PGECons
PostgreSQL Enterprise Consortium

大規模DBを見据えた PostgreSQLの性能検証

WG1: 性能WG活動成果中間報告

WG1主査
SRA OSS, Inc. 日本支社
石井　達夫

WG1（性能ワーキンググループ）の役割

- 技術部会で設定された課題に具体的に取り組む実働部隊の一つで主に性能関係の課題をテーマとする
  - ミッションクリティカル性の高いエンタープライズ領域への適用に向けて、本体・周辺ツールに関する技術ノウハウを共有する
- 性能に関する課題の例
  - CPUマルチコアに対する性能懸念
  - 負荷分散方式が確立されていない
  - などなど（総数約50件）
  - この中から問題点を整理し、PGEConsとして取り組んで効果がありそうなものをテーマ化
- さらに今年度実施するものを絞り込む

活動テーマ候補の一部抜粋

大項目	小項目	概要
性能	性能評価手法	業務別（オンライン業務やバッチ業務、大容量の分析）性能モデルの整備
		サイジング手法の整備
	性能向上手法	マルチコアCPUでのスケールアップ性検証
		負荷分散クラスタでのスケールアウト性検証
		クエリキャッシュ適用手法の検証
		パーティショニングによる性能改善の検証
		初期ロードの高速化手法の整備
	チューニング	チューニングノウハウの整備
		実行計画の制御手法の検討
可用性	高可用 クラスタ	高可用クラスタ手法の検証
	災害対策	非同期レプリケーションによるBCP手法の整備

企業でのニーズ（WG1）

	具体的な企業ニーズ	検証しておきたい技術
①	「最近のサーバはマルチコア化している。マルチコアサーバでも性能メリットがあるDBMSを利用したい。」	コア数差による性能
②	「月次で、本店から町工場へ部材情報をバッチ転送している。町工場でも、よりタイムリーに本店の情報を見れると嬉しい。」	Cascading Replication
③	「複数のサーバを並べて負荷分散をはかりたい。」	pgpool-II　Postgres-XC
④	「震災・停電時のBCP対策として、ディザスタリカバリ構成をとりたい。」	Cascading Replication　pgpool-II
⑤	「自国から様々な海外拠点へ、DBのデータを送って使用したい。」	Cascading Replication
⑥	「サイズの大きいデータベースを扱いたい」	PostgreSQL本体　Postgres-XC
⑦	WGにて寄せられた意見：「PGEConsメンバ企業視点でPostgres-XCはどんな印象？」	Postgres-XC

2012年度の注力部分（囲み）

PGEConsへのご意見

- PGEConsへ寄せられた意見（スケールアウト関連のみ抜粋）

	エンタープライズ領域で採用するための課題	ご意見元	技術検証項目候補
1	更新系のレプリケーション、複数台でのレプリケーション	技術部会/WG1会合 セミナーアンケート（2012年7月）	マスタDB更新性能への影響 多段構成（孫レプリケーション）
2	性能スケーラビリティが不足	セミナーアンケート（2012年7月）	性能スケーラビリティ情報の拡充
3	負荷分散構成の情報が不足している	技術部会	実機検証情報の拡充
4	同期レプリケーション情報が不足している	技術部会	実機検証情報の拡充
5	既存アプリケーションの改修が必要	セミナーアンケート（2012年7月）	スケールアウト、スケールアップに伴う改修要否
6	高可用構成での実績や事例が少ない	セミナーアンケート（2012年7月）	実機検証情報の拡充

WG1（性能ワーキンググループ）の今年度テーマ

- スケールアップ検証
  - 多コアCPUでの性能検証
  - PostgreSQL 9.2での到達点の把握
    - pgbenchによる検索性能検証
    - JDBCrunnerによるTPC-Cライクなベンチマーク
- スケールアウト検証
  - 以下のOSSについてスケールアウト特性を検証する
    - PostgreSQL 9.2+Cascading streaming replication
      - 非同期レプリケーション
    - pgpool-II（レプリケーションモード）
      - 同期レプリケーション+検索負荷分散
    - Postgres-XC
      - 同期データ分散+更新負荷分散

期待される成果物

- スケールアップ、スケールアウト検証の手順および結果文書
- 具体的なハードソフト構成、構築手順、検証結果データを公開
- 検証結果、手順書、スクリプトなどは再利用、再配布可能なライセンスを設定する予定
- 2013年4月に公開予定

実施体制

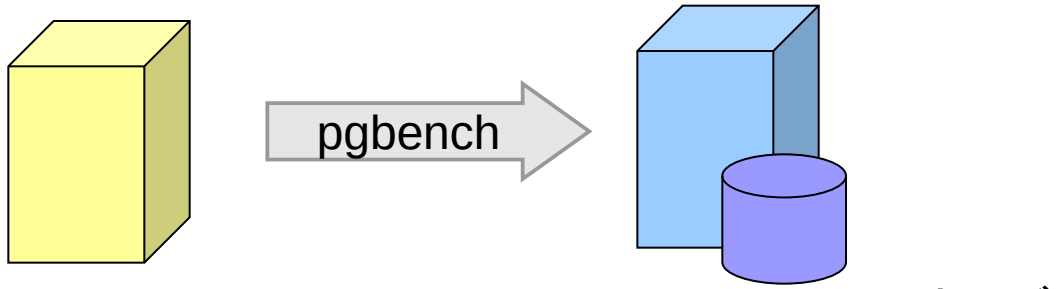
- 参加企業（企業名順）
  - 株式会社アシスト
  - SRA OSS, Inc.日本支社（主査）
  - NECソフト株式会社
  - 日本電気株式会社
  - 日本電信電話株式会社
  - 日本ヒューレット・パッカード株式会社
  - 株式会社日立製作所
  - 富士通株式会社

中間報告1: スケールアップ検証

- 1.1 pgbenchによる検索性能スケールアップ検証
  - 80コアCPUマシンでのPostgreSQLの検索性能のスケールアップ特性を検証
  - その結果、コア数と同じ80クライアントまで性能が向上することが確認できた

pgbench によるスケールアップ検証

- 【 目的 】多数コアでの参照において、クライアント数が増加したときにスケールアップするかどうかを確認する
- 【 方法 】カスタムスクリプトを作成し、pgbench -h [host] -p [port] [dbname] -c [c] -j [j] -T 30 -n -f custom.sql を実行する
  - クライアント数(-c)を変動させる。スレッド数(-j)はその半分とする。
  - pgbench はクライアント用のマシンから実行する。



pgbenchとは

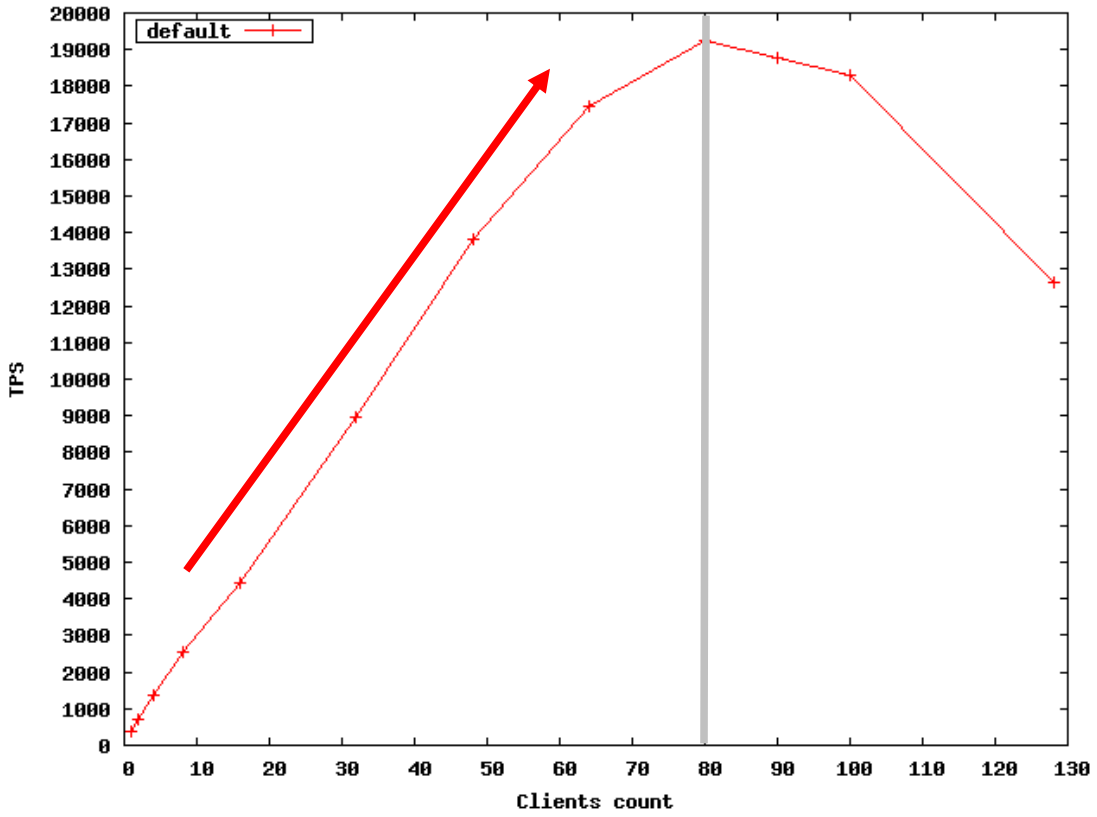
- PostgreSQL上でベンチマーク試験を行うプログラム
  - PostgresSQLに標準添付
  - SQLコマンドの並びを自動実行し、実行結果のトランザクションの速度（tps: 1秒当たりのトランザクション数）の平均を計算する
  - デフォルトで実行された pgbench は、1トランザクション当たり5つのSELECT、UPDATE、INSERTコマンドを含むおおよそTPC-Bに基いたシナリオを実行する
  - カスタムシナリオを実行させることも可能

実施したスクリプト（参照系トランザクション）

```
¥set nbranches :scale
¥set ntellers 10 * :scale
¥set naccounts 100000 * :scale
¥set row_count 10000
¥set aid_max :naccounts - :row_count
¥setrandom aid 1 :aid_max
¥setrandom bid 1 :nbranches
¥setrandom tid 1 :ntellers
¥setrandom delta -5000 5000

SELECT count(abalance) FROM pgbench_accounts WHERE aid
BETWEEN :aid and :aid + :row_count;
```

結果



pgbench -S のように平易なトランザクションだと変化がでにくいが、負荷の高いトランザクションの場合、コア数と同じ 80 クライアントまで性能が向上する

中間報告1: スケールアップ検証

- 1.2 JDBCrunnerによるスケールアップ検証
  - 更新系ベンチマークにより、セッション数に応じて性能が向上することが確認できた

更新系処理におけるスケールアップ検証

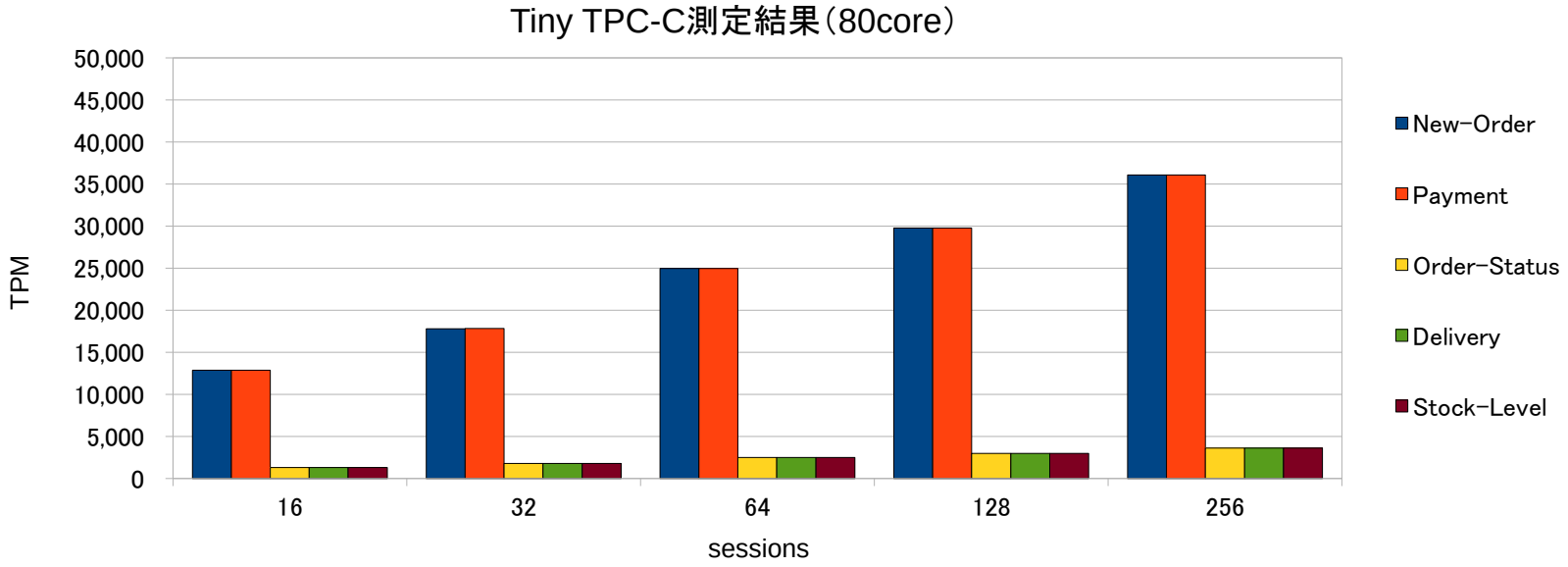
- 検証概要
  - JdbcRunnerによる更新系処理での性能測定
  - データベース内部処理による性能への影響度合いを確認
- JdbcRunnerとは

特徴	・国産のJavaベース(JDBC接続)のデータベース負荷ツール ・ライセンスはNew BSD License ・各種DB（PostgreSQL、Oracle、MySQL)に対応し、Tiny SysBench、Tiny TPC-B、Tiny TPC-Cなどのテストスクリプトも提供されている
Tiny TPC-C	・TPC-C Standard Specification5.10.1を簡易実装したJdbcRunnerスクリプト ・仕様書のうち以下の章節を実装 1 LOGICAL DATABASE DESIGN 2 TRANSACTION and TERMINAL PROFILES 2.4 The New-Order Transaction (2.4.1.1、2.4.3を除く) 2.5 The Payment Transaction (2.5.1.1、2.5.3を除く) 2.6 The Order-Status Transaction (2.6.1.1、2.6.3を除く) 2.7 The Delivery Transaction (2.7.1.1、2.7.2、2.7.3を除く) 2.8 The Stock-Level Transaction (2.8.1、2.8.3を除く) 4 SCALING and DATABASE POPULATION 4.3 Database Population 5 PERFORMANCE METRICS and RESPONSE TIME 5.2 Pacing of Transactions by Emulated Users 5.2.4 Regulation of Transaction Mix
ダウンロード	JDBC Runnerのダウンロードは以下より http://hp.vector.co.jp/authors/VA052413/jdbcrunner/

（参考）http://hp.vector.co.jp/authors/VA052413/jdbcrunner/

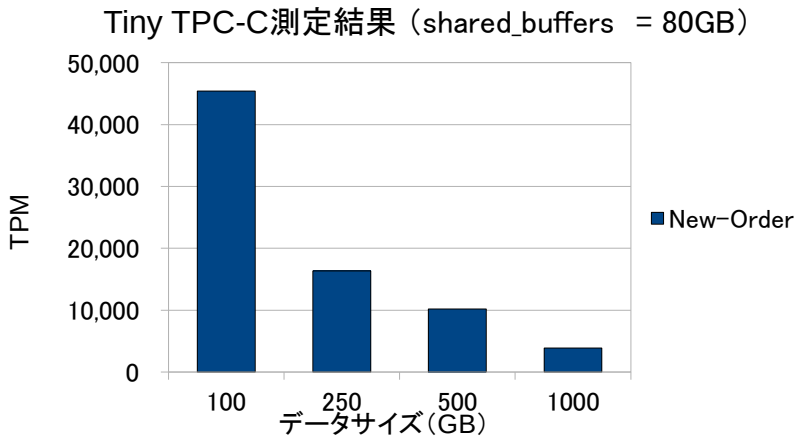
JdbcRunnerによるスケールアップ検証

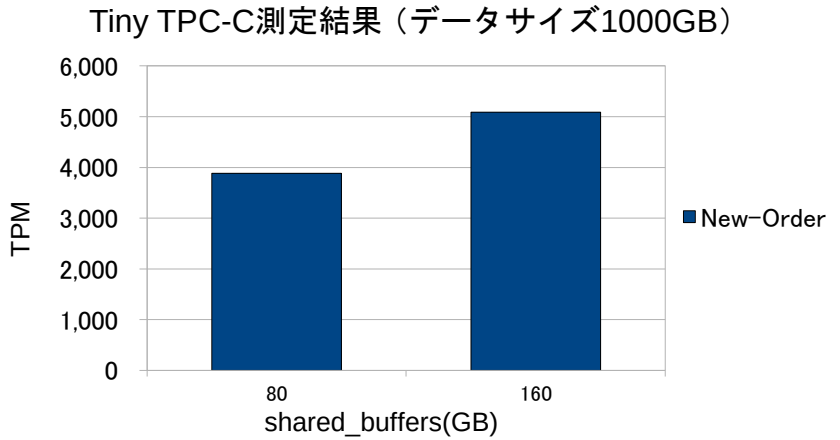
- セッションスケーラビリティ
  - 検証内容
    - CPUコア数を固定した状態で同時接続数を増加し、平均TPMの変化を確認
  - 結果
    - セッション数の増加に伴い、平均TPMが向上することを確認



JdbcRunnerによるスケールアップ検証

- アクセス対象データ範囲の拡大
  - 検証内容
    - データサイズおよびshared_buffersの値を増加させ、平均TPMの変化を確認
  - 検証結果
    - アクセス対象データの増加に対して急激な性能劣化はない
    - shared_buffersのサイズを大きくすることにより性能が向上することを確認





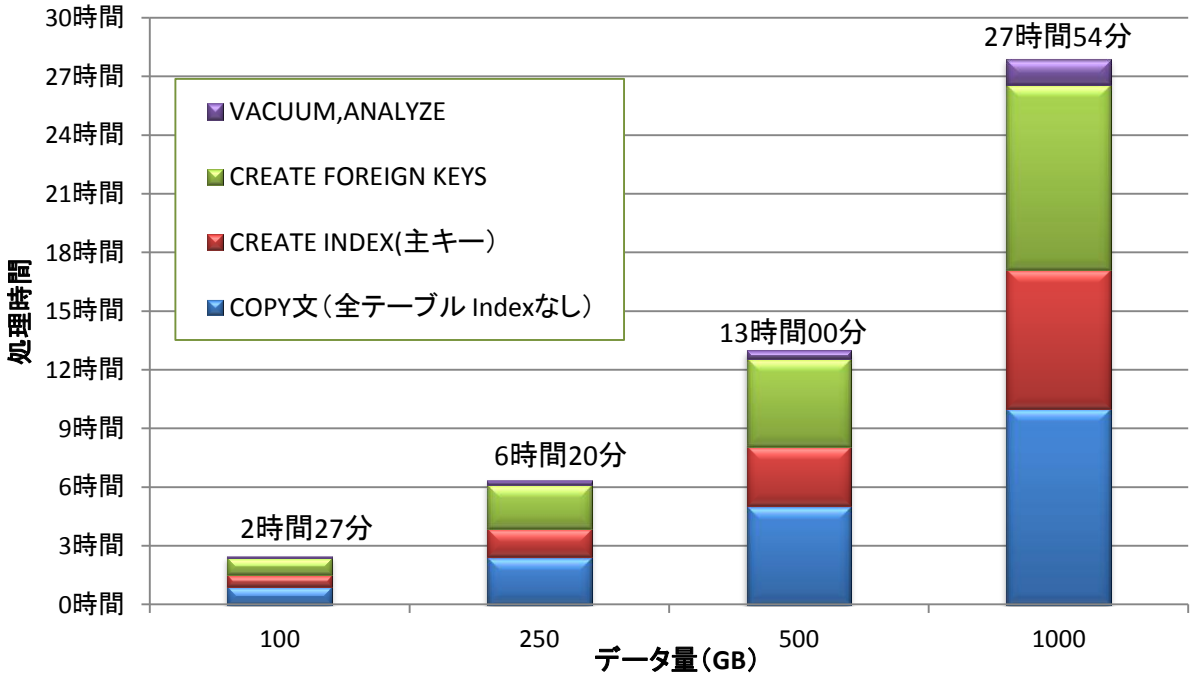
JdbcRunnerによる基本検証

- データベース内部処理
  - 検証内容
    - 以下のデータベース内部処理が性能に与える影響度合いを確認
      - checkpoint
      - 自動VACUUM
  - 検証結果

メンテナンス処理	結果	考察・課題
checkpoint	性能への影響は軽微	checkpoint_completion_targetパラメータによる処理分散の効果が高い （今回の検証では0.7に設定）
自動VACUUM	性能への影響は軽微	遅延VACUUM機能による処理分散の効果が高い （今回の検証ではデフォルト値）

大容量データ(CSV)のロード性能測定

- COPYおよびインデックス生成等の実行時間を計測



- 測定条件
  - PostgreSQL 9.2.1
  - JDBC RunnerのTiny TPC-Cのデータ作成部をベースにCSVを生成しCOPYを実行
  - archive_mode = off
  - 全テーブル FILLFACTOR = 80

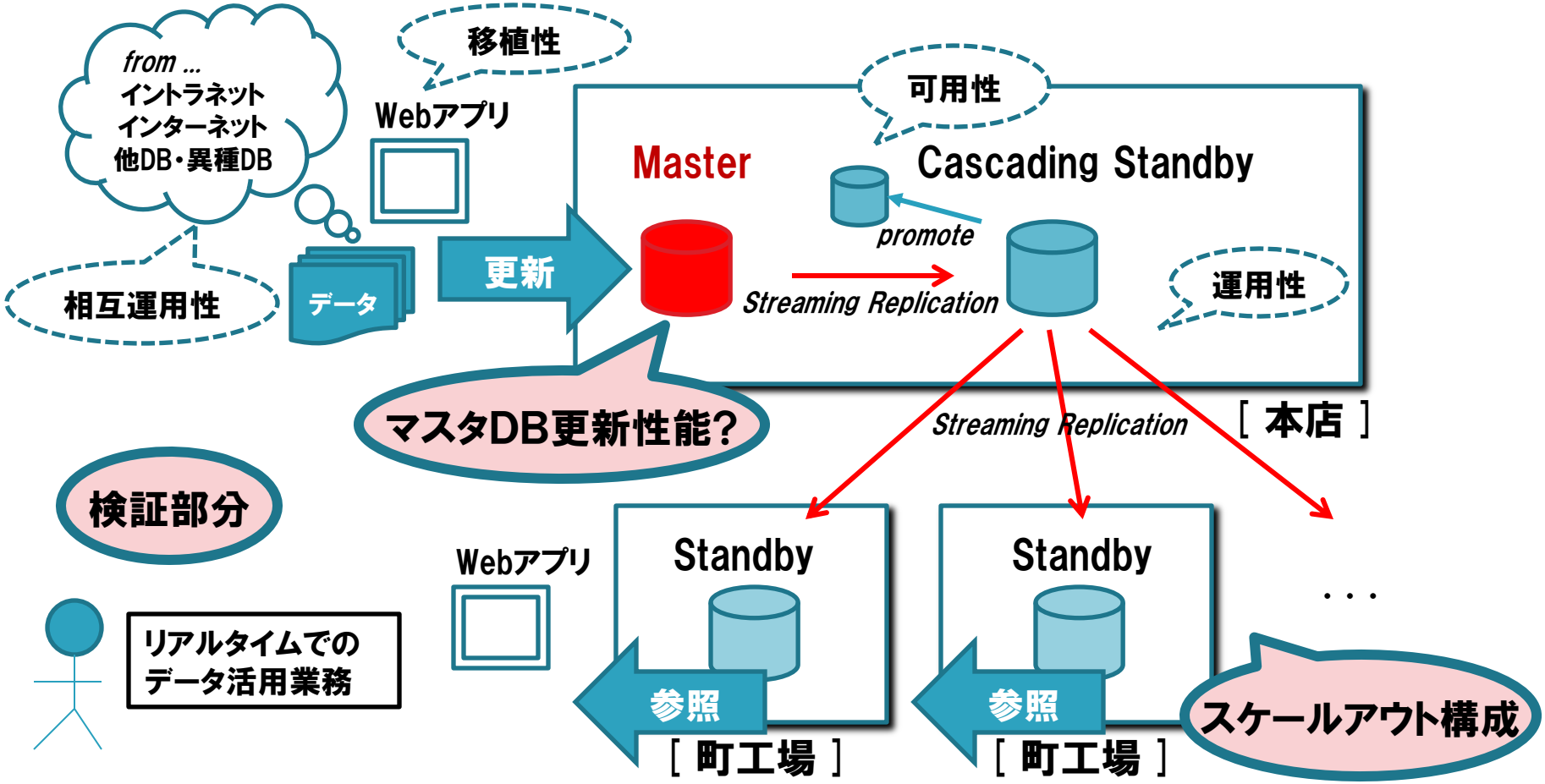
CSVのデータ量	100GB	250GB	500GB	1000GB
VACUUM , ANALYZE	4分	12分	26分	1時間17分
CREATE FOREIGN KEYS	51分	2時間14分	4時間29分	9時間28分
CREATE INDEX	34分	1時間27分	3時間01分	7時間06分
COPY文	56分	2時間25分	5時間02分	10時間00分
合計	2時間27分	6時間19分	13時間00分	27時間53分

中間報告2: スケールアウト検証

- 2.1 PostgreSQL 9.2 カスケードレプリケーション検証
  - レプリケーションノード(マスタから見て孫ノード)数を変化させて、マスタDBの更新性能を測定
  - レプリケーションノードの増加がマスタDBの更新性能に与える影響は小さいことが確認できた

更新系・複数台レプリケーション検証 (PostgreSQL 9.2)

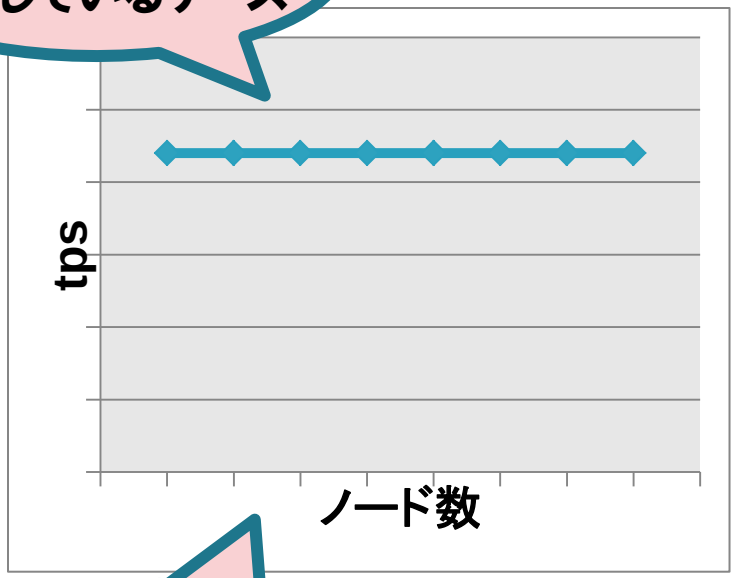
- カスケードレプリケーション(非同期)を適用したモデルとして、本店から町工場へ部材情報をタイムリーに流す形態を想定



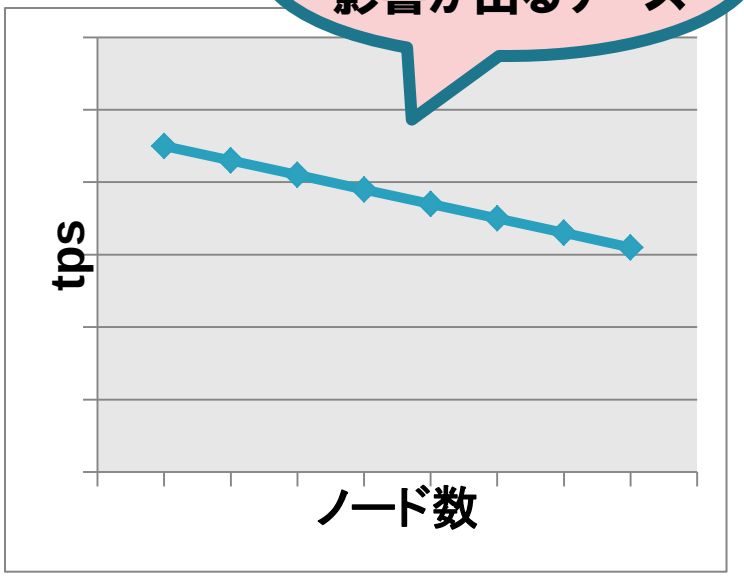
検証目的：マスタDB更新性能への影響確認

- カスケードレプリケーション（非同期）ではノード数が増えていくと、マスタDBの更新性能はどのような特性になるのか？

更新性能は
安定しているケース



更新性能に
影響が出るケース

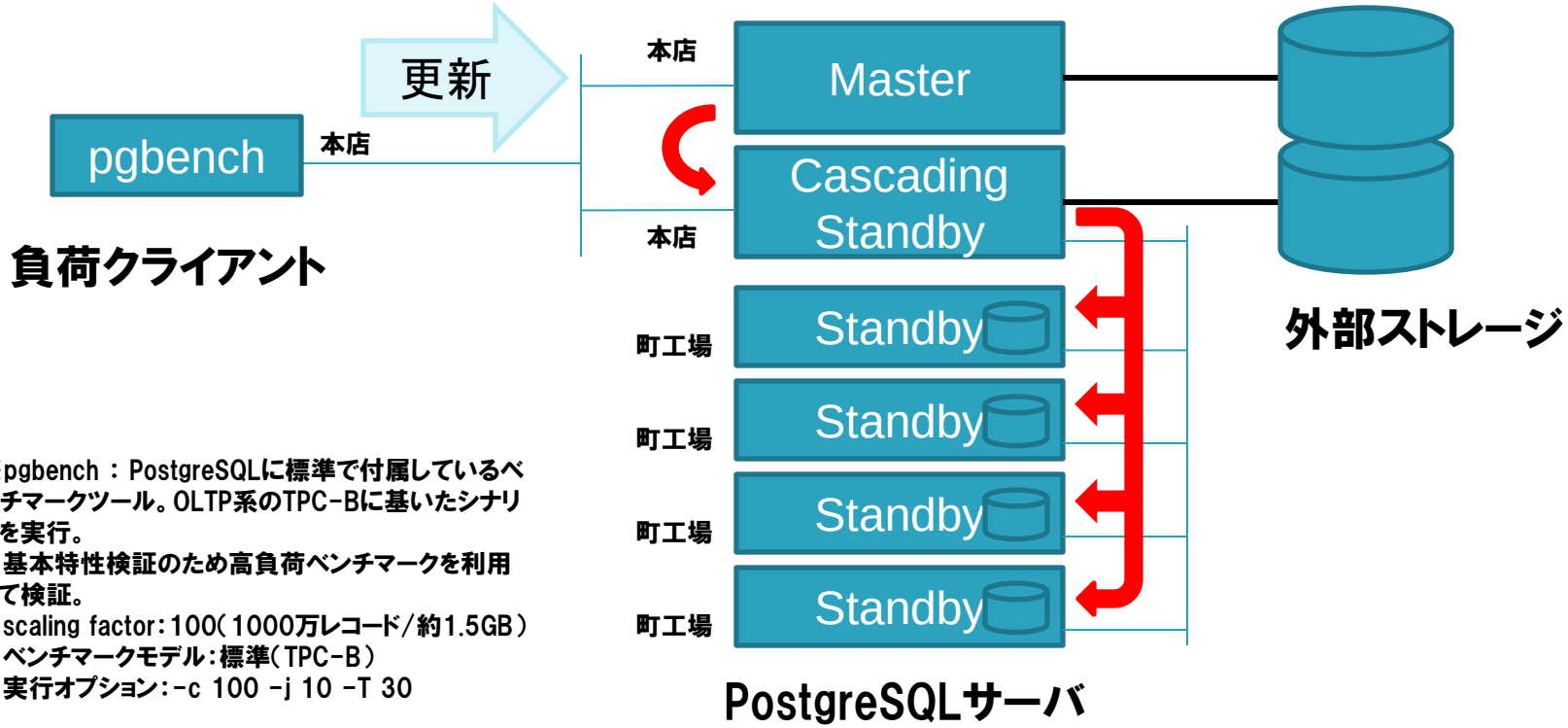


ノード数を変化させる

tpsは本店での更新スループット、
ノード数は町工場の数に相当

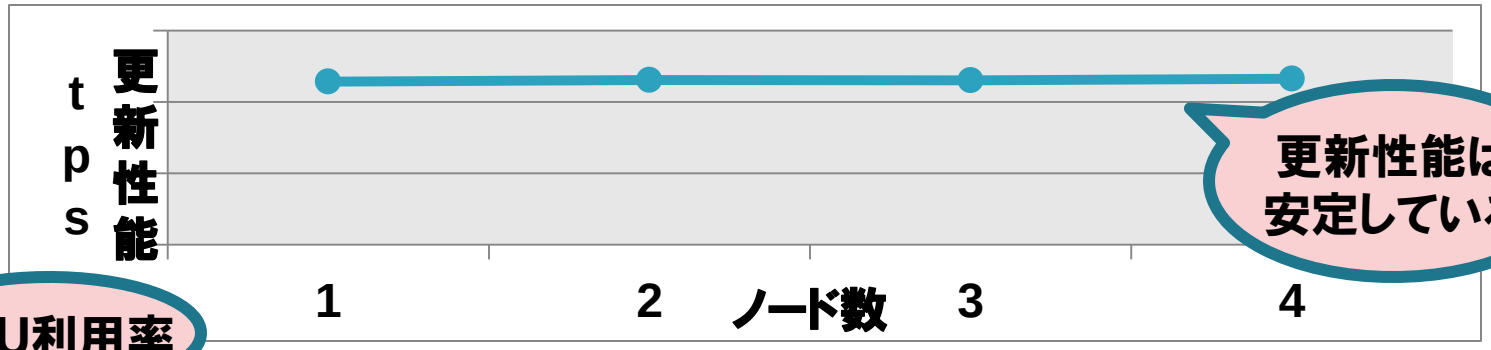
検証構成

- MasterとCascadingStandbyは外部ストレージ、Standbyはスタンドアローン
- 負荷クライアントからネットワーク経由でMasterへデータ更新
- Master→CascadingStandby→Standby×4 へネットワーク経由でレプリケーション



※pgbench ： PostgreSQLに標準で付属しているベンチマークツール。OLTP系のTPC-Bに基いたシナリオを実行。
- 基本特性検証のため高負荷ベンチマークを利用して検証。
- scaling factor：100（1000万レコード/約1.5GB）
- ベンチマークモデル：標準（TPC-B）
- 実行オプション：-c 100 -j 10 -T 30

PostgreSQL 9.2 カスケードレプリケーション検証結果



CPU利用率

Master

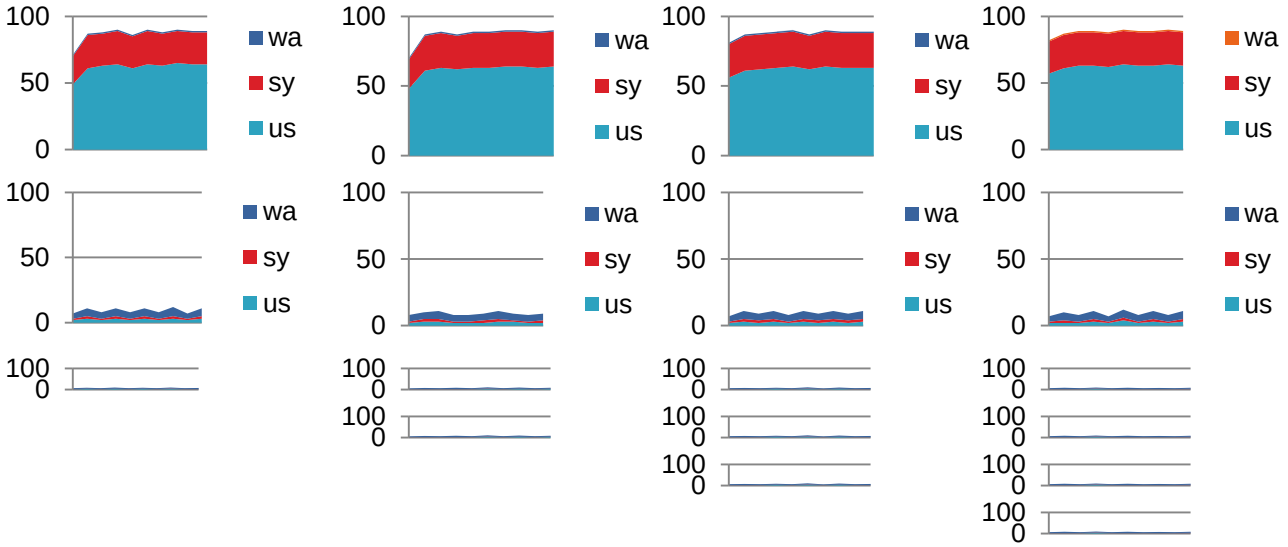
Cascading Standby

Standby

Standby

Standby

Standby



ノード数が増えてもマスタDBの更新性能はほぼ一定で安定している
機器構成にもよるが基本特性としてはノード数増がマスタDBへ与える影響は小さい

中間報告2: スケールアウト検証

- 2.2 pgpool-IIによるスケールアウト検証
  - 更新系処理では、ノード数に応じて性能が劣化することが確認できた
  - 参照系処理では、ノード数に応じて性能が向上することが確認できた

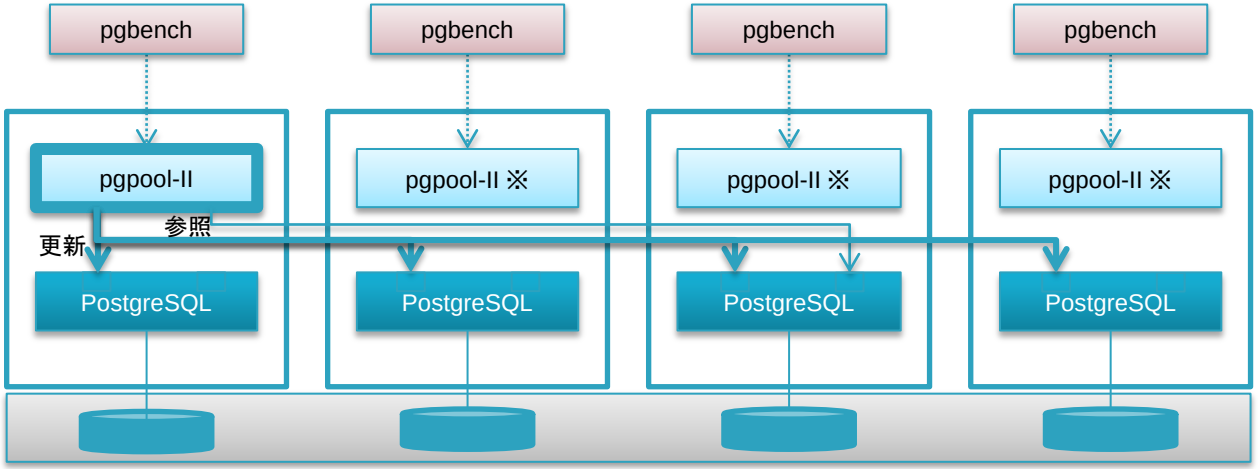
pgpool-IIとは

- PostgreSQL サーバとクライアントの間に入ってプロキシサーバとして動作するソフトウェア

機能	概要
コネクションプール	データベースへの接続を保持して使いまわす機能（DBにとってコストが高い新規受け付け処理を低減できる）
接続数の制限	設定した最大同時接続数を超えた接続要求に対して、接続待ちをさせる機能（PostgreSQL←→アプリ間に pgpool-II を挟むことで、DBの同時接続数をエラーを出さずに一定数に抑えることができる）
レプリケーション	データベースの複製を作って、その後にデータ更新処理があっても、同じデータ状態を維持していく機能 同期レプリケーション形式で実装されている（全てのノードで SQL がコミットされてからアプリケーションに処理を返す）
負荷分散	SQL の内容が参照処理か更新処理かを自動的に区別し、参照の問い合わせについては処理を振り分けて実行する機能 ※ pgpool-II は、参照SQLはどれか1ノードのDBに対してのみ実行させる
パラレルクエリ（並列問い合わせ）	1つの SQL について対象となるテーブルを分散配置しておき、各ノードで並列に処理を行い、統合した結果を返すようにする機能

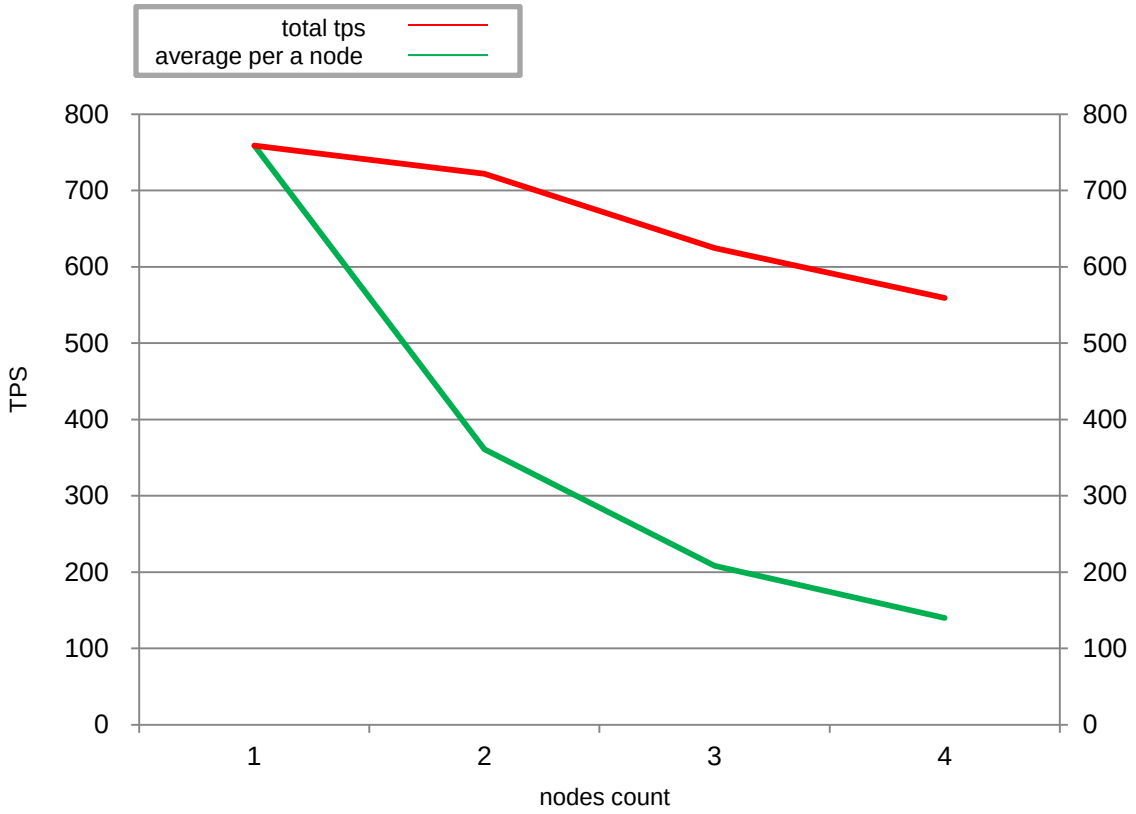
pgpool-II によるスケールアウト検証

- 目的
  - pgpool-II (3.2.1) & PostgreSQL (9.2.1) で、ノード数を増やすと全体の処理能力が向上するかどうかを確認する
- 方法
  - pgbench の各ノードでの結果を合計する
    - pgbench はクライアント用のマシンから実行する
    - 更新系性能の測定は、pgbench のデフォルトシナリオを用いる
    - 参照系性能の測定は、極力 I/O ネックの発生を抑えた状態で実行するため、カスタムシナリオを用いる



※ 2台目以降簡略化の為矢印は省略。

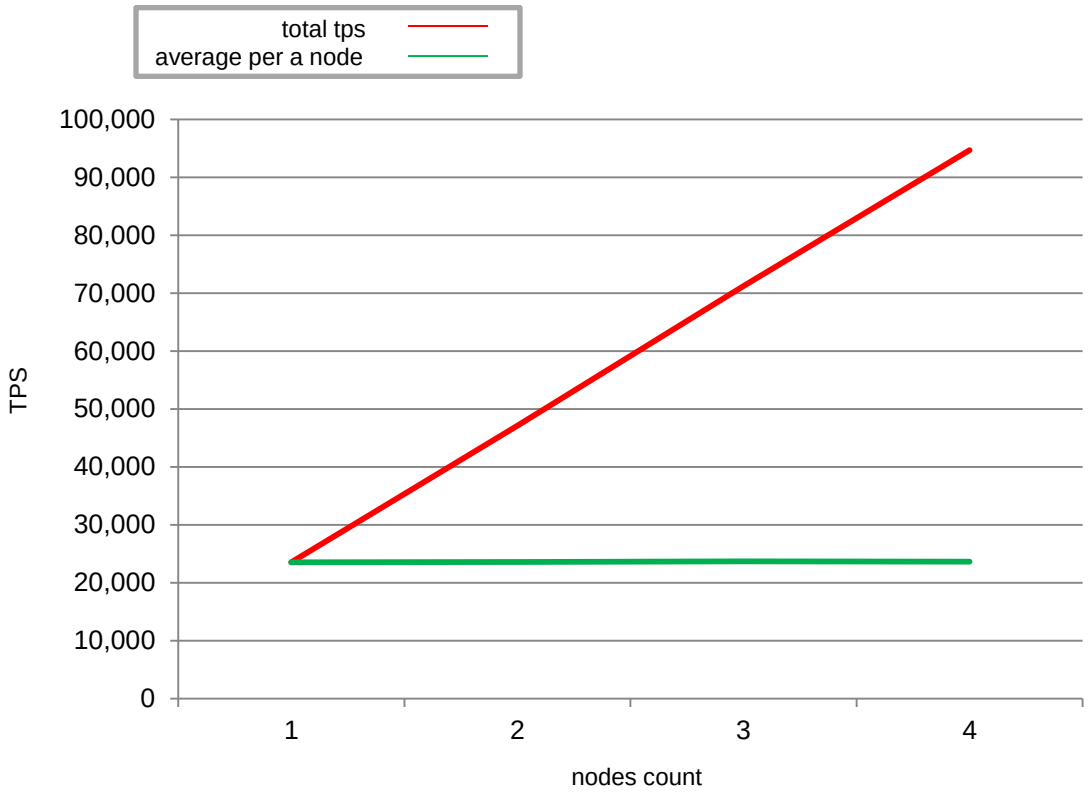
更新系の結果



- 以下の条件・指定でpgbenchを実行
  - シェアードメモリ: 16GB
  - SF値: 1000 でデータを準備
  - 実施時間(-T): 300秒
  - 同時実行クライアント数(-c): 100
  - ワーカスレッド数(-j): 20

- 2 ノード以上では、ノード数が増えるほど合計の tps が減少する
- 同期レプリケーションを行うため、更新処理を行わせた場合性能が出にくい
- 各ノードのデータの書き込み先を物理的に分散させることで、性能の劣化度を多少改善できる

参照系の結果



- 以下の条件・指定でpgbenchを実行
  - シェアードメモリ: 32GB
  - SF値: 1000 でデータを準備
  - 実施時間(-T): 300秒
  - 同時実行クライアント数(-c): 100
  - ワーカスレッド数(-j): 20
  - 実行シナリオ名(-f): カスタムシナリオファイル名を指定
  - 試験前バキュームの禁止(-n)

- 以下のカスタムシナリオを用意

```
¥set nbranches :scale
¥set ntellers 10 * :scale
¥set naccounts 100000 * :scale
¥set range 2000
¥set aidmax :naccounts - :range
¥setrandom aid 1 :aidmax
¥setrandom bid 1 :nbranches
¥setrandom tid 1 :ntellers
¥setrandom delta -5000 5000
SELECT count(abalance) FROM
pgbench_accounts WHERE aid
BETWEEN :aid and :aid + :range;
```

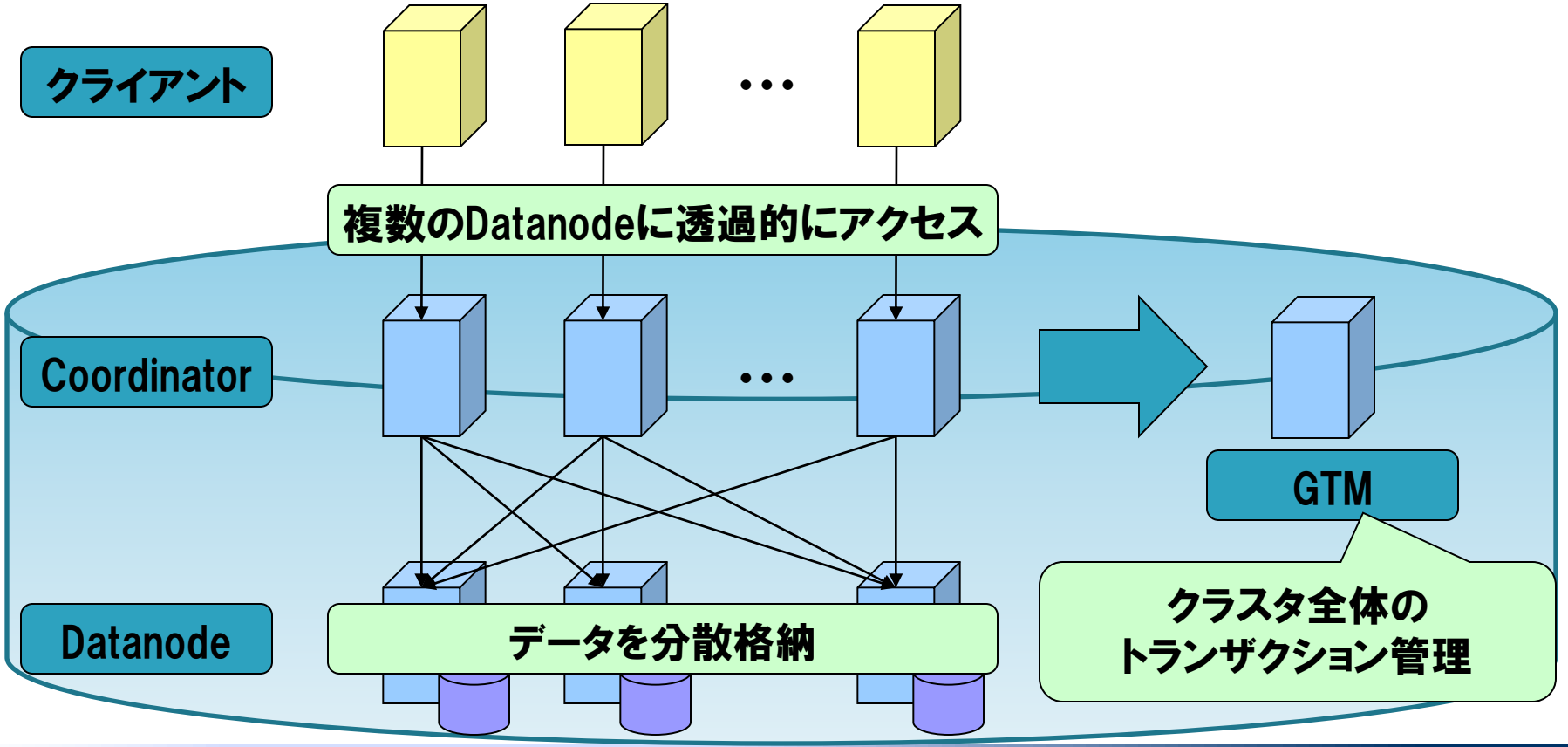
- ノード数が増えるほど、合計の tps が増える（スケールメリットあり）

中間報告2: スケールアウト検証

- 2.3 Postgres-XCによるスケールアウト検証
  - 更新系処理では、ノード数に応じてスケールアップすることが確認できた
  - 参照系処理では、データ量が多いときにスケールアップすることが確認できた

Postgres-XCとは①

- 大規模システムを想定したデータベースクラスタ
- 複数のDatanodeにデータを分散配置し**更新負荷を分散**
- 複数のCoordinatorとDatanodeに**参照負荷を分散**



Postgres-XCとは②

- Postgres-XCを構成するコンポーネント

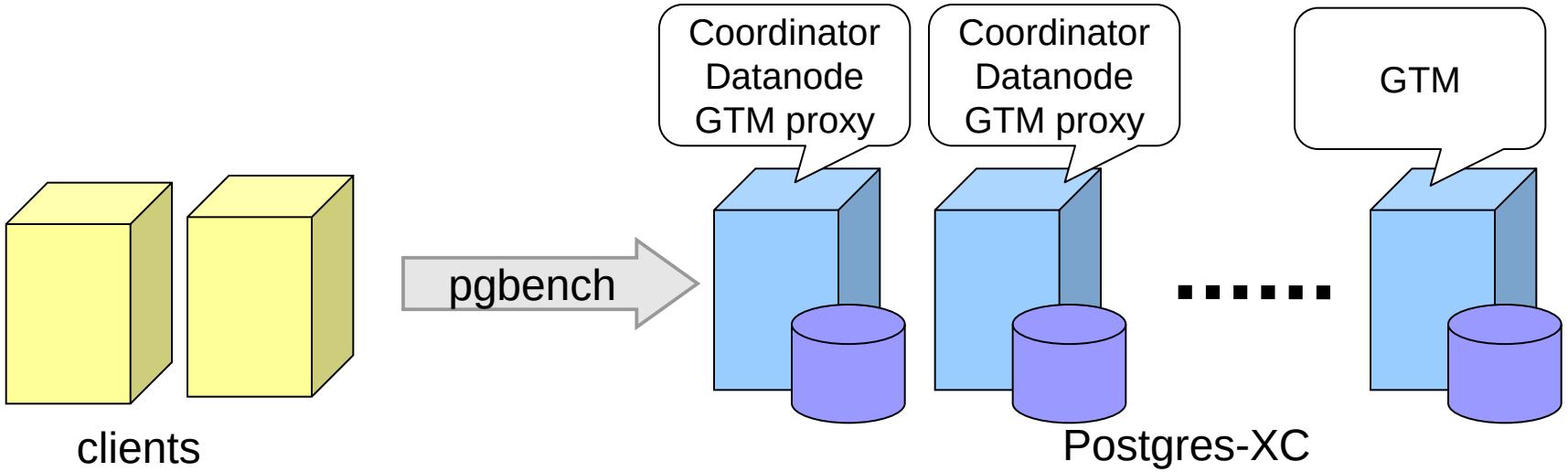
Coordinator	クライアントからの接続を受け付け、参照・更新要求をDatanodeに要求し、処理結果を取りまとめて返却する。
Datanode	実際にデータが格納され、参照・更新処理を実行する。
GTM	データベースクラスタ全体のトランザクション管理を行う。
GTM Proxy	GTMへの通信負荷を軽減するためのプロキシ。

Postgres-XC によるスケールアウト検証

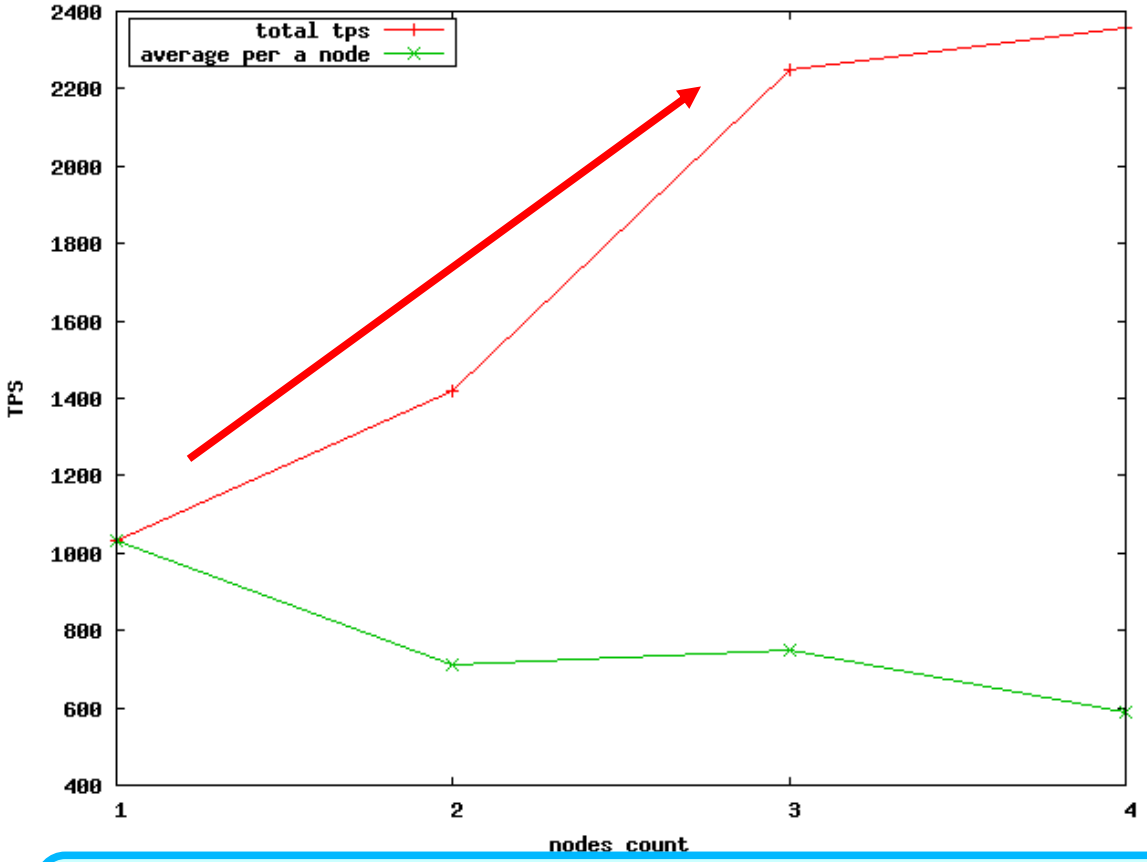
- 【 目的 】Postgres-XC 1.0.1(PostgreSQL 9.1.5 がベース)で、ノード数を増やすと全体の処理能力が向上するかどうかを確認する

【 方法 】pgbench の各ノードでの結果を合計する。

- pgbench はクライアント用のマシンから実行する。
- 使う pgbench は Postgres-XC 同梱の改造版。-k オプションをつけて初期化するとテーブルの分割を行なう。



更新系の結果

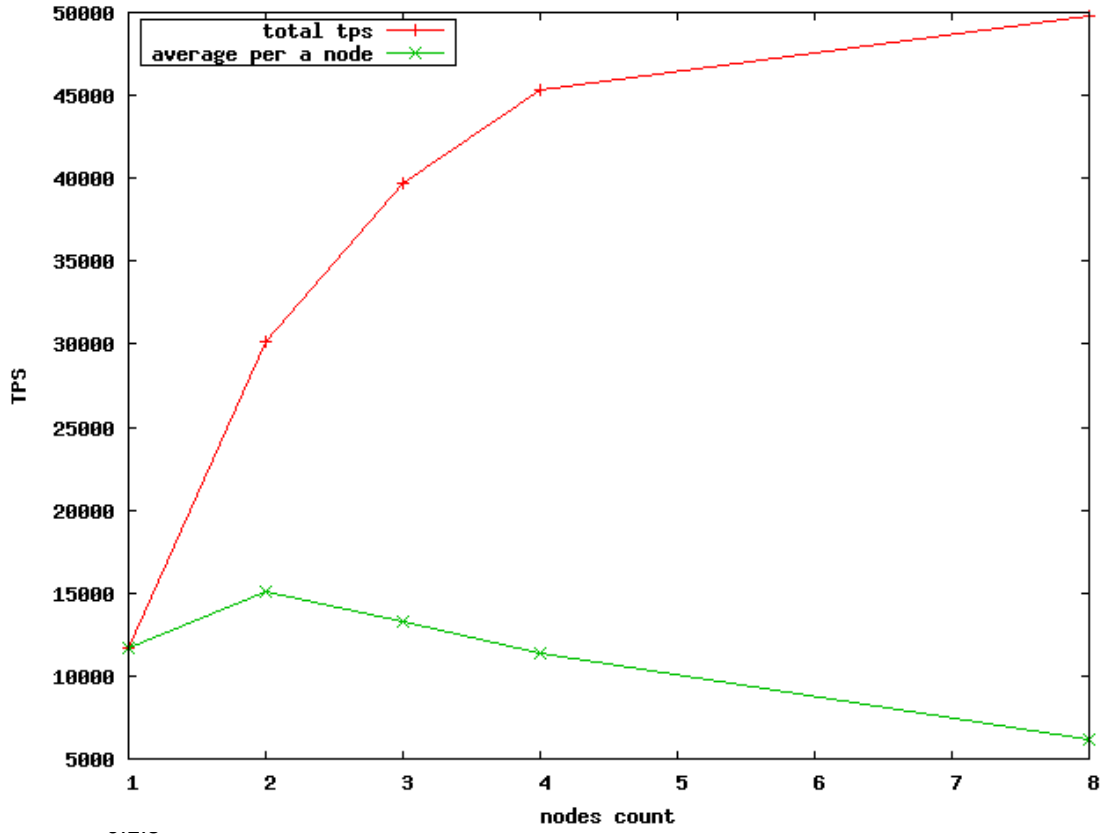


pgbench -i -k
-s 100
(total: 1.5GB)

pgbench
-c 100 -j 10
-n -k
-T 600

2 ノード以上では、ノード数が増えるほど合計の tps が増える。

参照系の結果(pgbench -n -k -S)



pgbench -i -k
-s 1000
(total: 15 GB)

pgbench
-c 100 -j 10
-n -k -S
-T 300

ノード数が増えるほど、合計の tps が増える。ただし、データが複数のDatanodeに分散されているため、参照系では性能が出にくい。

今後の予定

- 2013年3月まで
  - 検証結果の精査
  - 考察と結果のとりまとめ
- 結果報告
  - 2013年4月（予定）のPGEConsイベントにて結果を発表
  - 検証結果、手順書、スクリプトなどは再利用、再配布可能なライセンスを設定して公開予定
- 2013年度も取り組みを続けてまいります
  - 現在実施計画を議論中

まとめ

- スケールアップ検証
  - 従来公表されている64コアを上回る80コアでの検証を実施
  - 検索処理／更新処理ともにPostgreSQLが高いCPUスケーラビリティを持つことを確認できた
- スケールアウト検証
  - PostgreSQLカスケードレプリケーション、pgpool-II、Postgres-XCという異なるスケールアウトソリューションを検証
    - カスケードレプリケーションはノード数が増えても安定した更新性能を維持
    - Pgpool-IIは参照系に強み
    - Postgres-XCは更新系に強み

付録：今回使用した検証環境について

- スケールアップ検証では日本ヒューレット・パッカード株式会社様から検証環境をご提供いただきました
- スケールアウト検証では株式会社日立製作所様、日本電気株式会社様から検証環境をご提供いただきました
- PostgreSQLエンタープライズ・コンソシアムとして御礼を申し上げます

付録：検証環境１（提供：日本ヒューレット・パッカード株式会社）

■スケールアップ検証での使用機器／設備

HPソリューションセンター

HP ProLiant DL980G7

CPU: Xeon E7-4870(10Core) × 8 [80Core]

Memory: 2TB / HDD: 900GB x 8

HP ProLiant DL360pGen8

CPU: Xeon E5-2690(8Core) × 2 [16Core]

Memory: 64GB / HDD: 300GB 10k x 4

Network Switch

FC Switch

FC Switch

VPN

VRAID5 [20]

VRAID1 [20]

VRAID5 [20]

VRAID0 [20]

VRAID5 [20]

HP P6550 Enterprise Virtual Array

HDD: 900GB (6G SAS 10K 2.5inch) x 100 [20本×5セット]

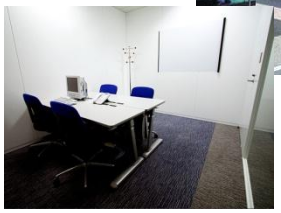
インターネット経由
リモートアクセス

付録：検証環境2 （提供： 株式会社 日立製作所）

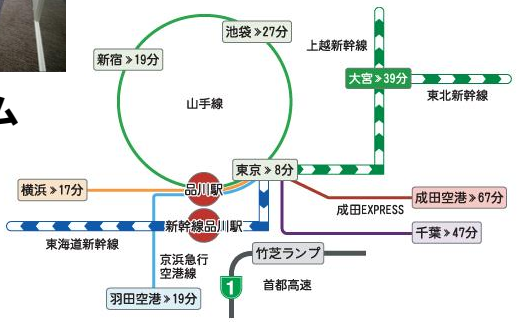
■ スケールアウト検証での使用機器／設備

ハーモニアス・コンピテンス・センター

サーバルーム



検証ルーム



JR品川駅　港南口より徒歩約3分
京浜急行品川駅　高輪口より徒歩約5分

★BS500：6ブレード
(BladeSymphony BS520H)

PostgreSQLサーバとして使用
CPU:インテルXeon E5-2670 (2.60GHz) 8コア×2
メモリ:64GB
HDD:600GB×2 RAID1

★AMS2000：1台
(Hitachi Adaptable Modular Storage 2100)

ストレージとして使用
450GB(SAS 15,000min-1)×15
RAID5 (4D+1P)

★HA8000：2台
(HA8000/RS210AL)

負荷クライアントとして使用
CPU:インテルXeon X5675 (3.06GHz) 6コア×2
メモリ:16GB
HDD:600GB×4 RAID6

付録：検証環境３（提供：日本電気株式会社）

■ スケールアウト検証での使用機器／設備

NECプラットフォームイノベーションセンター

- Server & Client
  - Express5800/B120d Blade 12台
    （DBサーバ 10台 - 負荷クライアント 2台）
    - CPU：Xeonプロセッサー E5-2470
      （2.30GHz 8Core 20M） × 2 socket
    - メモリ：32GB（8GB ×4）
    - HDD：300GB（2.5型SAS 15,000rpm） ×2
      - RAID1
    - 1000BASE-T：4ポート（標準の2ポート含む）
    - Fibre Channelコントローラ（8Gbps／2ch） x 1

- Storage （データベース格納領域）
  - iStorage/M300 1台
    - コントローラ数： 2台（FCコントローラ）
    - キャッシュメモリ： 16GB
    - FCポート数： 8個（8G対応 4ポート／1コントローラ）
    - ディスク：SAS 3.5型 15krpm 600GB ×20玉
      - RAID5（4D + P）

PGECons

PostgreSQL Enterprise Consortium