# PostgreSQL と 半導体ストレージの効果検証

SRA OSS Inc, Japan.
山田 努

# 目次

- 概要
  - PostgreSQL と半導体ストレージ
- 検証環境、検証方法について
  - ハードウエア
  - ソフトウエア
- 検証結果、考察
  - SolidSTOR
  - SuperCACHE
- まとめ

# 概要

- PostgreSQL で CMS 半導体ディスク装置を使った場合の効果を検証する
  - ベンチマーク測定
  - 磁気ディスク装置との比較

# データベースとディスク装置

- 最終的にデータが記録される領域
  - 読み書きともにディスクアクセスが発生する
- 一般的な磁気ディスクの読み書きは遅い
  - I/O ボトルネックにより処理性能に限界がある
- 半導体ディスクは I/O が速い
  - 半導体ディスクの利用により、性能向上が期待される

# 半導体ディスク装置

- SolidSTOR
  - メモリディスク装置
- SuperCACHE
  - RAID 装置のフロントエンドとなる大容量キャッシュ装置
- メモリを使って記録するため、物理的なシークが発生せず、ランダムアクセスにも強い

# 検証環境: ディスク装置

- CMS 半導体ディスク装置
  - SolidSTOR
  - SuperCACHE
    - (兼用機 メモリ64GB 搭載)
- RAID装置: DotHill (REvo 27320R)
  - SuperCACHE のソースデバイスとして接続

# 検証環境: サーバホスト

- データベースサーバ 1
  - HP ProLiant DL145 G3 (Opteron 2214HE)
    - 2 core, Memory 2GB
    - 本体内蔵SASディスク
    - HBA: LSI7204EP (4G FC x 2ports)
- データベースサーバ 2
  - CPU: Quad-Core AMD Opteron(tm) Processor 8374 HE (2.2GHz) x4
    - 16core, Memory 32GB
    - FC x 4ports

# 検証環境: ソフトウエア

- OS: CentOS 5
- サーバプログラム: postgresql-8.4.1
- クライアントプログラム: pgbench 8.5 (HEAD) pthread 対応版
  - (2009/12 時点での最新)

# PostgreSQL の設定

- 特別なチューニングはなし
- 設定はほとんど初期状態
  - 修正するのは以下程度
    - max_connection = 1000
    - shared_buffer = 512MB　# 搭載物理メモリ 2G の 1/4
    - checkpoint_segments = 64
    - autovacuum off
    - 接続条件関係 pg_hba.conf, listen_address
- DB サーバとは別のホストから、ネットワーク経由でリクエストを送る

# 実験方法 (1/2)

- pgbench を使い同時接続数を変えて性能 (tps = transaction per second) を確認する
- 同時に vmstat による cpu の利用状況や I/O の状況を観測する

# 実験方法 (2/2)

- 対象ディスク装置にデータベースクラスタを作成する
- pgbench の scale factor を指定して初期化することで、適当なデータサイズのデータベースを作成する (pgbench -i -s NNNN)
- scale factor 1000 でデータサイズが 約15GB になる
- scale factor 5000 であれば約75GB

# pgbench 実行パターン

- pgbench がデータベースに送出するリクエスト
  - 通常の pgbench が行う方法 (TPC-B)
  - -N オプション
    - 一部のテーブル更新を行わない
    - テーブルのロック待ちが減り、全体的な処理数が多くなる
- 同時接続クライアント数は 1～512
- 90秒間の実行を3～5回実行し平均を取る

# 参考

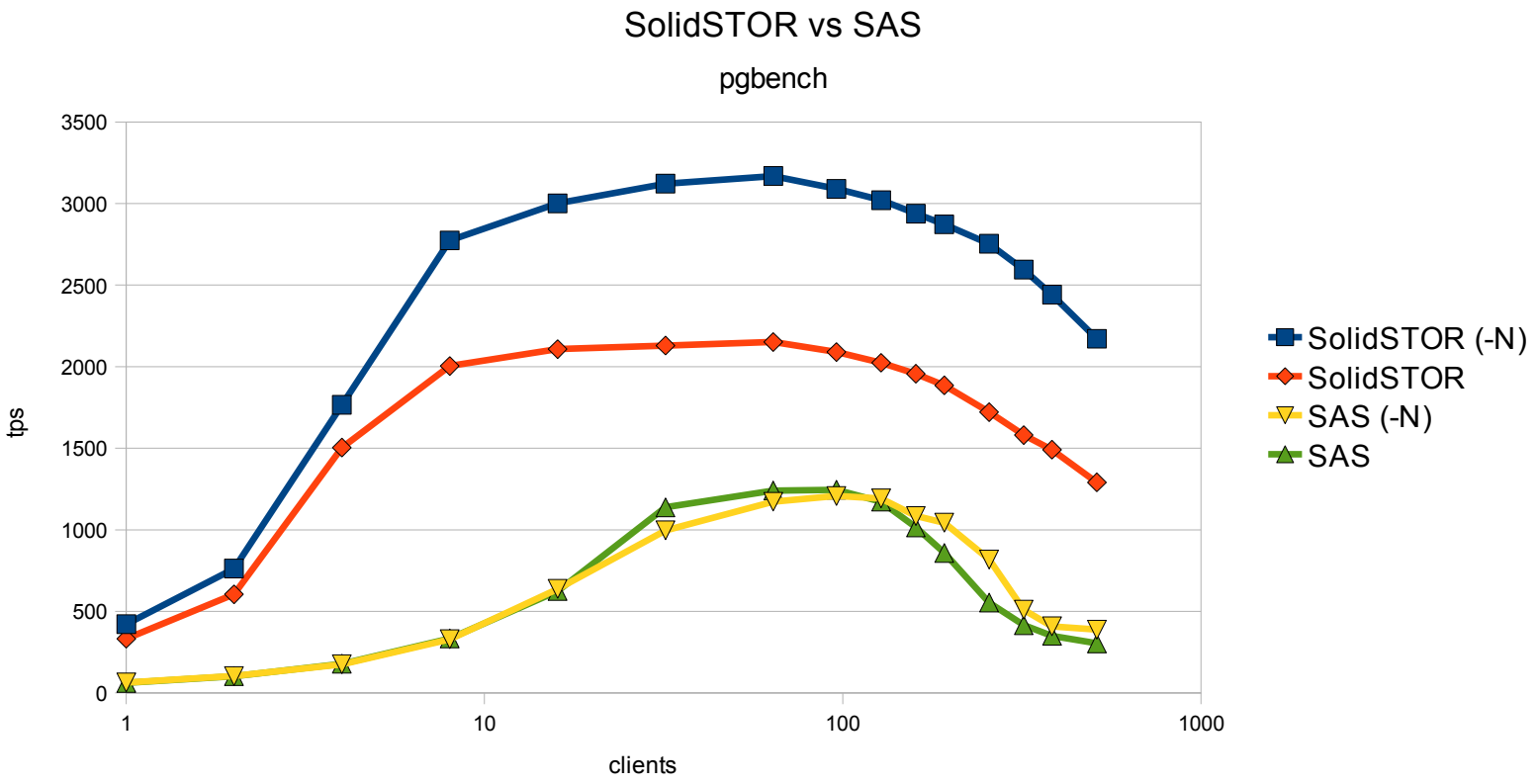
- 単純なディスク転送速度 (シーケンシャルアクセス)
  - CMS 400MBytes/sec
  - DotHill 250MBytes/sec
  - SAS 80MBytes/sec
- 半導体ディスクの場合、デバイスとホスト間でのデータ転送速度、つまりは FC 性能での限界がある

# 検証1: SolidSTOR

- サーバホスト 1 を使用
- SolidSTOR と SAS での性能比較
- それぞれのディスク装置に 15GBのデータベースを作成し、pgbench でアクセスし性能差を見る
- ディスク装置以外の条件は同じ
- ディスク性能の違いで、システム性能・負荷状態がどう変わるのかを見る
  - どのくらい速くなるのか?

# 結果1-1: SolidSTOR (pgbench)

- pgbench 結果 (vs SAS)

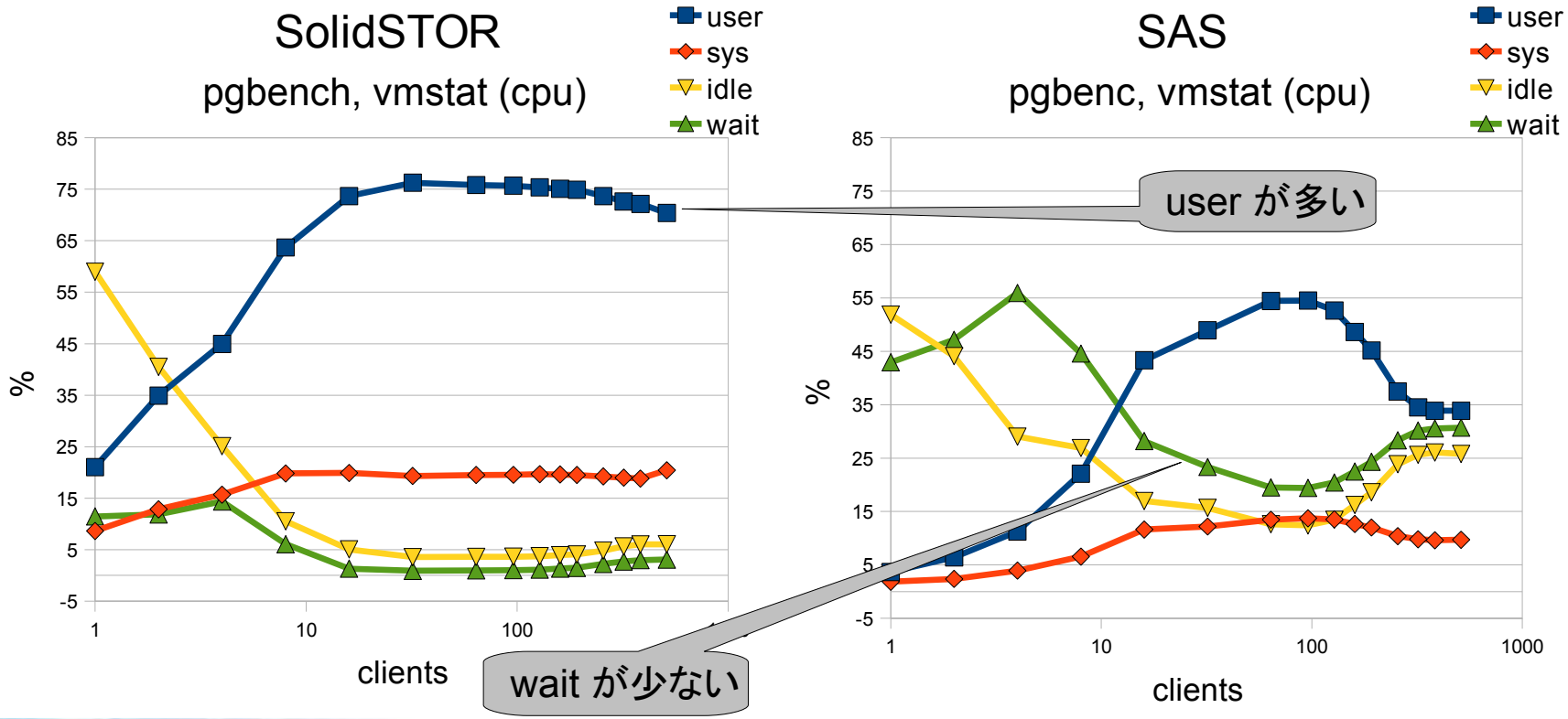


# 考察1-1: SolidSTOR (pgbench)

- TPS
  - SolidSTOR は、内蔵 SAS と比較してと3～10倍程度の性能が出ている
- 100クライアントを越えると全体的な性能が落ちるがPostgreSQL 側の処理効率の問題と考えられる

# 結果1-2: SolidSTOR (vmstat)

- pgbench 実行中の cpu 時間の平均値



# 考察1-2: SolidSTOR (vmstat)

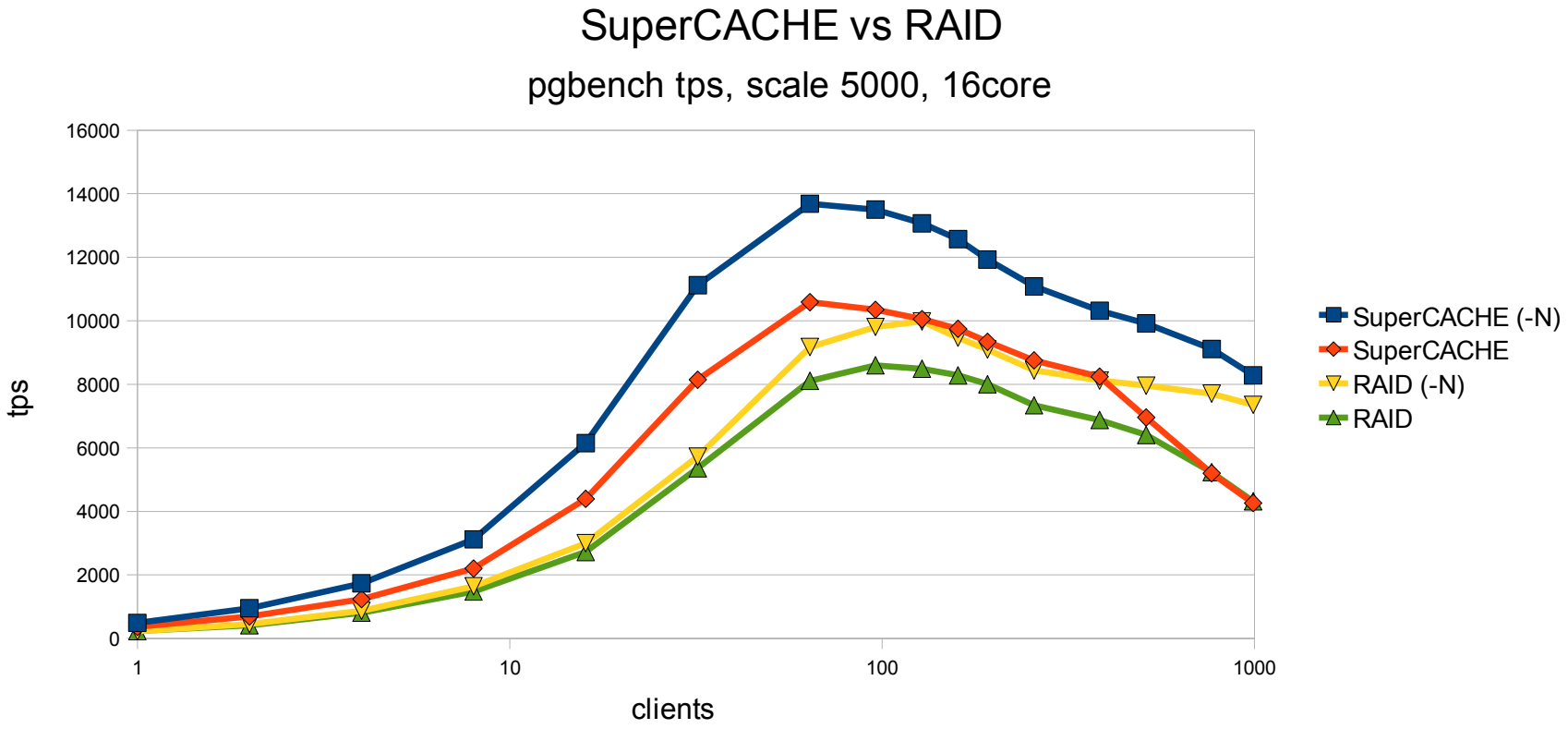
- vmstat
  - ディスク I/O 待ち (wait)の割合が減る
  - wait が減った分 user が増えており、処理効率が良くなっている
  - 同時接続数(クライアント数)が増えても wait はあまり増えない

# 検証2: SuperCACHE

- サーバホスト 2 を使用
- SuperCACHE と RAID装置での性能比較
- データベースサイズは 75GB (scale factor 5000)
  - データサイズが 15GB 程度では、すべてがメモリ上に載ってしまうため SolidSTOR とほぼ同じ結果になる
- キャッシュを追加することで、システム性能に与える影響を見る
  - どれくらいの効果が期待できるのか?

# 結果2-1: SuperCACHE (pgbench)

- pgbench 結果 (サーバ 2, vs RAID)

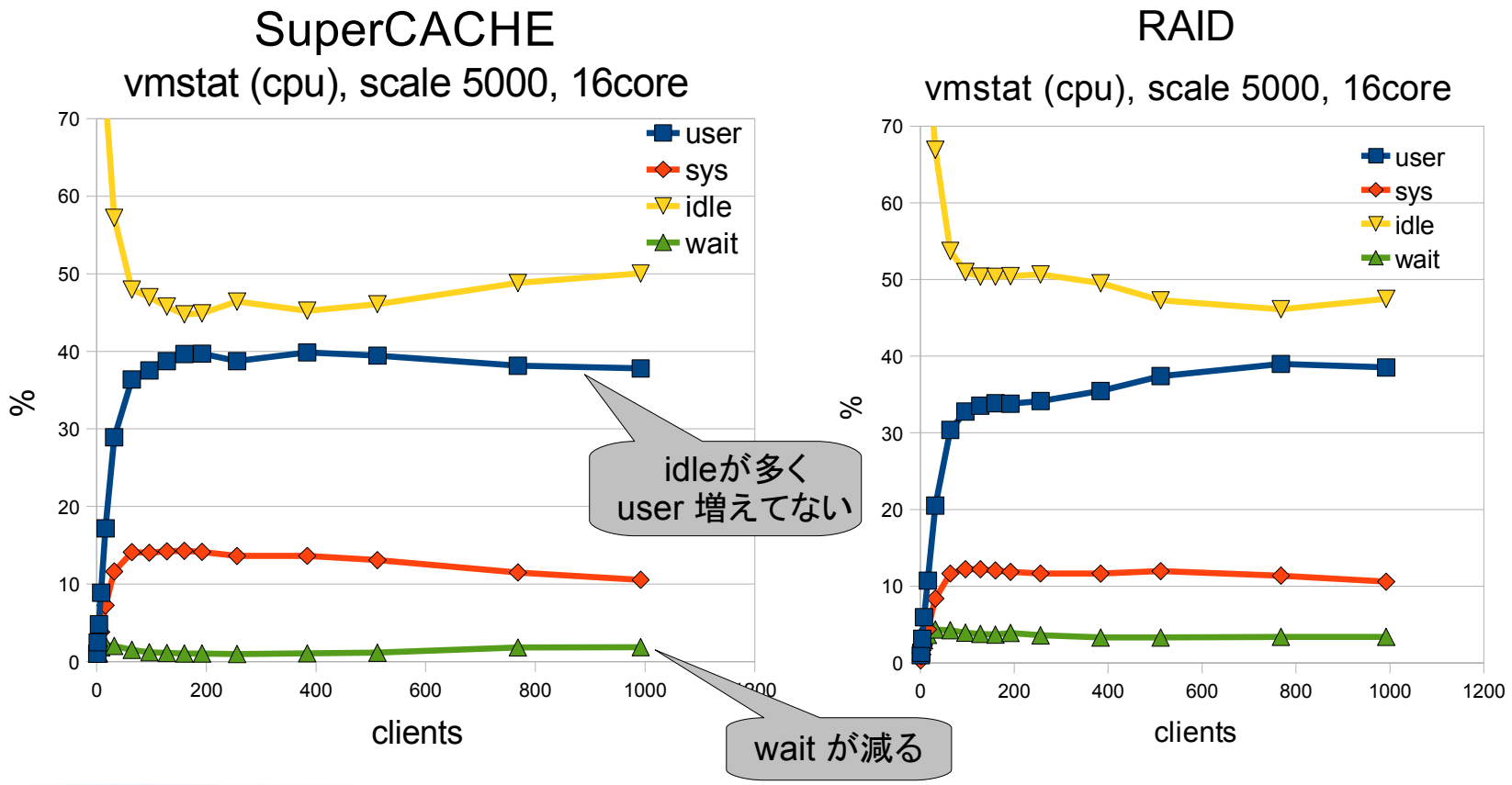


# 考察2-1: SuperCACHE (pgbench)

- TPS
  - SuperCACHE は、RAID装置と比較して 1.3倍程度の性能が出ている
- 100クライアントを越えると全体的な性能が落ちるがPostgreSQL 側の処理効率の問題と考えられる

# 結果2-2: SuperCACHE (vmstat cpu)

- サーバ 2 で pgbench 実行中の cpu



# 考察2-2: SuperCACHE (vmstat)

- vmstat
  - ディスク I/O 待ち (wait)の割合は減る
    - グラフでは差があまり見えないが SuperCACHE 1% vs RAID 4%
  - サーバ 2 はかなり高性能なホストを使ったが user の割合が頭打ちになっている
    - idle が多い。つまり cpu には余裕がある
    - wait は増えていないので、PostgreSQL 内部処理での待ちが大きくなっていると推定される

# その他の実験

- SuperCACHE データサイズ 75GB で VACUUM FULL など全データ領域を走査する処理を行う
  - ストレージ内部のキャッシュヒット率情報を観測してみたところ、当然だがヒット率は悪くなる

# まとめ

- PostgreSQLに高速ディスク装置を組み合わせるメリット
  - DBのボトルネックはCPUではなくてI/O
    - I/O 待ちが減少し、DBの性能向上が可能
    - I/O 負荷が多いシステムほど性能向上が期待できる
  - アプリケーションやOSなど、ソフトウェアを一切変更することなく性能向上できる
  - 性能向上のための作業時間が短かく、移行後の検証も不要
    - 基本的にはデータをコピーしてつなぎ替えるだけ
    - バージョンアップやチューニングでは、検証も含めて数日以上移行時間がかかる

# 製品の使い分け

- SolidSTOR
  - データベースサイズがあまり大きくなく、参照更新負荷が極端に大きい場合に有効
  - データの初期アクセスが遅くて困るケースに絶大な威力
    - DBサーバのメモリを増やしてキャッシュ効果をあげても、データの最初のアクセスにはI/Oが発生し、非常に遅くなる
- SuperCACHE
  - データベースサイズが大きい場合にも有効
  - ランダムアクセスが多い場合に有効
    - バッチ処理など全データを走査する場合でも、インデックスがキャッシュされれば効果はある