

Unboundの最適化

日本Unboundユーザ会 滝澤隆史

2011-03-04

日本Unboundユーザ会 OSC 2011 Tokyo/Spring発表資料

Unboundの最適化

最適化に関する文書

- 「Howto optimize」
 - http://unbound.net/documentation/howto_optimise.html
- 邦訳「最適化の方法」
 - http://unbound.jp/unbound/howto_optimise/

設定パラメータ

スレッド数

- num-threads
 - スレッド数を設定する
- 最適化
 - システムのCPUのコア数と同じくする

スラブ数

- msg-cache-slabs, rrset-cache-slabs, infra-cache-slabs, key-cache-slabs
 - キャッシュ内のスラブ数
 - スレッドによるロックの競合を減らす
- 最適化
 - num-threadsの値に近くなる2の累乗に設定

キャッシュサイズ

- msg-cache-size
 - メッセージのキャッシュサイズ
- rrset-cache-size
 - RRsetのキャッシュサイズ
- 最適化
 - rrset-cache-sizeはmsg-cache-sizeの2倍に設定する。
 - 総メモリ使用量は総キャッシュメモリ量の2~2.5倍消費する

オープンするポート数

- outgoing-range
 - オープンするポート数 (スレッド毎)
- 最適化
 - 可能な限り最大にする
 - ファイル記述子1024個の制限
 - 1024/コア数 - 50
 - 1コア: 950
 - 2コア: 450
 - 4コア: 200
 - libevent/libevを使うことで制限解除

スレッド毎に対応するクエリー数

- num-queries-per-thread
 - スレッド毎に対応するクエリー数
- 最適化
 - outgoing-rangeの半分に設定する

受信バッファサイズ

- so-rcvbuf
 - 受信バッファサイズ
 - SO_RCVBUFソケットオプション
- 最適化
 - 4mを設定する
 - あるいは、システムの受信バッファを設定
 - Linux: /proc/sys/net/core/rmem_max

送信バッファサイズ

- so-sndbuf (1.4.8以降)
 - 送信バッファサイズ
 - SO_SNDBUFソケットオプション
- 最適化
 - 4mを設定する
 - あるいは、システムの送信バッファを設定
 - Linux: /proc/sys/net/core/wmem_max

TTLの制限

- cache-max-ttl
 - キャッシュ内のRRsetのTTLの最大値
 - デフォルト: 86400
- cache-min-ttl
 - キャッシュ内のRRsetのTTLの最小値
 - デフォルト: 0

DNSKEYレコードのプリフェッチ

- prefetch-key (1.4.2以降)
 - DSがあれば、DNSKEYを事前に取得
 - 遅延が減る
 - DNSSECの検証を有効にしているときには効果的
- 最適化
 - prefetch-key: yes

ビルド オプション

libevent/libev

- libevent/libevを使うことにより、ファイル記述子1024個の制限を回避
- ビルド時に指定する
 - `./configure --with-libevent`
- `outgoing-range`を8192に、`num-queries-per-thread`を4096に設定して問題なく運用しているとの報告あり。

フォーク

- スレッドの代わりにプロセスをフォークする方法
- ビルド時に指定
 - `./configure --without-pthreads --without-solaris-threads`
- ロッキング不要のため10～20%速度向上
- プロセス間ではキャッシュを共有しないことに注意

終わり