールには表 3.8-1 のようなツールが含まれる。

表 3.8-1 バッチユーティリティツール

No.	ツール名	内容
1	csv_from_cisco.pl	CISCO フォーマットの BGP フルルート情報から、ヒントサーバに必要な経路情報に加工するツール。
2	format_bgp.pl	BGP データを所定の形式に加工するツール。
3	add_link_map.pl	リンク情報ファイルを分析してデータベースに登録するツール。
4	calc_cost_from_link_map.pl	link_map から dijkstra アルゴリズムによって最短経路を算出し、cost_map マトリックスを生成するツール。

4 データベースへの初期データ投入

構築が完了したら、ネットワーク距離計算に必要な各種データをデータベースに投入する。

4.1.1 地域情報データベース (GeoIP) の登録

GeoIP 情報を mysql の geoip テーブルに投入する。

※ GeoIP 情報は市販の商用データベースであるため、本ガイドラインでは詳細は省略する。

4.1.2 BGP フルルート情報の取得と整形

- (1) 各 ISP の対外接続ルータで 'show ip bgp' 等のコマンドにより BGP フルルートをダンプし、ターミナルソフトのログ出力機能等によりテキストファイルに出力する。これを my_AS_ISP-A_YYYYMMDD.txt として保存する。
- (2) 精度を上げるためには、できるだけ多くの ISP から BGP フルルート情報を得るのが良い。複数のフルルート情報は、cat コマンドで単純にマージして一つのファイルにしておく。
- (3) 上記フルルート情報を加工して "<prefix>,<AS path>" からなる CSV ファイルを生成する。対応していないルータベンダの場合は新規に作成する必要がある。<my_as>の値は、ASpath の先頭を見る、もしくは whois 等で取得する。パッチユーティリティツールの " csv_from_cisco.pl " を利用する。

コマンド利用例：ISP-A より入手した BGP データから CSV を生成する例。

```
% ./csv_from_cisco.pl <my_AS> my_AS_ISP-A.txt > my_AS.csv
```

作成された my_AS.csv ファイルの例 (head コマンドで先頭を表示)：

```
% head my_AS.csv
1.9.0.0/16,2516 4788
1.11.128.0/17,2516 9318 38091 17839
```

4.1.3 サブネットと PID のマッピング

ヒントサーバの初期値では AS 番号一つごとに PID を一つ持つ。(このあと必要に応じて PID を任意に追加することもできる。)

この初期値を 4.1.2 で作成したデータをもとに以下のようにして登録する。

- (1) 4.1.2 で作成したフルルート情報 `my_AS_ISP-A.txt` を用意する。
- (2) AS 番号を羅列したテキストファイル `ASLIST.txt` を用意する。AS 番号は JPNIC にて一般に公開されている。本実証実験では国内 AS と一部の海外 AS のみ約 700 件を登録したファイルを使用した。ASLIST に含まれていない AS 番号は下記の処理ですべて `AS=0` と置換され、「その他 AS」として接続優先度の低いグループとして扱われる。
- (3) バッチユーティリティツールの”`add_route.pl`”を利用して `pid_map` に登録する。

コマンド利用例：

```
% ./add_route.pl my_AS_ISP-A.txt ASLIST.txt
```

4.1.4 AS リンク情報の抽出

4.1.2 で作成したフルルート情報から、バッチユーティリティツール”`format_bgp.pl`”によりリンク情報を抽出する。リンク情報とは、AS ペアのピアリング接続の有無と、そのリンクコストを”`<AS1> <-> <AS2> COST`”の書式で表したテキストファイルである。

コマンド利用例：

```
% ./format_bgp.pl -pb ASLIST.txt my_AS_ISP-A.txt > ASconnpairs.txt
```

作成された `ASconnpairs.txt` ファイルの例（`head` コマンドで先頭を表示）：

```
% head ASconnpairs.txt
2497 <-> 0 50
2497 <-> 2516 1
2497 <-> 17676 1
```

4.1.5 link_map テーブルへの投入

上記で作成した `ASconnpairs.txt` を `link_map` テーブルに投入する。バッチユーティリティツールの”`add_link_map.pl`”を利用する。

コマンド利用例：

```
% ./add_link_map.pl ASconnpairs.txt
```

投入されたデータは以下のようにして確認できる。

```
mysql> select * from link_map limit 2;
```

id	src_as	dst_as	link_cost	flag
1	2497	0	50	0
2	2497	2516	1	0

4.1.6 link_map テーブルから最短経路の計算と cost_map を生成

link_map テーブルにデータが入った状態で、バッチユーティリティツールの”calc_cost_from_link_map.pl”を利用し、link_map から cost_map を生成する。このツールは link_map テーブルの情報を dijkstra アルゴリズムにかけ、最短経路を算出し、任意の 2 点間の最小 link_cost を計算し、cost_map テーブルに投入するという一連の処理を実行する。

コマンド利用例:

```
% ./calc_cost_from_link_map.pl
```

生成されたデータは以下のようにして確認できる。

```
mysql> select * from cost_map limit 2;
```

src	dst	cost	note	src_as	dst_as	hops	aspath
0	11337728	12800	NULL	0	173	50	0 173
0	163643392	12800	NULL	0	2497	50	0 2497

5 運用手順

5.1 データベースのメンテナンス

データベースは次のような場合にデータの更新、追加、または削除をおこなう。

(1) AS 番号や IP プレフィックスに改定があったとき

AS 番号、IP プレフィックスともに数ヶ月に 1 回の頻度で改定があるので、改定があったときにデータベースを 4 章の手順により再構築する。

(2) ルーティング情報に変化があったとき

ルーティング情報や ISP 間のピアリング接続は数カ月に 1 回は更新されるので、改定があったときはデータベースを 4 章の手順により再構築する。

5.2 ログのメンテナンス

5.2.1 ログの種類

ログは以下の 3 つが生成される。運用中はそれぞれ定期的に監視する。

表 5.2-1 ログ一覧

No.	ファイル名	内容
1	access-log	アクセスログ (Apache httpd 互換)
2	error-log	エラーログ (Apache httpd 互換)
3	peerselection.log	コスト応答ログ (独自フォーマット)

それぞれのログのサンプルを表 5.2-2 に添付する。

表 5.2-2 ログのサンプル

ファイル名	ログのサンプル
access-log	202.178.*.* - [17/Feb/2011:01:50:00 +0900] "POST /hintsvr/EndpointCostLookup HTTP/1.1" 200 200 "-" "UGLIVE/3.3.17_fixfor20110120" 202.178.*.* - [17/Feb/2011:01:50:44 +0900] "POST /hintsvr/EndpointCostLookup HTTP/1.1" 200 200 "-"

error-log	[Mon Jan 31 20:58:09 2011] [error] [client 202.178.*.*] File does not exist: /var/www/html/bad_request_uri
peerselection.log	Dec 22 16:47:18 p2p-hint10 EndpointCostLookup[15215]: 20101222164718-28408:answers=4;time=0.024;content={"data":{"map":{"202.178.*.*":{"202.208.*.*":"512","218.219.*.*":"512","202.178.*.*":"16","210.191.*.*":"512"},"cost-mode":"numerical","cost-type":"routingcost"},"type":"endpoint_cost_map","meta":{"status":{"code":1,"version":1}}} Dec 22 16:47:56 p2p-hint10 EndpointCostLookup[15215]: 20101222164756-53965:EndpointCostLookup from REMOTE_ADDR=202.178.*.*,src=202.178.*.*,dst=218.219.*.*, 210.191.*.*, 202.208.*.*, 202.178.*.*

5.2.2 ログの書式

access-log および **error-log** は一般的な Apache httpd のログと同じである。Apache の公式ウェブサイト (<http://www.apache.org/>) を参照のこと。

peerselection.log はヒントサーバ独自のフォーマットであり、全てのリクエストおよびレスポンスの電文を書き出している。（高負荷の要因でもあるので、高負荷運用時はログの出力を減らすことを考慮したほうが良い）

peerselection.log は表 5.2-2 のサンプルのとおり、表 5.2-3 のような値の羅列である。

表 5.2-3 ログパラメータ

Timestamp: 処理をした時間（ヒントサーバの localtime）
Hostname: ヒントサーバの hostname
Command: ヒントサーバにリクエストされた API の種別
[session ID]: セッション ID
answers: レスポンス時に計算したコストの数。
time: レスポンスに要した時間（秒）。
content: レスポンス本文。
REMOTE_ADDR: アクセスのあったクライアントの IP アドレス。
src: コスト計算起点アドレスとしてクライアントから指定のあった IP アドレス。
dst: コスト計算終点アドレスの一覧。

別紙 1. データベース初期構築 SQL (file: geoloc_create.sql)

```
CREATE TABLE `cost_map` (  
  `src` bigint(20) unsigned NOT NULL,  
  `dst` bigint(20) unsigned NOT NULL,  
  `cost` int(2) NOT NULL DEFAULT '0',  
  `note` text,  
  `src_as` int(4) unsigned DEFAULT '0',  
  `dst_as` int(4) unsigned DEFAULT '0',  
  `hops` int(2) unsigned DEFAULT '0',  
  `aspath` text,  
  PRIMARY KEY (`src`,`dst`),  
  KEY `src` (`src`),  
  KEY `dst` (`dst`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `geoip` (  
  `start_inet` int(4) unsigned NOT NULL,  
  `end_inet` int(4) unsigned NOT NULL,  
  `country` char(8) NOT NULL,  
  `region` int(8) NOT NULL,  
  `updated` int(8) unsigned NOT NULL,  
  PRIMARY KEY (`start_inet`,`end_inet`),  
  KEY `start_inet` (`start_inet`),  
  KEY `end_inet` (`end_inet`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TABLE `link_map` (  
  `id` int(4) unsigned NOT NULL AUTO_INCREMENT,  
  `src_as` int(4) unsigned NOT NULL DEFAULT '0',  
  `dst_as` int(4) unsigned NOT NULL DEFAULT '0',  
  `link_cost` int(2) unsigned NOT NULL DEFAULT '0',  
  `flag` int(2) unsigned NOT NULL DEFAULT '0',  
  PRIMARY KEY (`id`),  
  KEY `src_as` (`src_as`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=3866 DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `pid_map` (  
  `start_inet` int(4) unsigned NOT NULL,  
  `end_inet` int(4) unsigned NOT NULL,  
  `masklen` int(1) unsigned NOT NULL,  
  `asnum` int(4) unsigned NOT NULL,  
  `flag` int(2) unsigned NOT NULL DEFAULT '0',  
  `suffix` int(2) unsigned NOT NULL DEFAULT '0',  
  `pid` bigint(20) unsigned NOT NULL,  
  `superhint` text,  
  PRIMARY KEY (`start_inet`,`masklen`),  
  KEY `start_inet` (`start_inet`),  
  KEY `end_inet` (`end_inet`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `prefix_as` (  
  `prefix` char(18) DEFAULT NULL,  
  `start_inet` int(4) unsigned NOT NULL,  
  `end_inet` int(4) unsigned NOT NULL,  
  `masklen` int(1) unsigned NOT NULL,  
  `asnum` int(4) unsigned NOT NULL,  
  `aspath` char(255) DEFAULT NULL,  
  `aspathlen` int(2) unsigned DEFAULT NULL,  
  `updated` int(8) unsigned NOT NULL,  
  PRIMARY KEY (`start_inet`,`masklen`),  
  UNIQUE KEY `prefix` (`prefix`),  
  KEY `start_inet` (`start_inet`),  
  KEY `end_inet` (`end_inet`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE `prop_map` (  
  `start_inet` int(4) unsigned NOT NULL,  
  `end_inet` int(4) unsigned NOT NULL,  
  `masklen` int(1) unsigned NOT NULL,  
  `prefix` char(18) DEFAULT NULL,
```

```
`aspath` char(255) DEFAULT NULL,  
`aspathlen` int(2) unsigned DEFAULT NULL,  
`updated` int(8) unsigned NOT NULL,  
`note` text,  
`prop_id` int(4) unsigned NOT NULL,  
PRIMARY KEY (`start_inet`,`masklen`),  
UNIQUE KEY `prefix` (`prefix`),  
KEY `start_inet` (`start_inet`),  
KEY `end_inet` (`end_inet`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `prop_master` (  
  `prop_id` int(4) unsigned NOT NULL,  
  `prop_value` text NOT NULL,  
  `prop_flag` int(8) unsigned NOT NULL DEFAULT '0',  
  `prop_name` text,  
  PRIMARY KEY (`prop_id`),  
  KEY `prop_id` (`prop_id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `region_code` (  
  `region` int(8) NOT NULL,  
  `name` char(16) DEFAULT NULL,  
  PRIMARY KEY (`region`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```