PostgreSQL インサイド

パーティショニングにおける性能改善への取り組み

－富士通の技術者に聞く！PostgreSQL の技術－

データを複数に分割して保持・管理する「パーティショニング」機能は、データベースを利用している技術者の方にはおなじみの機能でしょう。データを分割することで、性能や運用性が向上するとともに、故障の影響を局所化できます。
代表的なオープンソースデータベースである PostgreSQL でも、バージョン 10 より「宣言的パーティショニング」が実装されています。実装当初は「パーティションを大量に作成するとトランザクション性能が出ない」と言われていましたが、PostgreSQL 12 ではトランザクション性能が飛躍的に向上しています。
本特集では、PostgreSQL のパーティショニングにおける性能改善への取り組みについて当社の「加藤 翔」と「今井 良一」が語ります。

加藤 翔 Sho Kato
富士通株式会社　ソフトウェア事業本部 データマネジメント事業部
専門分野：データベース
2013 年から PostgreSQL をベースとした「FUJITSU Software Enterprise Postgres」のサポート業務に従事。2018 年からは、同製品の開発に携わるとともに、PostgreSQL コミュニティーの活動にも参加。

今井 良一 Yoshikazu Imai
富士通株式会社　ソフトウェア事業本部 データマネジメント事業部
専門分野：データベース
入社時より PostgreSQL に携わった業務に従事し、2018 年からは PostgreSQL コミュニティーの活動にも参加。

PostgreSQL におけるパーティショニング

“AI・機械学習”や“IoT”に代表されるようにデータベースで扱うデータ量が飛躍的に増加している現在、データベースにおけるデータ管理技術については、関心の集まるところと思います。PostgreSQL でもパーティショニングが導入されましたが、いかがでしょうか？

加藤

そうですね。パーティションの作成に必要とされる基本的な機能が網羅されていて、とても使いやすいと感じます。例えば、サブパーティションの実装や、親パーティションにインデックスを作成すると子パーティションにも自動的にインデックスが作成されるといった点です。

今井

FDW と組み合わせることもできるため、商用データベースにも引けを取らない機能を実現していると言って良いと思います。

- FDW：Foreign Data Wrapper の略で、あるデータベースからその外部のデータへアクセスするための仕組みであり、PostgreSQL データベースと他のさまざまなデータを連携できる機能。

加藤

PostgreSQL 9 以前ではパーティショニングが実装されていなかったため「PostgreSQL でパーティションを使いたい」というお客様の要件に応えるために、擬似的なパーティショニングで対応していました。

疑似的なパーティショニングですか？

加藤

はい。まず、「継承」という機能でテーブルの親子関係を定義します。次に「CHECK 制約」でデータの範囲を定義します。さらに、INSERT 先を振り分ける「トリガー関数」を作成して、親テーブルに「トリガー」を定義することで実現していました。
しかし、この方法ではトリガーを経由するため、トランザクション性能が出ませんでした。また、パーティションの追加や削除を行うと、トリガー関数を再定義する必要があり、メンテナンスが大変でした。このため、お客様からもパーティショング機能の実装を望む声が多く、PostgreSQL 10 で実装されたときは有難いと感じました。

トランザクション性能が伸びない

結構大変だったのですね。しかし、PostgreSQL 11 以前はパーティション数が 100 を超えると、トランザクション性能が急激に悪化したと伺っています。

今井

そうですね。確かに PostgreSQL 11 以前は、パーティション数が 100 を超えた辺りから一気に性能が劣化しました。パーティション数が 100 を超えるのはそれほど珍しいことでもないため、PostgreSQL コミュニティーでも課題となっていました。

加藤

当社のお客様からも 1,000 を超えるパーティションで運用する基幹業務システムを PostgreSQL で構築したいという要件がありました。当社でベンチマークテストなどを繰り返して検証しましたが、UPDATE コマンドを使った 1 レコードを更新するようなワークロードでは秒間 7 トランザクションしか処理できず、基幹業務で使用するには厳しいと言わざるを得ないレベルでした。

性能改善に向けてひたむきに取り組む

確かに、時系列データを扱うシステムでは 1,000 を超えるような大量のパーティションを作成することもありますね。PostgreSQL コミュニティーでも課題になっていたということですが、どのように取り組まれたのですか？

今井

PostgreSQL コミュニティーで性能改善における議論の中心となっていたのは、SQL の構文をどのようなパターンで記述しても性能が改善されるようにすることでした。
どういうことかと言うと、A さんが提案した修正方法ではパターン 1 の構文は性能が上がるのにパターン 2 の構文では性能が上がらない、B さんの提案した修正方法ではパターン 2 の構文は性能が上がるのに、パターン 3 の構文では性能が上がらないということが発生していたのです。これがなかなか解決に至らず、時間を要しました。

加藤

富士通としても、この課題はぜひ解決したいものでした。というのも、当社のお客様には、PostgreSQL をミッションクリティカル領域で使用したいという方が少なからずいらっしゃいます。富士通には、このようなお客様の要件に応えるべく、PostgreSQL を発展させたいという思いがあります。このため、PostgreSQL コミュニティーで性能改善に取り組むにあたり、次のような当社独自の目標を設定しました。

- 秒間 1,100 トランザクションを超える
- パーティショニング使用時のボトルネックを抑える
- パーティション数をどれだけ増やしてもオーバーヘッドを一定にする

今井

最初は、問い合わせ処理のどこかにあるはずのボトルネックを見つけ出すことから始めました。
具体的には、プランナとエクゼキュータに目星をつけ、それぞれ INSERT コマンドや UPDATE コマンドなど、どの処理にボトルネックがあるのかを確かめ、サーバーログからそのフェーズを特定しました。さらに、サーバーログでは内部処理の関数のレベルまで突き止めることができないので、Linux の perf も使ってボトルネックのある関数を抽出しました。このとき、PostgreSQL のソースにも手を入れて、それぞれの実行時間を測定し、処理に時間がかかっている関数を特定しました。

- プランナ：SQL の問い合わせに対しデータの並び順や物理的な配置などの様々な統計情報を基に最適な実行計画を作る機能。
- エクゼキュータ：プランナで作成された実行計画を実際に処理する機能。
- perf：Linux の性能解析ツール。

加藤

この作業が大変で、出力されたログはあまりに膨大な情報のため目視ではとても解析しきれませんでした。そこで、ログを解析して統計情報としてまとめるツールを開発しました。

独自ツールを開発したのですか？

加藤

ツールと言っても大げさなものではありません。オープンソースソフトウェアを含め、求めている機能を持ったツールを探したのですが、存在しなかったため一から作成しました。
ツールを作成したことによりログの解析スピードは向上しましたが、やはり膨大なログを解析するのは簡単なことではありません。しかし、時間をかけて丁寧に解析した結果、問い合わせ処理全体の中からプランナとエクゼキュータに潜んでいたボトルネックを発見することができました。

今井

早速ボトルネックを発見したことを PostgreSQL コミュニティーにフィードバックしました。それを受け、PostgreSQL コミュニティーでは活発な議論が重ねられ、解決策を模索し、プログラムの修正が行われました。当社はテストやレビューを中心に参画していましたが、プログラムの修正量は約 4,000 行にものぼり、テストやレビューだけでも半年という膨大な時間がかかりました。

半年ですか！相当大変だったのですね。どんな点で苦労されたのですか？

今井

プランナはさまざまな機能が集約されているコア機能ですので、そこを修正するということは他の機能に影響を及ぼす可能性があります。そのため、ボトルネック解消のためのプログラム修正が、他の機能に影響を及ぼさないことを確認するのが非常に大変でした。

加藤

「プログラム修正に対する影響確認」と言うのは簡単ですが、今回のケースはコアの機能を全て把握していないと実行できない非常に難しい作業です。PostgreSQL コミュニティーでもプランナの機能に立ち入ることが出来る人はとても限られているため、この作業は困難を極めました。
当社には今井をはじめとしてコア技術に精通した人員がいるため、プランナの機能について隅々までテストとレビューをしっかり行うことができました。

今井さんは見たところお若いのですが、どのくらい PostgreSQL に携わっているのですか？

今井

2017 年に入社し、現在 3 年目です。入社 1 年目から PostgreSQL に携わった業務を行っていましたが、2 年目から PostgreSQL コミュニティーでの開発に携わっています。

入社 3 年目ですか！？それは、素晴らしいですね。

今井

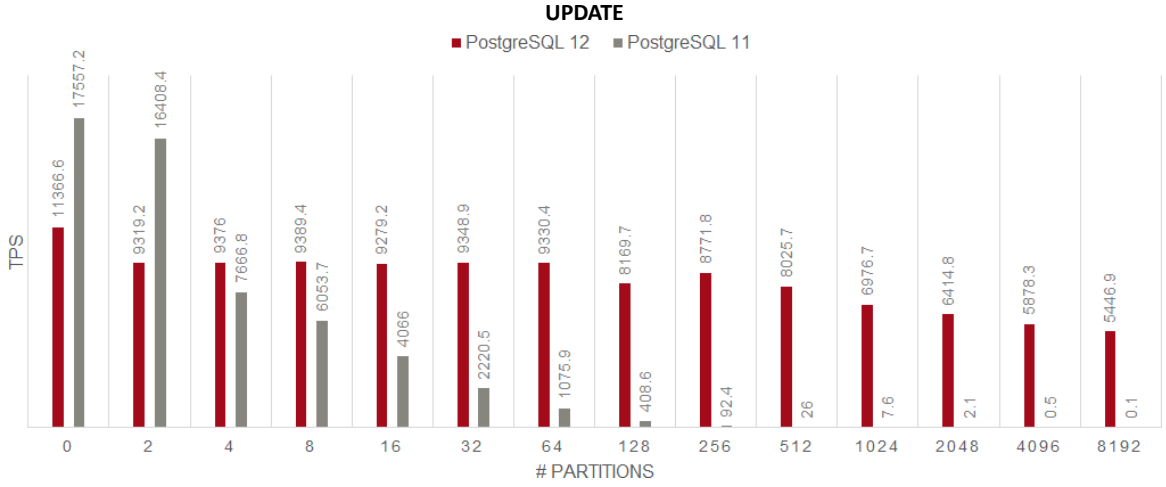
最初は非常に苦労しましたが、PostgreSQL コミュニティーに携わることでさまざまなスキルが身について、今回は性能改善にも貢献することができて、非常にやりがいを感じています。

PostgreSQL 12 で驚きの性能向上

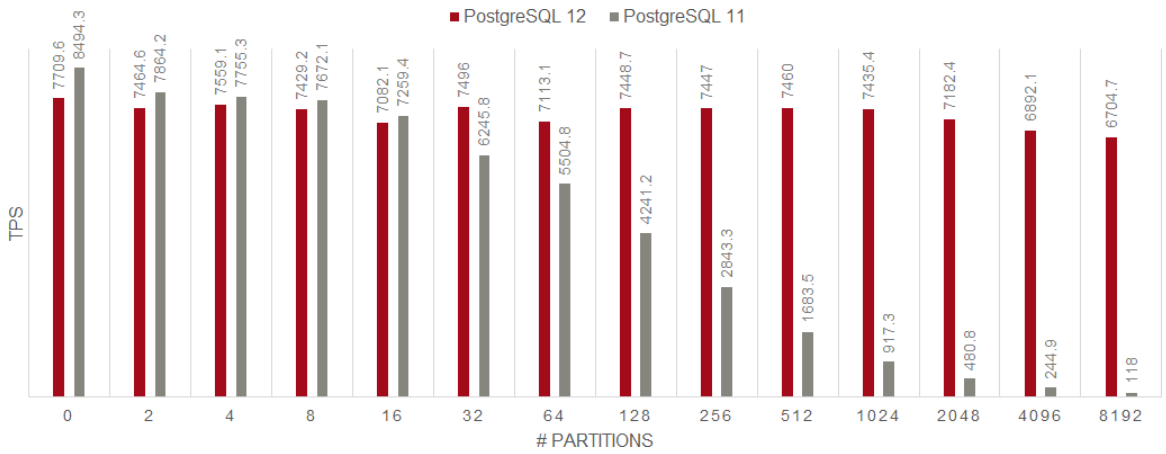
その性能改善ですが、さきほど性能改善にあたって目標を設定して取り組んだというお話でしたが、PostgreSQL 12 では目標に対して具体的にどのくらい性能が向上したのですか？

加藤

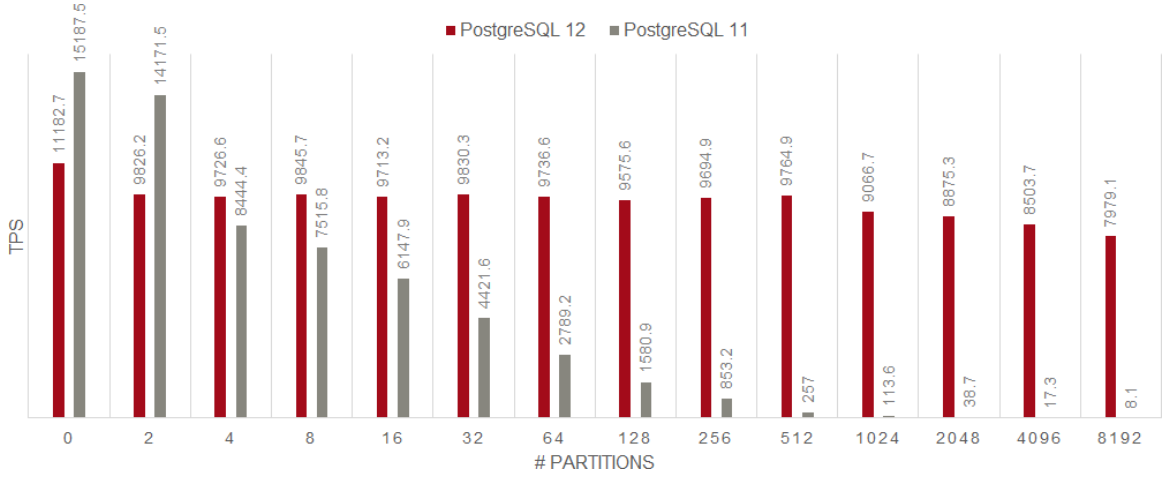
目標は PostgreSQL 12 でほぼ達成しています。
次のグラフを見てください。これは PostgreSQL 11 および PostgreSQL 12 で、SQL コマンドごとにベンチマークした結果をグラフ化したものです。「UPDATE」「SELECT」「INSERT」「DELETE」をシングルセッションで実行し、1 つのレコードを更新または参照したものです。縦軸がトランザクション数（TPS）、横軸がパーティション数を表しています。
ご覧いただくと一目瞭然ですが、パーティション数が多くなればなるほど PostgreSQL 11 に対して目を見張る性能向上が PostgreSQL 12 で実現できていることが分かります。加えて、パーティション数が増えてもトランザクション処理数の落ち込みが少ないということがお分かりいただけると思います。トランザクション処理数が変わらないということは「処理が安定している」ということであり、全体的に「性能が向上している」と言えます。



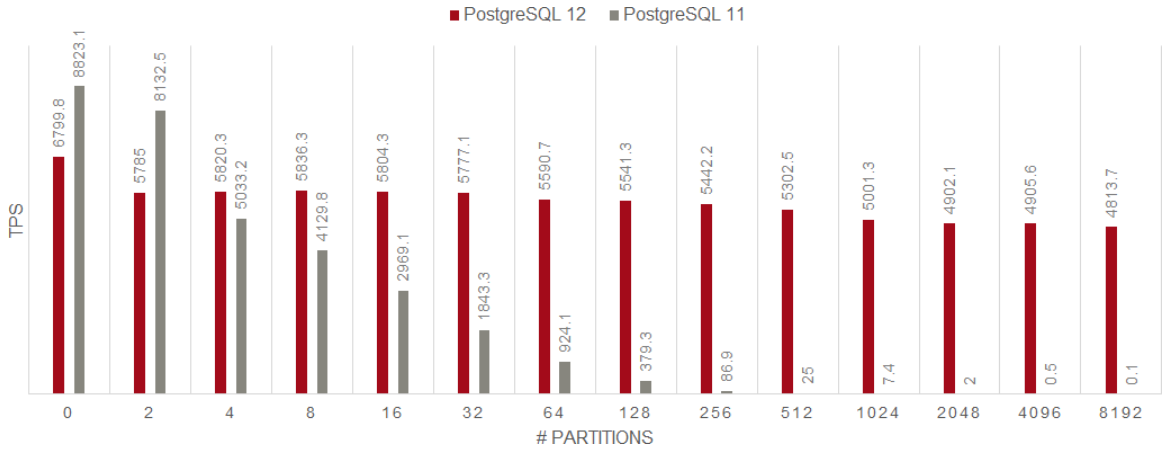
SELET



INSERT



DELETE



今井

当社のお客様で、1 つのテーブルを数千パーティションに分割して運用しているシステムがあるのですが、移行先として PostgreSQL に興味を持っていただいています。今回の性能改善で、このようなお客様の性能要件を満たすことができ、私たちとしても自信をもって PostgreSQL を提案できます。

目を見張る性能向上ですね！確かにこれなら大量のパーティションが必要なシステムでも、PostgreSQL が適用できそうですね。最後に、さらに改善したい点など、今後の取り組みについて教えてください。

加藤

今後は、さらにパーティショニングを使用した際のトランザクション性能向上を目指していきたいと考えています。具体的に大きく分けると 2 つの課題に取り組んでいます。

1 つ目は「汎用プランの作成時間短縮」です。
汎用プランは PREPARE 文を使用してキャッシュすることができ、そのキャッシュを利用することでプラン生成の処理をスキップできます。しかし、現状では最初のプラン生成に時間がかかります。これではせっかくのキャッシュ機能も十分に威力を発揮できないため、最初のプラン生成にかかる時間を改善したいと考えています。

2 つ目は「パーティション定義情報によるメモリー使用量の圧縮」です。
現状、大量のパーティションを作成すると、各セッションがそれぞれのパーティション定義情報を全てメモリーに蓄積していくため、メモリーが圧迫されてしまいます。これに対し、各セッションで保持しているパーティション定義情報のうち、同じパーティションの定義情報を共有することでメモリーの消費を抑えようとしています。
これらの機能は、PostgreSQL 13 に実装すべく当社が主体となって PostgreSQL コミュニティーと一緒に開発を進めています。

- 汎用プラン：PREPARE 文におけるバインド変数の実際の値を考慮せずに生成されるプラン。
- パーティション定義情報：目的のパーティションにアクセスするために必要なパーティションキーなどの情報。メモリーに蓄積しておくことで、素早く目的のパーティションへアクセスできる。

さらなる性能向上を目指していくということですね。ますます PostgreSQL の性能に注目が集まりますね。今日はありがとうございました。

2019 年 12 月 20 日