

BINDマルチコア/プロセス パフォーマンステスト

2008/7/9

住商情報システム株式会社

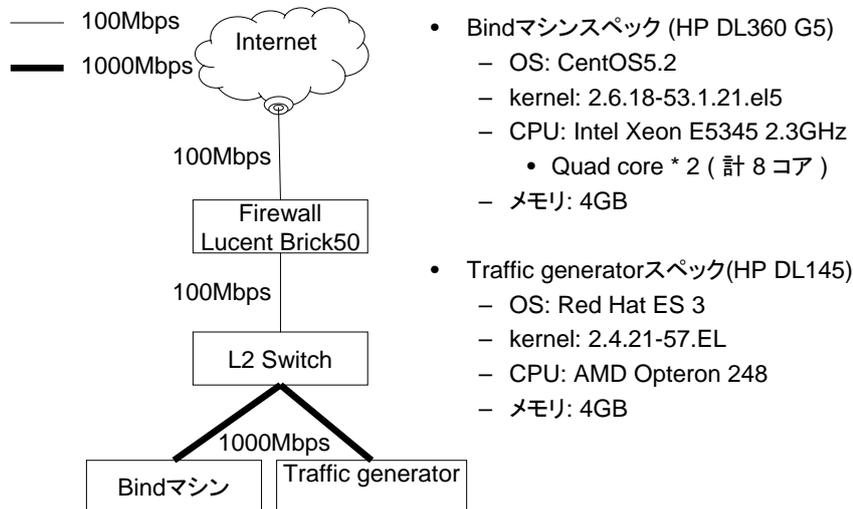
服部 成浩

s.hattori@scs.co.jp

テストをした背景と内容

- マルチコアの製品の低廉化
 - Bindはどのくらいパフォーマンスでるのか?
 - 神明さんパッチ
 - Nominum製品はマルチコア対応でない
- テスト内容
 - 2種類のテストを実施
 - テスト1: コア数と処理性能
 - テスト2: 1プロセス時と複数プロセス時の比較
 - セキュリティ機能: queryport-pool機能

テスト構成: ネットワーク構成図



テスト構成: テスト方法

- Bindテスト対象バージョン
 - 9.4.2 , 9.5.0
 - 9.5.0 use-queryport-pool有効
 - デフォルト設定:queryport-pool-ports 8
 - 9.5.0 use-queryport-pool無効
- トラフィックジェネレーター
 - Nominum resperf 1.0.1.0
 - http://www.nominum.com/services/measurement_tools.php
- クエリリスト(2パターン)
 - 全てユニークなクエリ
 - ただしトップレベル,セカンドレベルは重複しているクエリもある
 - 全てのリクエストに対し、キャッシュサーバは少なくとも1回は権威サーバに問い合わせる必要がある
 - すべて同じクエリ(100%キャッシュヒット)

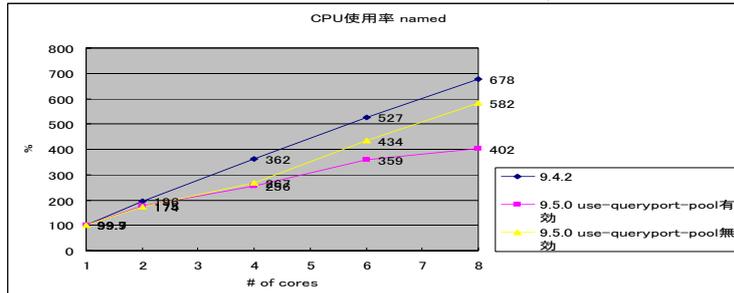
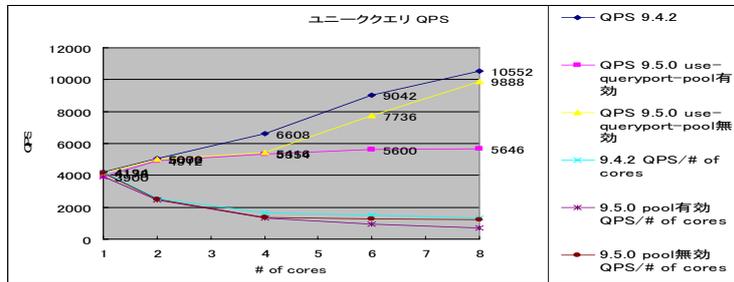
テスト構成: テスト方法

- Bind設定
 - キャッシュサイズ: 1,400MB
 - recursive-clients: 50,000
 - recursive-clientsは上限値に達しないように事前に確認し、上記値に設定
- Bindはソースからインストール
 - シングルコアプロセス
 - ./configure --prefix=xx;make;make install
 - マルチコアプロセス
 - ./configure --prefix=xx --enable-threads;make;make install
- Bindマシンはテスト毎にリブートを実施

テスト構成: テスト方法

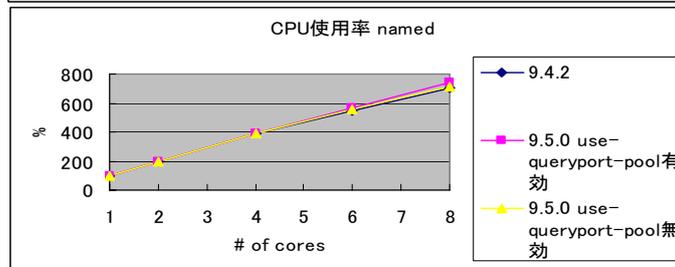
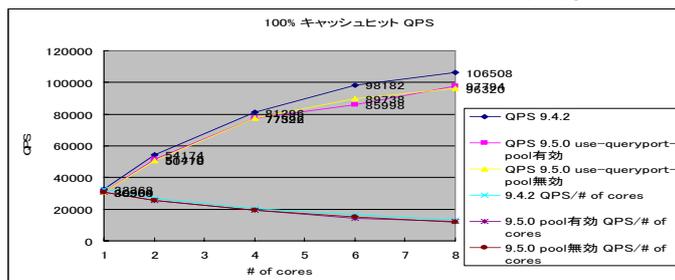
- resperfのパラメータ
 - ./resperf -s BindのIP -d query_list -m 20000
 - -m 20000: 60秒かけて20,000qpsまで負荷を上げる
- namedのリソースはtopコマンドでチェック
- カーネルに認識させるコア数の指定方法
 - grub.conf の maxcpus オプションで指定

ユニーククエリテスト結果



Sumisho Computer Systems Corp. All Rights Reserved.

100%キャッシュヒットテスト結果



Sumisho Computer Systems Corp. All Rights Reserved.

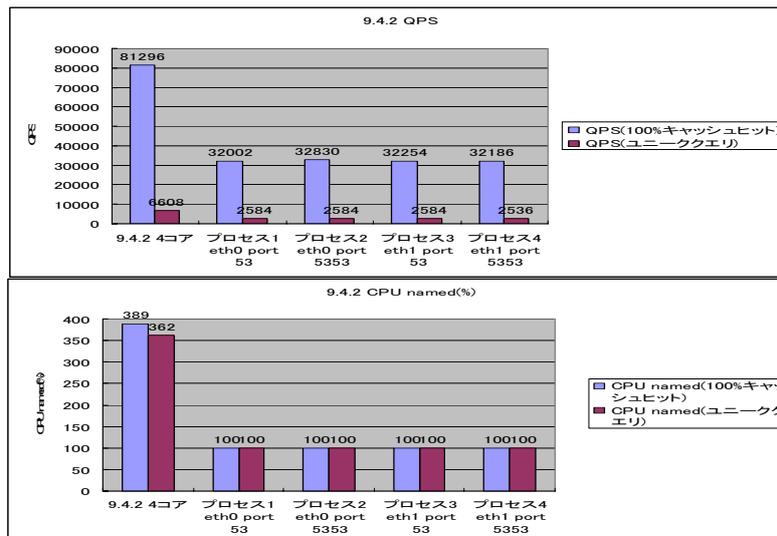
BIND 9.4.2 , 9.5.0

マルチコアプロセス(コア*4) vs. シングルコアプロセス*4

- マルチコアプロセスのnamed(コア*4)と、シングルコアプロセスのnamed*4のときのパフォーマンスを比較する
- シングルコアプロセス*4のときの設定
 - eth0 port 53 , eth0 port 5353 , eth1 port 53 , eth1 port 5353
 - tasksetコマンドで各プロセスが使用するコアを指定(2番、3番、6番、7番)
- クエリリスト
 - ユニーククエリ
 - 100%キャッシュヒット

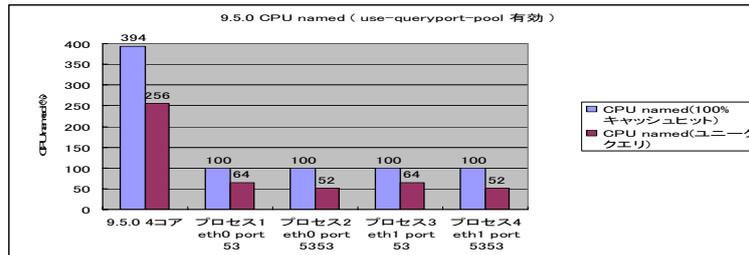
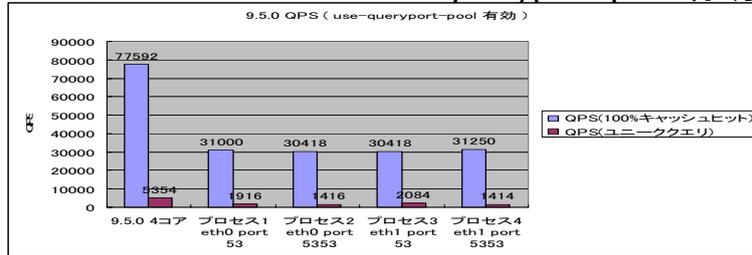
BIND 9.4.2

4コア vs. 4プロセス



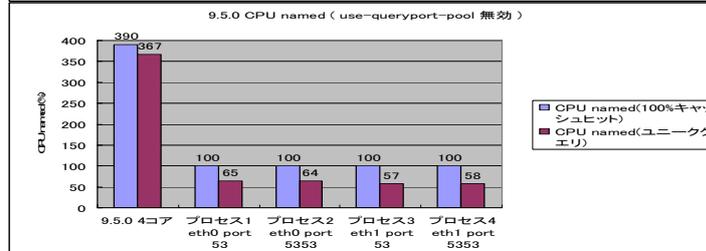
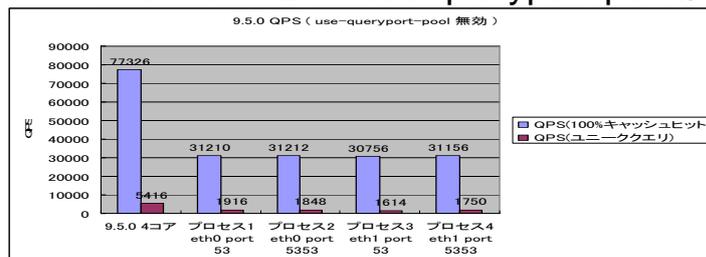
BIND 9.5.0

4コア vs. 4プロセス use-queryport-pool有効



BIND 9.5.0

4コア vs. 4プロセス use-queryport-pool無効



まとめ (1 of 2) コア数 vs. QPS

		シングルコア プロセス(QPS)	マルチコアプロセス (コア*8) (QPS)	
BIND 9.4.2	ユニーククエリ	4,194	10,552	2.5倍
	100%キャッシュヒット	32,368	106,508	3.2倍
BIND 9.5.0 use-queryport-pool有効	ユニーククエリ	3,900	5,646	1.4倍
	100%キャッシュヒット	30,564	97,794	3.1倍
BIND 9.5.0 use-queryport-pool無効	ユニーククエリ	4,178	9,888	2.3倍
	100%キャッシュヒット	30,900	96,320	3.1倍

- シングルコアプロセス
 - 9.4.2と9.5.0 use-queryport-pool有効,無効でパフォーマンスに大差なし
- マルチコアプロセス(コア*8)
 - 9.5.0 use-queryport-pool有効、ユニーククエリのケース:
 - マルチコア数によるQPSの向上があまり見られなかった
 - CPUの使用率が他のケースと比べ低い → QPSが出ない原因?
 - use-queryport-pool無効だと、9.4.2に近い値なので、use-queryport-poolオプションが起因しているように見える

まとめ (2 of 2)

マルチコアプロセス vs. シングルコアプロセス複数起動

		マルチコアプロセス (コア*8) (QPS)	シングルコアプロセス*4 (合計QPS)
BIND 9.4.2	ユニーククエリ	10,552	10,288
	100%キャッシュヒット	106,508	129,254
BIND 9.5.0 use-queryport-pool有効	ユニーククエリ	5,646	6,830
	100%キャッシュヒット	97,794	123,086
BIND 9.5.0 use-queryport-pool無効	ユニーククエリ	9,888	7,128
	100%キャッシュヒット	96,320	124,334

- **BIND 9.4.2**
 - シングルコアプロセス*4で、マルチコアプロセス(コア*8)とほぼ同等のパフォーマンスが出た
 - 各シングルコアプロセスのQPSはほぼ同じ
- **BIND 9.5.0 use-queryport-pool有効/無効**
 - ユニーククエリ: 各プロセスのQPSにばらつきがある。CPU使用率が低い
 - 100%キャッシュヒット:各プロセスのQPSに大きなばらつきはない
 - 原因は不明