

DNSサーバパフォーマンス評価

ETJP DNS-WGまとめ+

N+I2005 DNSホットトピックス

2005/6/10

藤原和典

株式会社日本レジストリサービス(JPRS)

fujiwara@jprs.co.jp

ENUMとは

- 電話番号を用いてインターネット上の様々な通信サービスへの統一的なアクセスを可能にするプロトコル

- RFC3761で定義

- 電話番号をドメイン名に対応 (.e164.arpa)
 - +81-3-1234-5678 (E.164電話番号)
 - +81312345678 (AUS)
 - 8.7.6.5.4.3.2.1.3.1.8.e164.arpa (ドメイン名)

- インターネットのサービスをURIとして登録 (NAPTR RR)
 - IN NAPTR 100 0 "u" "E2U+sip" "!^.*\$!sip:810123456789@example.jp!" .

ENUM Trial Japan (ETJP)

- ENUMに関する技術実験を行うことを目的に設立

- 基本機能と実用性の技術的検証
 - DNSによる基盤サービス
 - 通信アプリケーション
 - 通信サービス

- サービス化に向けた技術的課題の整理と検討

- <http://etjp.jp/>

ETJP DNS-WG:概要と目的

概要

- トライアルおよび将来のENUMの商用利用に向けた基礎データ収集を目的
- 日本国内で展開しうるENUM DNSのモデル
 - 定義
 - 要求仕様
 - 評価基準
 - 現在のDNS実装を性能評価
- DNSSECのENUMへの適用について検討と評価

活動の成果

- ENUM DNSに関するモデル・要求仕様
- DNSサーバ評価結果
- DNSSECのENUMへの適用についての調査結果

ENUM DNS要求条件

□背景

- ENUMの商用利用の形態をモデル化
- 日本の主要な電話サービスのトラフィック情報などの数値は公開されている
 - ▶テレコムデータブック2004

□DNS-WGで想定するENUMモデル

- 電話サービスのアドレス解決にENUMを用いる場合を想定
- 電話契約数、通話数をもとに必要条件を決定

□登録数

- 系全体で1～2億件程度(日本の電話番号は約1億6000万)

□DNSパフォーマンス条件

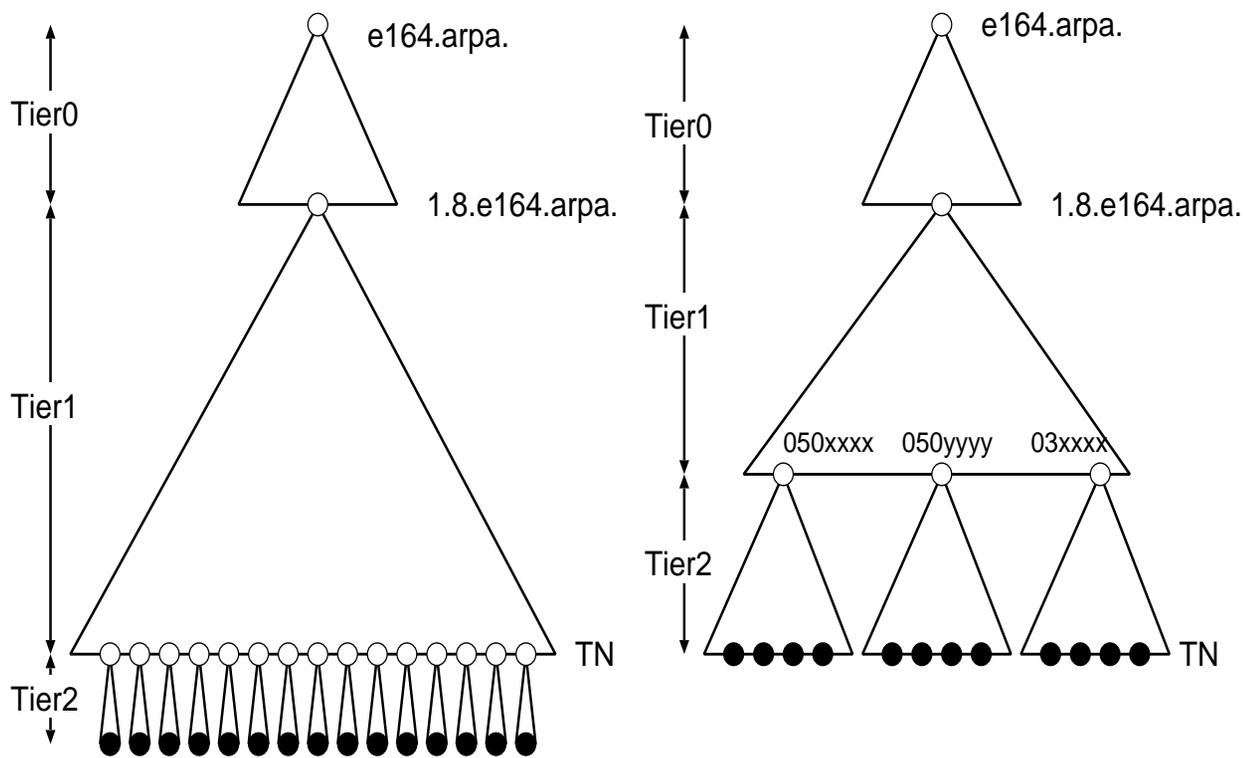
- 系全体で5万qps程度(電話の平均発呼数の10倍)
 - ▶平成14年度の固定電話(公衆電話含む)、携帯電話、PHSの総通信回数 132,391(百万回)
 - ▶固定電話 最繁時(9時～10時)の発呼数は6,484(百万回)=約4,934呼/秒

□DNSデータ更新頻度条件

- 変更までに30分以内

想定するENUM DNS Tier構造

ENUM研究グループ報告書 23,24ページ (2)と(3)を検討



○ : NS Record

● : NAPTR Record

(2)

(3)

(2)たけのこモデル

(3)割り当て単位モデル

DNSサーバ性能評価

- DNS Authoritativeサーバ単体での能力測定

- 共通のテストデータ
 - たけのこモデル、割り当て単位モデル
 - Tier1、Tier2 それぞれ
 - ▶ ただし、割り当て単位モデルTier1は件数が少ないため、たけのこモデルTier1で代用
 - パラメータで電話番号数を指定
 - ▶ データが主記憶に入り切るサイズのデータ
 - ▶ 主記憶の量により、測定できるエントリ数が決まる

- 測定項目
 - DNSデータロード時間・リロード時間
 - メモリ使用量(プロセスサイズ)
 - サーバ応答性能(queryperfを使用)

テストデータ(1)

□ たけのこ Tier1

- 機械的に生成するゾーンファイル
- エントリ数を指定して生成
- 一つの番号あたり、NS RR 2行

9.8.7.6.5.4.3.2.1.0.1.8.e164.arpa. IN NS ns1.etjp.jp.

9.8.7.6.5.4.3.2.1.0.1.8.e164.arpa. IN NS ns2.etjp.jp.

□ たけのこ Tier2

- 機械的に生成する多数のゾーンファイル
- 一つの番号にNAPTR RR 1行, NS RR 2行
- ゾーン数を指定して生成
- named.confも生成

\$ORIGIN 9.8.7.6.5.4.3.2.1.0.1.8.e164.arpa.

IN SOA

IN NS ns1.etjp.jp.

IN NS ns2.etjp.jp.

IN NAPTR 100 0 "u" "E2U+sip" "!.^.*\$!sip:810123456789@example.jp!" .

テストデータ(2)

□局番単位 Tier1

- 機械的に生成するゾーンファイル
- 局番数を指定、15万エントリ程度
- 一つの局番あたり、NS RR 2行

```
$ORIGIN 1.8.e164.arpa.  
@ IN SOA ns0.etjp.jp. postmaster.etjp.jp. (1 1H 5M 7D 10M)  
IN NS ns1.etjp.jp.  
IN NS ns2.etjp.jp.  
0.0.0.0.0.0 IN NS ns1.isp000.jp.  
0.0.0.0.0.0 IN NS ns2.isp000.jp.  
1.0.0.0.0.0 IN NS ns1.isp001.jp.
```

...

□局番単位 Tier2

- 機械的に生成する多数のゾーンファイルとnamed.conf
- 各ゾーンファイルに1万番号
 - ▷1番号ごとにNAPTR RR 1行
- ゾーン数を指定して生成

```
$ORIGIN 0.0.0.0.0.1.8.e164.arpa.  
$TTL 120  
@ IN SOA ns1.etjp.jp. postmaster.etjp.jp. (1 1H 5M 7D 10M)  
IN NS ns1.etjp.jp.  
IN NS ns2.etjp.jp.  
0.0.0.0 IN NAPTR 100 0 "u" "E2U+sip" "!^.*$!sip:00000000@sipisp.jp!" .  
1.0.0.0 IN NAPTR 100 0 "u" "E2U+sip" "!^.*$!sip:00000001@sipisp.jp!" .  
2.0.0.0 IN NAPTR 100 0 "u" "E2U+sip" "!^.*$!sip:00000002@sipisp.jp!" .
```

...

DNS-WG: 評価基準

- 要件を満たす実装と、その仕組み(組合せ)の提案を目指す
 - 各要素を実用的に実現する方法
 - DNS Authoritative サーバ単体での性能測定
 - Authoritative サーバを何台に分割すると要求を満たせるか
 - 実用台数組み合わせで実現できるモデル
 - ▶運用可能な実用的なシステム数を10から100と想定

DNS-WG: 調査対象DNSサーバ

- BIND 8: 8.3.7
 - ISCが開発、従来の標準
 - Authoritative server機能とFull Resolver機能

- BIND 9: 9.3.1
 - ISCが参照実装として開発
 - DNSSEC対応
 - Authoritative server機能とFull Resolver機能

- NSD: 2.3.0
 - NL NetLabsが開発
 - DNSSEC対応
 - Authoritative server機能のみ

- djbdns: 1.0.5
 - D. J. Bernstein助教授が開発
 - Authoritative server機能(tinydns)とFull Resolver機能(dnscache)を分離

DNSサーバの特徴 BIND, NSD

□BIND 8, BIND 9 共通点

- ゾーンファイル、制御ファイルは人間が読み易いテキスト
- ゾーンごとに1ファイル
- named.conf にゾーン情報を記述
- namedはnamed.confに従い、ゾーンファイルを直接解釈

□BIND 8

- ゾーンデータ読み込み、再読み込み時には無応答

□NSD

- BIND形式のゾーンファイルからnsd.dbを作成
 - ▷ zonecコマンドでゾーンファイルをコンパイル
 - ▷ nsd.zones: zone <ゾーン名> <ゾーンファイル名>
- nsdサーバプログラムが読むファイルは独自内部形式のnsd.db
- データロード時には zonec実行時間と、nsd.db読み込み時間が含まれる

DNSサーバの特徴 djbdns

□djbdns(tinydns)

- 独自のテキスト形式 (1文字コマンド + データ)
- 複数ゾーンがあっても1ファイル
- tinydns-data コマンドで、data.cdb に変換
- tinydnsは、queryのたびに data.cdb をmmapし、検索
- そのため、データセットの切替えは瞬時
- データロード時間は tinydns-data コマンド実行時間
- メモリ使用量は data.cdb のサイズ

□tinydnsパフォーマンス向上パッチ

- 毎回mmapでは遅く、mmap頻度を1秒に一度にしても実害はない
- mmap頻度を下げ、応答性能を上げる非公式パッチを適用したもので評価
- Lennert Buytenhek氏作成

▷ <http://people.FreeBSD.org/~roam/ports/patches/dns/tinydns-persistmmap-20040418.patch>

□NAPTR/SRV patch

- Guilherme Balena Versiani氏作成

▷ <http://mywebpage.netscape.com/guibv/#djb>

測定環境

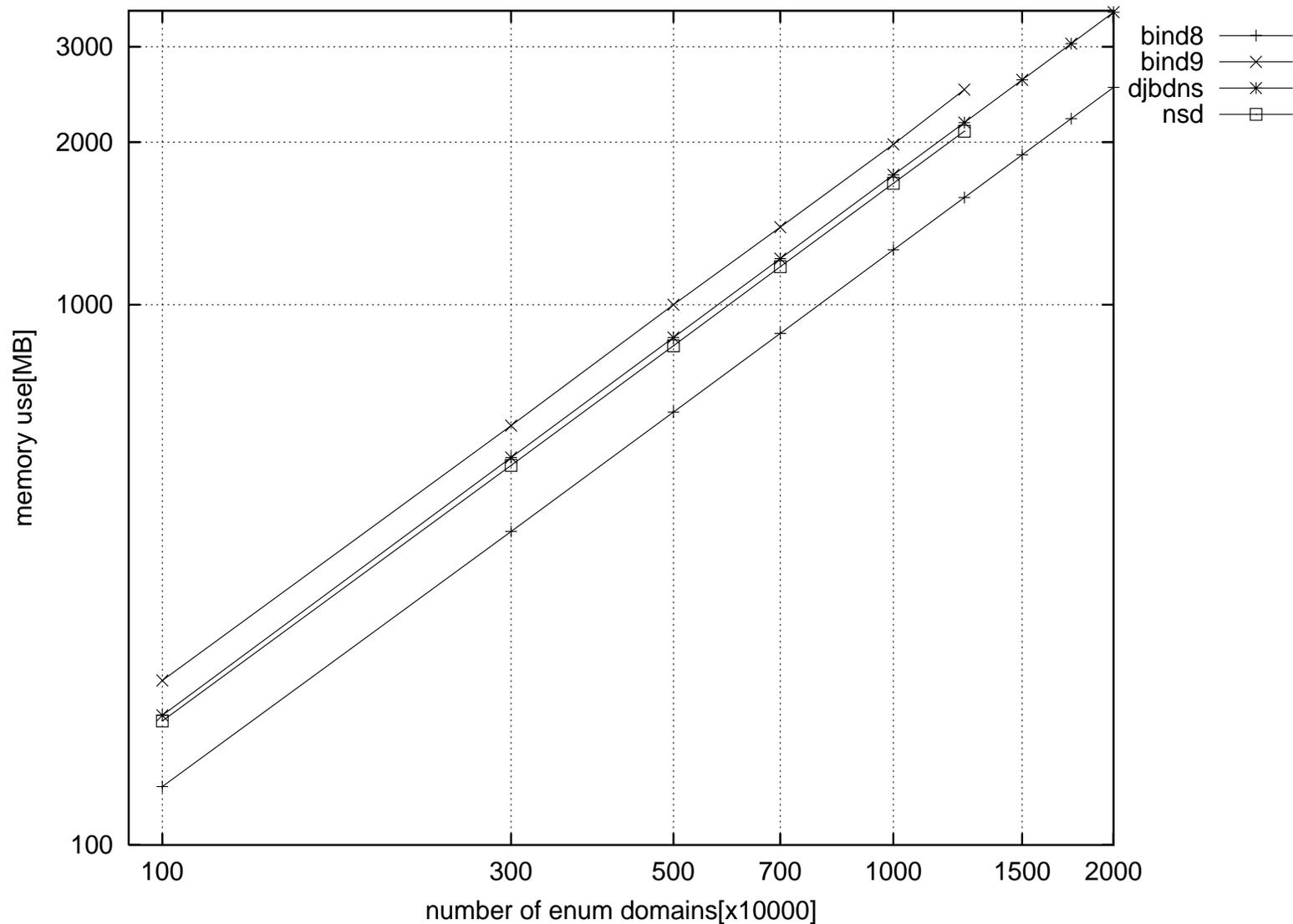
□DNSサーバPC-----Ethernet-----測定用PC

- 測定対象DNSサーバPC 1台
- 測定用PC 1台
- 同一ネットワークに接続

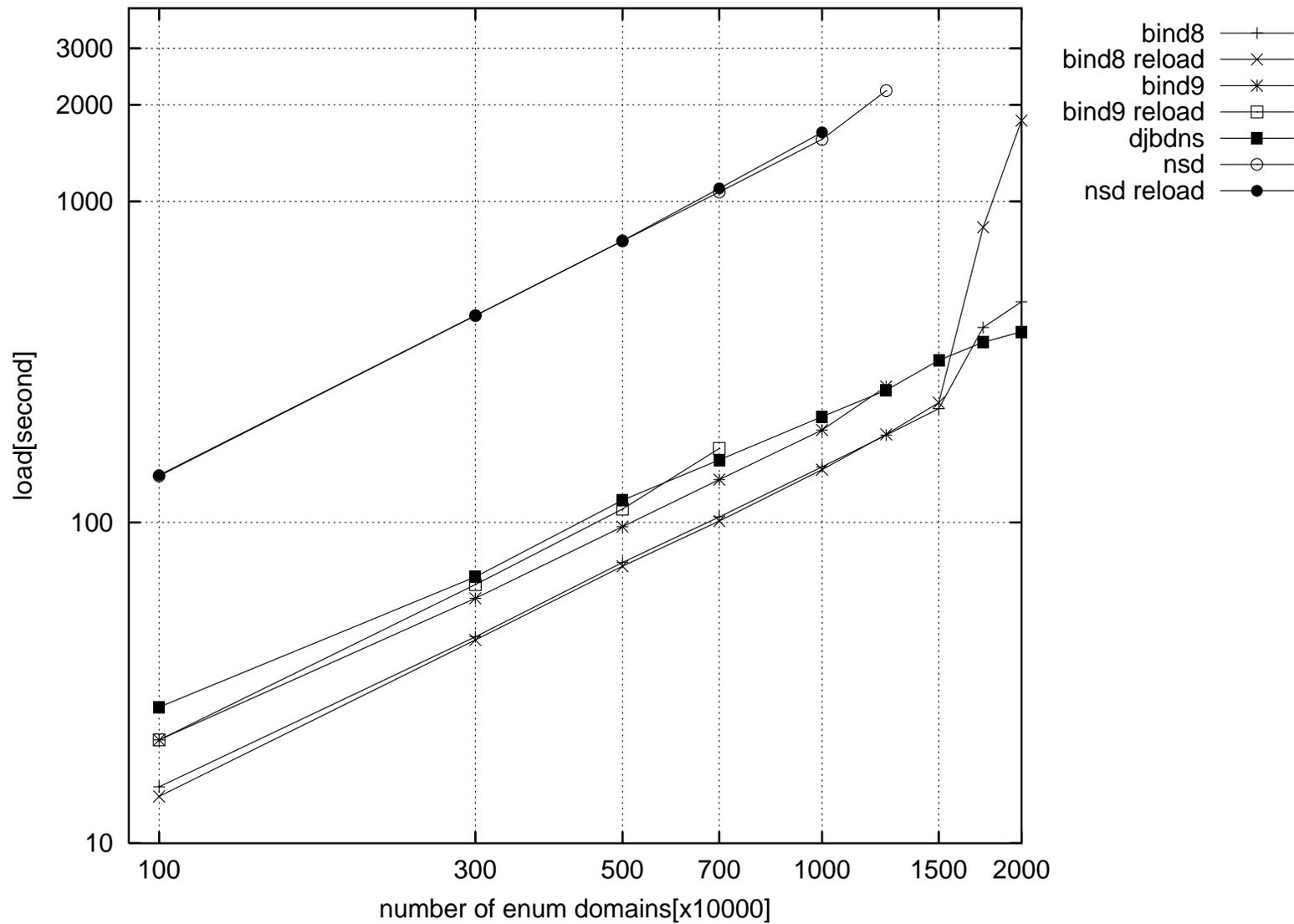
□テスト環境

- DNSサーバPC
 - ▷Pentium4-3GHz, memory 2.5GB, FreeBSD 4.11
- queryperf PC
 - ▷Pentium4-2.7GHz, FreeBSD 4.11
- Ethernet
 - ▷1000baseT

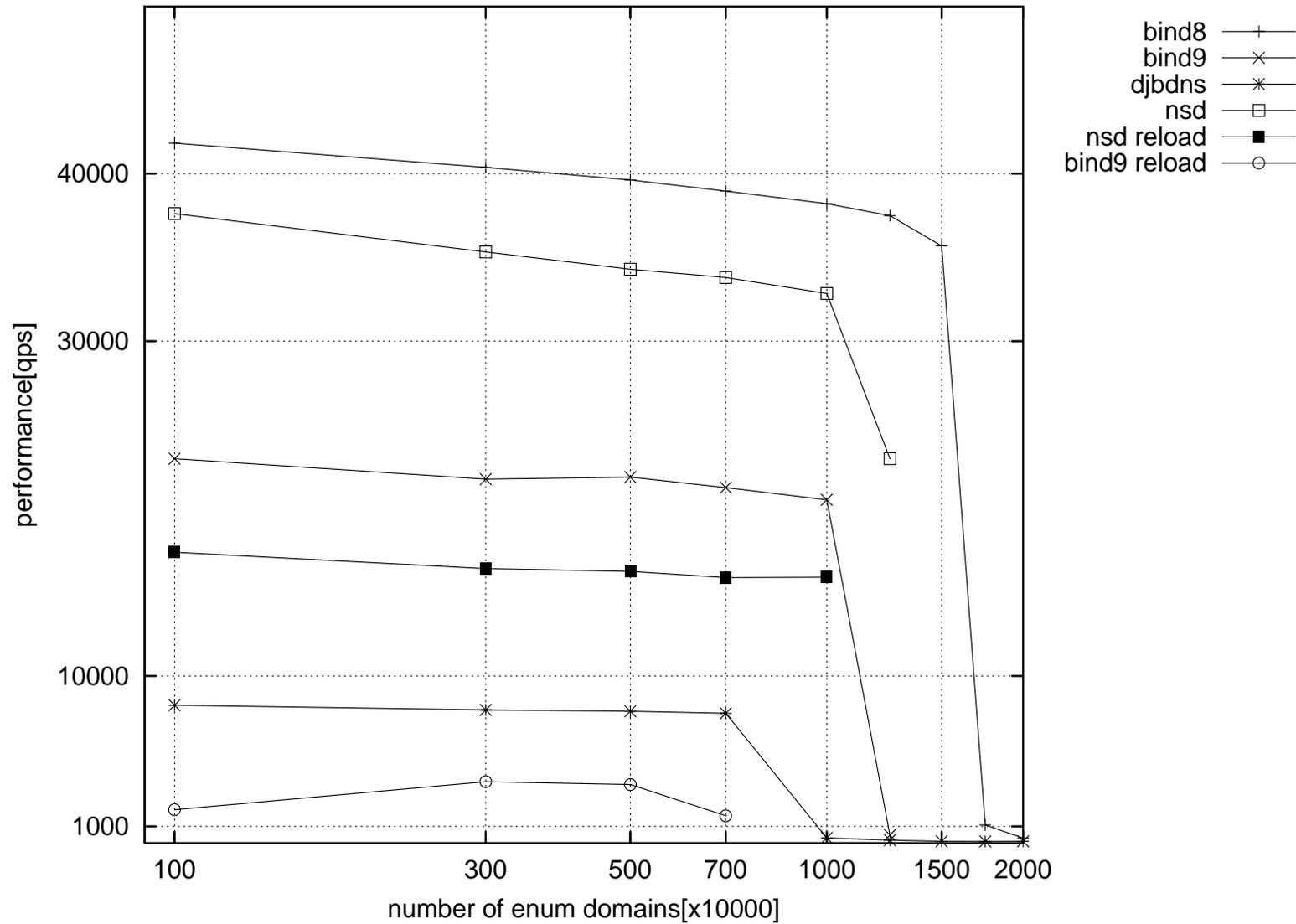
たけのこモデルTier1 メモリ使用量



たけのこモデルTier1 ロード時間



たけのこモデルTier1 応答性能



たけのこモデルTier1 評価結果(1)

□登録数

- どのサーバでも 1000万番号 2000万エントリまでは正常動作
 - ▷20システムにわけること2億対応可
 - ▷BIND 9ではreload時にメモリが不足した

□サーバパフォーマンス

- 1000万エントリで、NSDでは32000qps、BIND 8では38000qps、BIND 9では20000qps
- サーバを2,3台用いることで5万qps達成
- djbdnsは性能がでない

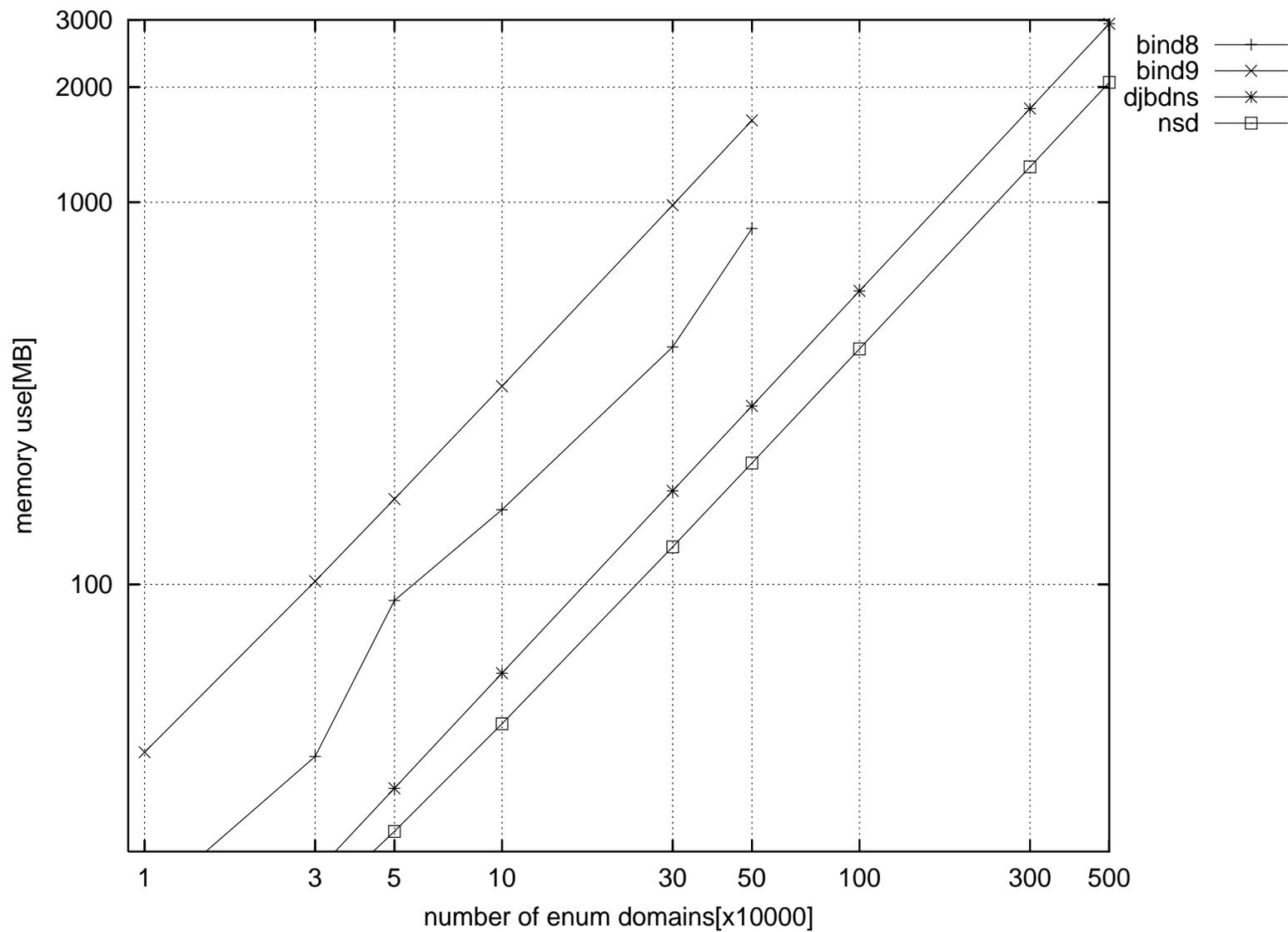
□更新頻度条件

- NSDの場合はデータのコンパイル・ロードだけで30分かかる
- BIND 8, BIND 9, djbdnsの場合は5分弱でロード可能

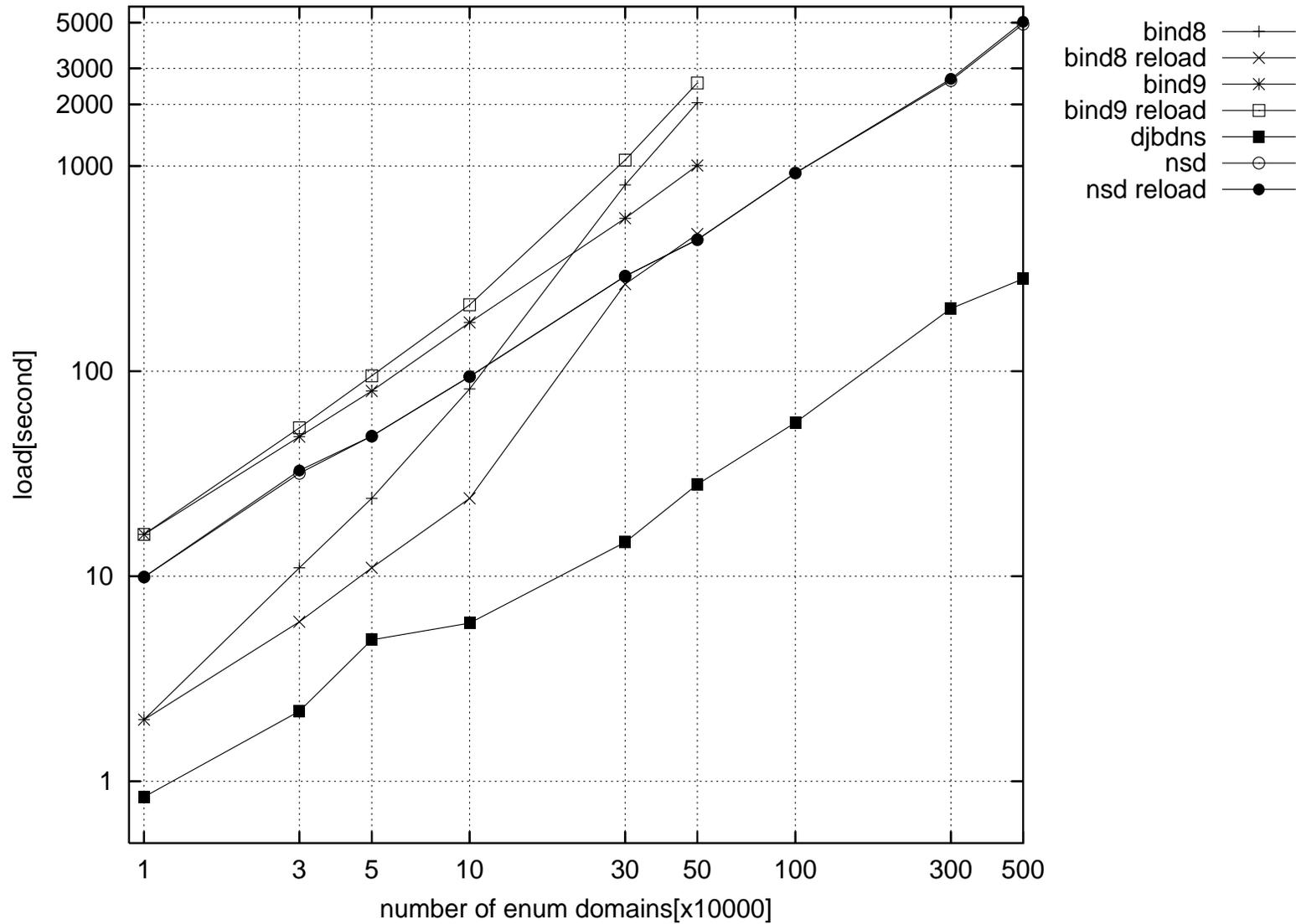
□まとめ

- BIND 8, BIND 9を3台用いたシステムで1000万番号保持
- 20システム配置することで2億番号対応可能

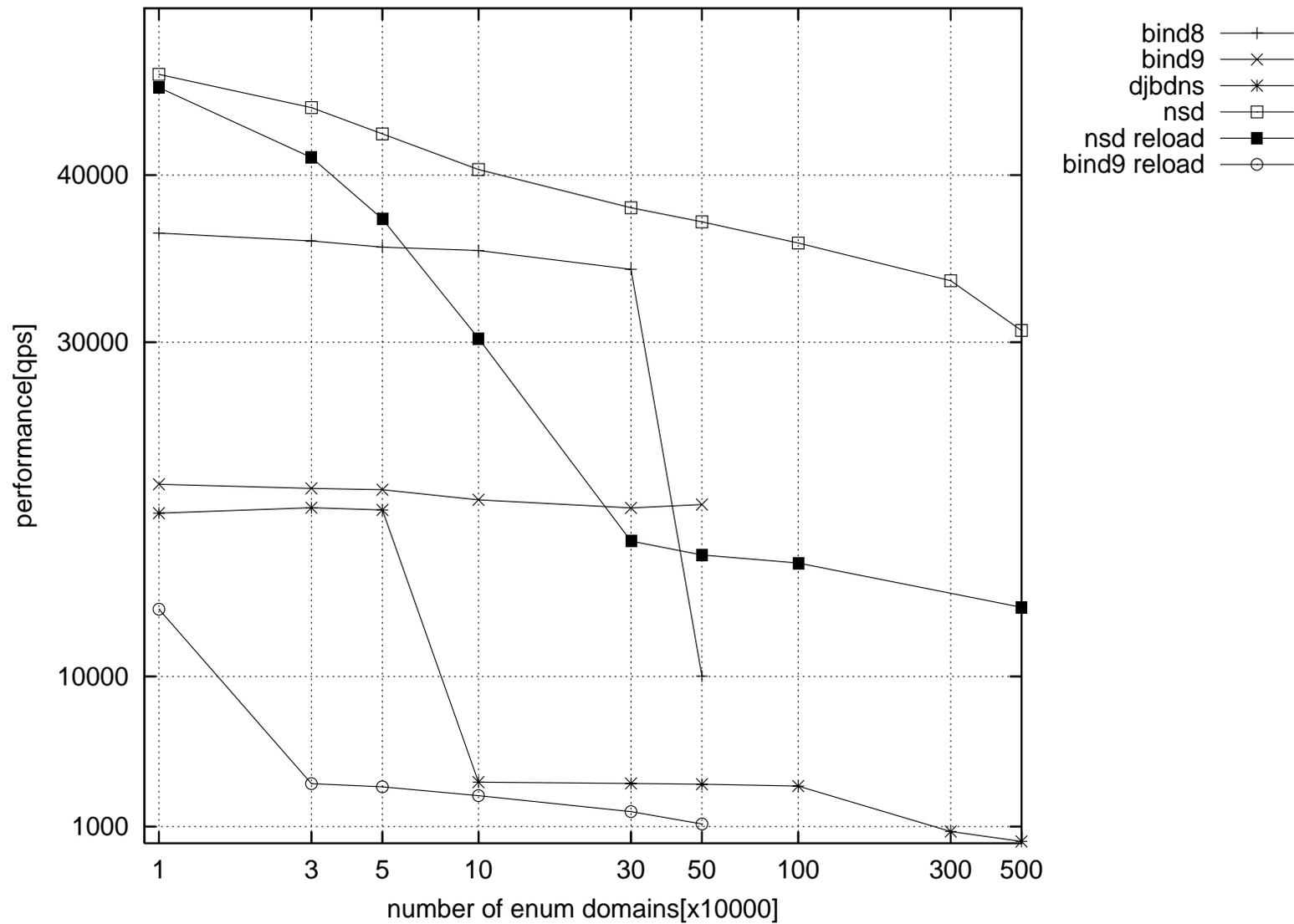
たけのこモデルTier2 メモリ使用量



たけのこモデルTier2 ロード時間



たけのこモデルTier2 応答性能



たけのこモデルTier2 評価

□登録数(保持可能数)

- BIND 8, BIND 9では50万
- NSD, djbdnsでは500万

□サーバパフォーマンス

- NSD(500万番号)、BIND 8(30万)は30000qps以上
- BIND 9(50万番号)では20000qps
- djbdns(100万番号)では3400qps、300万番号では700qps

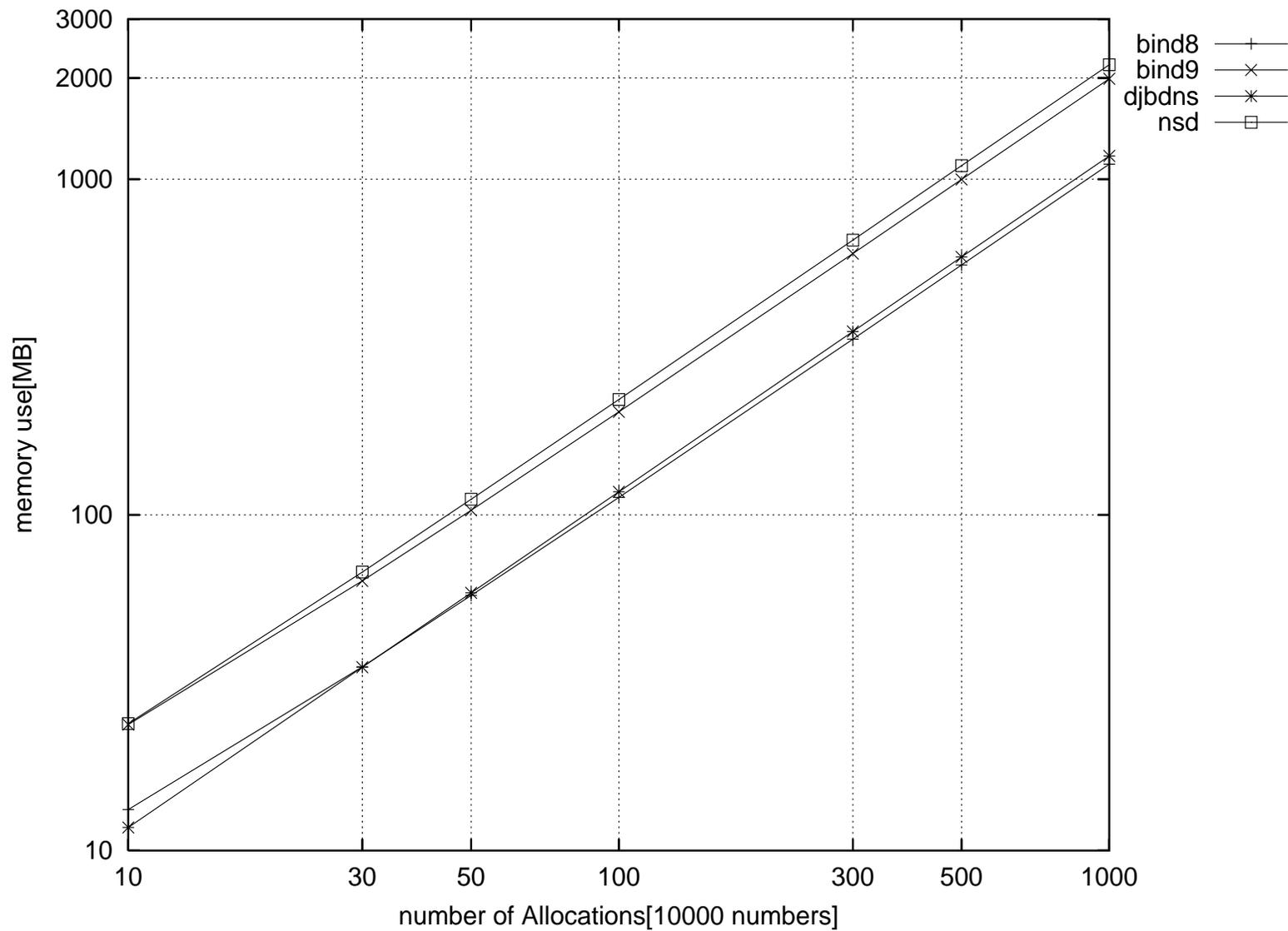
□更新頻度条件

- NSDでは100万ドメイン名で1000秒、500万で5000秒
- BIND 8, 9は30万ゾーンの読み込みに20分程度
- djbdnsのデータファイル変換の時間がもっとも小さい
 - ▶100万番号で1分、300万番号で3分、500万番号で5分

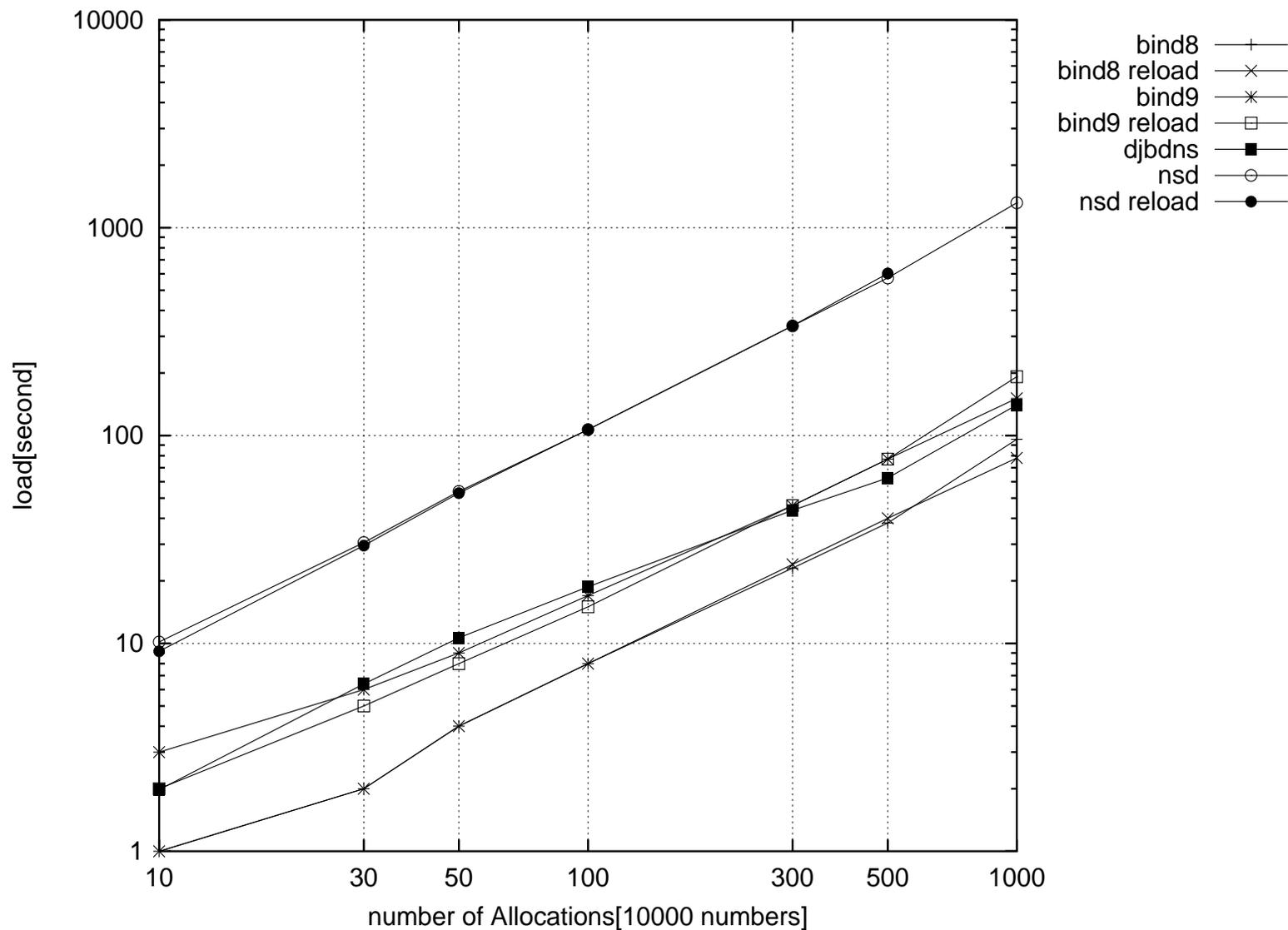
□まとめ

- djbdnsを用い、1システムで300万番号保持する
- すべての番号帯に均等に問い合わせがくる場合、全体の300万/1億6000万の問い合わせ = 1000qps
- 300万番号ごとに2台のサーバを用い、67システム配置すれば2億番号対応可能

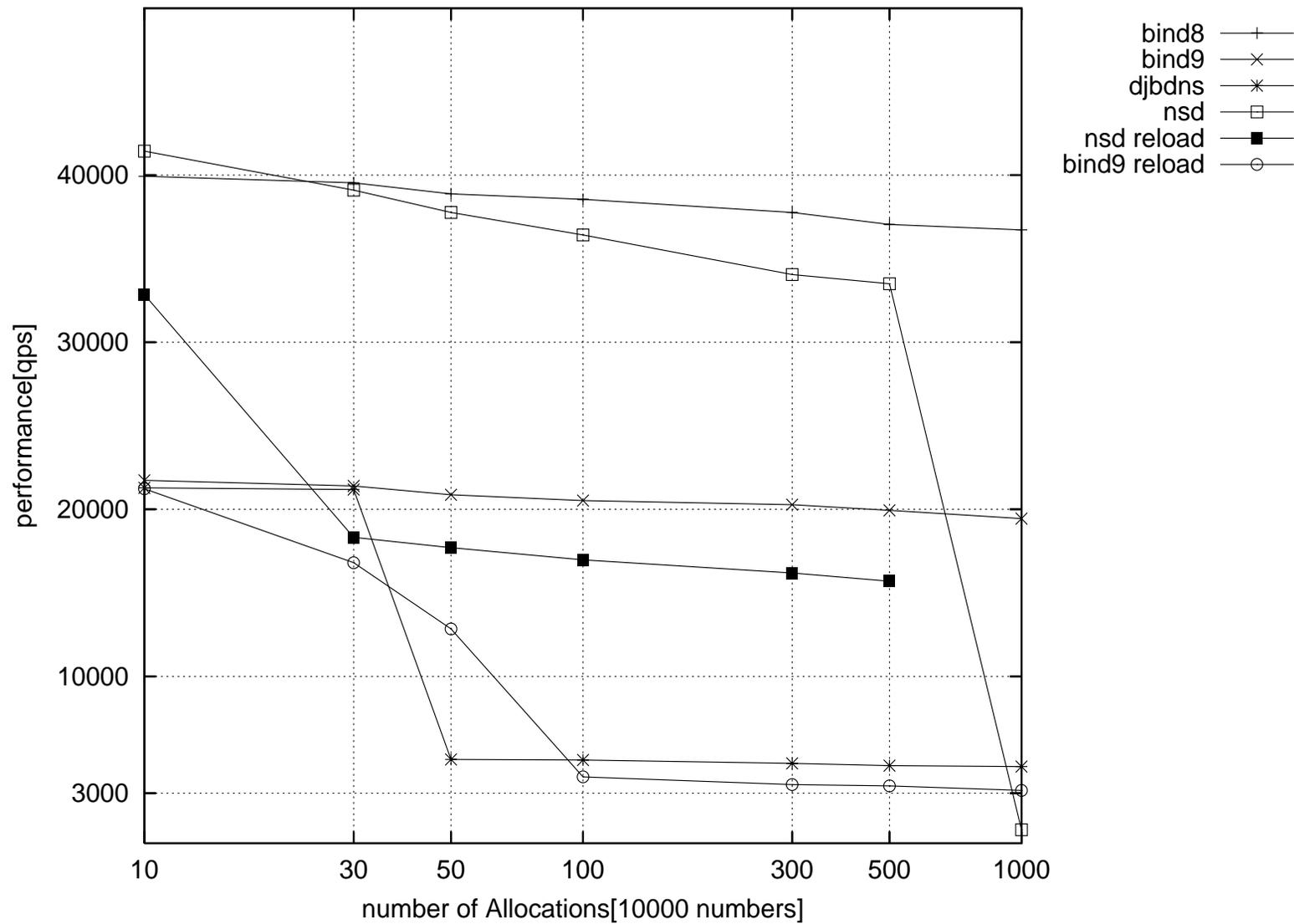
割当単位モデルTier2 メモリ使用量



割当単位モデルTier2 ロード時間



割当単位モデルTier2 応答性能



割り当て単位モデルTier2 評価

□登録数

- 20000ゾーンで2億番号
- どのサーバでも1万エントリのゾーンを1000ゾーン蓄積できた

□サーバパフォーマンス

- BIND 8は1000ゾーンまで3万qps以上の性能を示した
- NSD, BIND 9は500ゾーンまで(メモリの制約)
- djbdnsはほぼ一定であるが、他に比べ低い

□更新頻度条件

- BIND 8, 9は比較的短時間で読み込むことができる
- djbdnsのデータファイル変換の時間が小さい
- NSDはゾーンのコンパイル、読み込みに他の実装の10倍程度の時間がかかる

□まとめ

- BIND 8で1000ゾーンを運用する場合、1システムを3台のマシンで構成し、20システムで構成可能
- BIND 9で500ゾーンを運用する場合、1システムを3台のマシンで構成し、40システムで構成可能

ENUM DNS評価・結論

- たけのこモデルTier1
 - BIND 8, BIND 9を3台用いたシステムで1000万番号保持
 - 20システム配置することで2億番号対応可能
 - 合計60台
- たけのこモデルTier2
 - djbdnsを用い、1システムで300万番号保持、2台のサーバ
 - 67システム配置すれば2億番号対応可能
 - 合計134台
- 割り当て単位モデルTier1
 - 局番数 15万より、NSDのサーバを2台で構成可能
- 割り当て単位モデルTier2
 - BIND 8で1000ゾーンを運用する場合、1システムを3台のマシンで構成
 - 20システム配置することで2億番号対応可能
 - 合計60台

- たけのこモデルでも割り当て単位モデルでも実現可能

TLD(汎用jp)型ゾーンファイルの評価

□ ドメイン名リスト

- /usr/share/dict/web2から生成、文字を追加
- (平均ドメイン名長は実際のjpと似た値)

□ 各ドメイン名ごとにネームサーバ 2

□ 各ネームサーバごとに IPv4アドレス 1

\$ORIGIN jp.

jp. IN SOA a.dns.jp. postmaster.dns.jp. (1 1H 5M 7D 10M)

jp. IN NS a.dns.jp.

jp. IN NS b.dns.jp.

a.dns.jp. IN A 10.0.0.1

b.dns.jp. IN A 10.0.0.2

...

example.jp. IN NS ns1.example.jp.

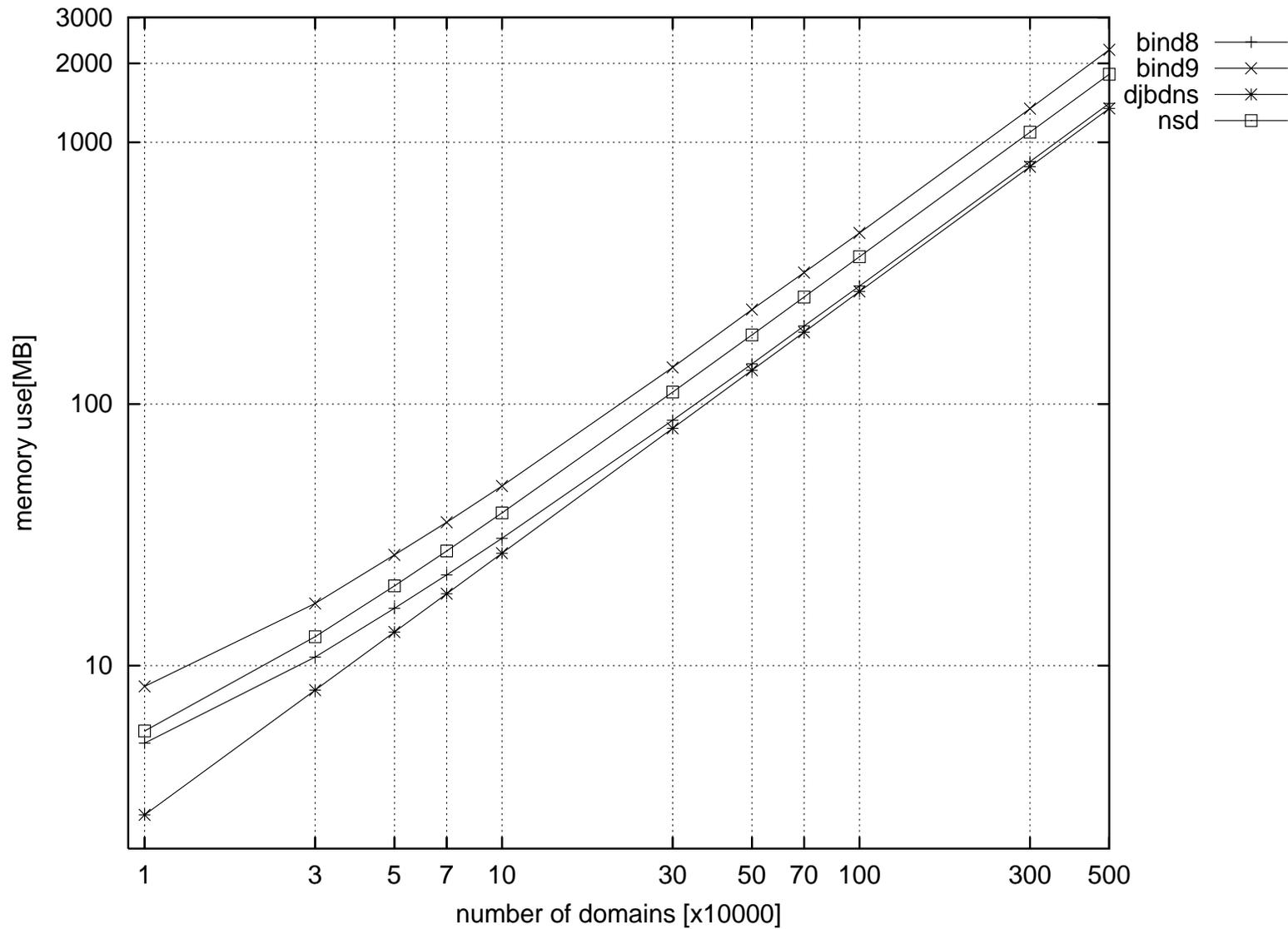
example.jp. IN NS ns2.example.jp.

ns1.example.jp. IN A 192.168.1.1

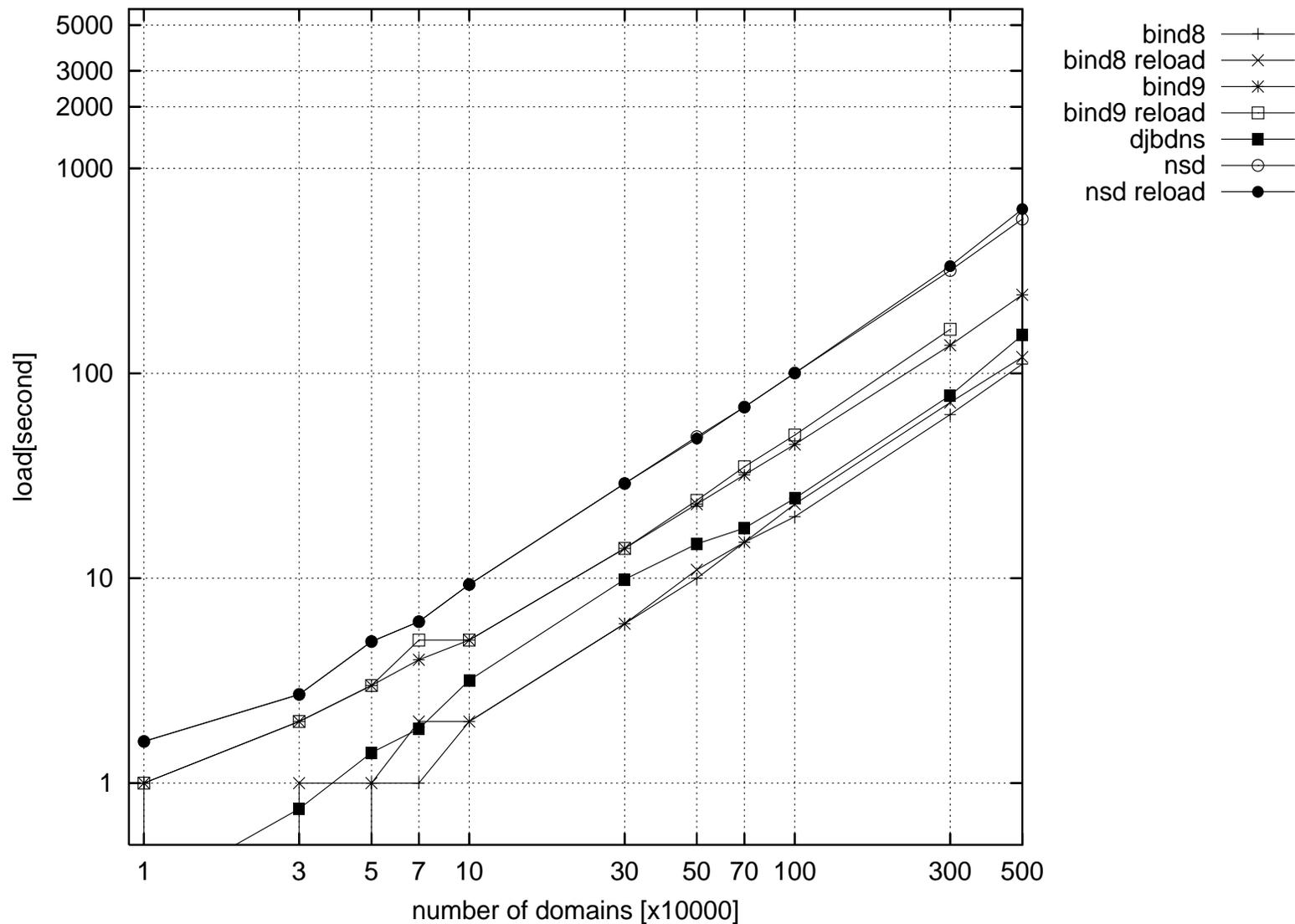
ns2.example.jp. IN A 192.168.1.2

...

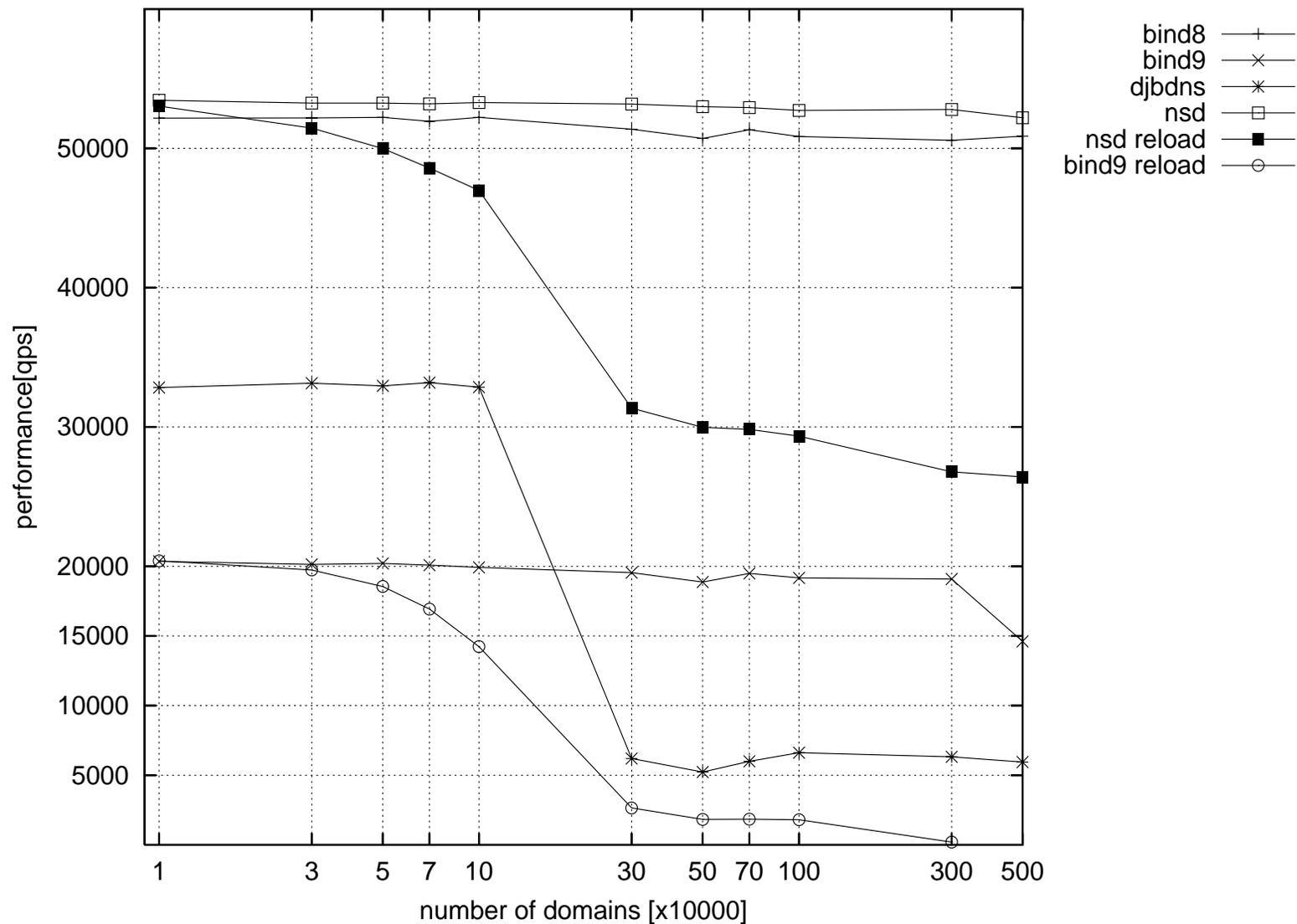
TLD(汎用JP)型 メモリ使用量



TLD(汎用JP)型 ロード時間



TLD(汎用JP)型 応答性能



TLD(汎用JP)型 評価

- メモリ使用量はドメイン名数にほぼ比例
 - ENUM型それぞれ、TLD型共に
 - 500万ドメイン名で32bit CPUの限界に達する

- ロード時間はドメイン名数にほぼ比例
 - 更新頻度を考えなければ500万ドメイン名でも動く
- リロード時間は、それぞれロード時間とほぼ同じ

- 応答性能
 - BIND 8, 9, NSDの場合、ドメイン名数が増えても、ほぼ一定(1割程度の劣化)
 - ▷メモリが足りなくなると急激に劣化する
 - djbdnsの場合、ドメイン名数が増えるに伴い、突然性能が劣化するが、劣化後も条件によっては有効な応答性能を示す
- リロード中の応答性能
 - NSDの再読み込み中の応答性能は定常時の50%
 - BIND 9の再読み込み中の応答性能は定常時の10%

その他 DNSサーバについての情報

□BIND 8

- DNSSECなしの場合、選択肢として考えられる
- 読み込み中は問い合わせに答えない(運用上の考慮が必要)

□BIND 9.3.1

- ロード、リロード後、新しいデータを答えるようになってから、しばらく内部で重い処理を行なっている
- その間は応答性能が悪い
- 内部処理が終るとよい性能がでる
- 今回は、CPU負荷が1%になるまで待ち、そのあと応答性能を評価した

- BIND 9.3.0は、ENUM型ドメイン名(. を多く含むドメイン名)の場合に応答性能が劣化したが、9.3.1で改善

その他 DNSサーバについての情報

□ djbdns (tinydns)

- 大量のデータを取り扱っていると応答性能は悪くなるが、データ変換時間は短く、サーバへのロード時間はゼロなので更新に強い
- 大量のゾーンを扱う場合、djbdnsでは一つのファイルで済む
 - ▶ BINDやNSDではゾーン数分のファイルを管理する必要あり
- 非公式パッチのチェックが必要

□ NSD

- よいパフォーマンス、長いデータ準備・ロード時間
- 周辺ツールにいくつか不具合があった
 - ▶ すでにpatchを送り、修正済み