


MySQL 5.6 パラメータ検討会



MyNA(日本MySQLユーザ会)会 2013年7月
2013/07/29 平塚 貞夫
2013/08/06 Revision 2

自己紹介



- DBエンジニアやっています。専門はOracleとMySQL。
 - システムインテグレータで主にRDBMSのトラブル対応をしています。
 - 仕事の割合はOracle:MySQL:PostgreSQL=5:3:2ぐらいです。
- Twitter: @sh2nd
- はてな:id:sh2
-  ORACLE
ACE
- 写真は実家で飼っているミニチュアダックスのオス、アトムです。



本日のお題



MySQL 5.6のパラメータ検討

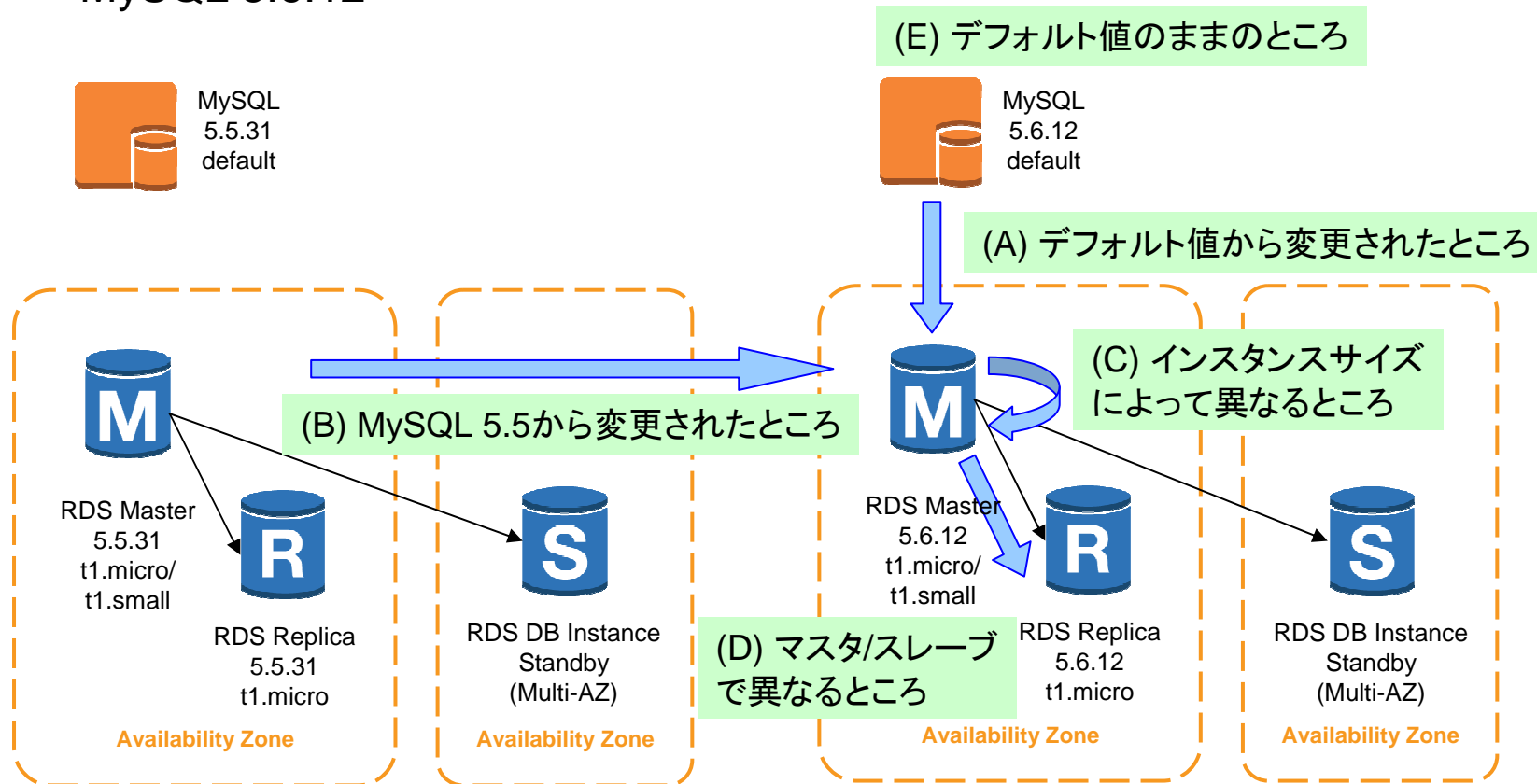


- MySQL 5.6にはさまざまな新機能が追加され、パラメータ数が大幅に増えました。
- SHOW GLOBAL VARIABLESの項目数で100個以上増えています。
 - MySQL 5.5.31:318個
 - MySQL 5.6.12:424個
 - 116個追加、10個削除
 - ちなみにOracle Database 12.1.0.1.0:367個
- もともとあまり役に立っていませんでしたが、my-large.cnfやmy-innodb-heavy-4G.cnfなどのサンプルファイルはMySQL 5.6では同梱されなくなりました。何も無いところからmy.cnfを書き上げるのは結構大変な作業ではないでしょうか。
- そこで本日はAmazon RDSの設定を参考にして、要注目パラメータを手早く確認していきたいと思います。
- 青字はRevision 2で追記、修正したところです。

調査方法



- Multi-AZ、リードレプリカ付きの構成でRDSのインスタンスを準備しました。
- 以下のバージョンで調査を行いました。
 - MySQL 5.5.31
 - MySQL 5.6.12



パラメーター一覧表



- 調査結果を表にまとめました。

No.	Variable name	MySQL 5.5.31					MySQL 5.6.12				
		5.5.31 default	different from default	different from t1.micro	different from t1.micro	different from t1.micro	5.6.12 default	different from 5.5.31	different from t1.micro	different from t1.micro	different from t1.micro
1	auto_increment_increment	1	1	1	1	TRUE	1	1	1	1	TRUE
2	auto_increment_offset	1	1	1	1	TRUE	1	1	1	1	TRUE
3	autocommit	ON	ON	ON	ON	TRUE	ON	ON	ON	ON	TRUE
4	automatic_sp_privileges	ON	ON	ON	ON	TRUE	ON	ON	ON	ON	TRUE
5	back_log	50	50	50	50	TRUE	80	56	74	56	TRUE
6	basedir	/usr	/rdsdbbin/mysql/	/rdsdbbin/mysql/	/rdsdbbin/mysql/	FALSE	/usr	/rdsdbbin/mysql/	/rdsdbbin/mysql/	/rdsdbbin/mysql/	FALSE
7	big_tables	OFF	OFF	OFF	OFF	(N/A)	OFF	OFF	OFF	OFF	(N/A)
8	bind_address	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	*	*	*	*	(N/A)
9	binlog_cache_size	32768	32768	32768	32768	TRUE	32768	32768	32768	32768	TRUE
10	binlog_checksum	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	CRC32	CRC32	CRC32	CRC32	TRUE
11	binlog_direct_non_transactional_updates	OFF	OFF	OFF	OFF	(N/A)	OFF	OFF	OFF	OFF	(N/A)
12	binlog_format	STATEMENT	MIXED	MIXED	STATEMENT	FALSE	STATEMENT	MIXED	MIXED	STATEMENT	FALSE
13	binlog_max_flush_queue_time	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	0	0	0	0	TRUE
14	binlog_order_commits	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	ON	ON	ON	ON	TRUE
15	binlog_row_image	(N/A)	(N/A)	(N/A)	(N/A)	(N/A)	FULL	FULL	FULL	FULL	FALSE

- デフォルト値 (青: MySQL 5.5と異なる)
- RDS t1.micro (赤: デフォルト値と異なる、青: MySQL 5.5と異なる)
- RDS t1.small (赤: t1.microと異なる)
- RDS t1.micro スレーブ (赤: マスタと異なる)
- RDSでの変更可否 (青: MySQL 5.5と異なる)

(A) デフォルト値から変更されているパラメータ



(A) デフォルト値から変更されているパラメータ (1/2)



- (back_log)
- (basedir)
- **binlog_format**
- (character_sets_dir)
- (datadir)
- **explicit_defaults_for_timestamp**
- (general_log_file)
- (have_openssl)
- (have_ssl)
- (host_cache_size)
- (hostname)
- (innodb_buffer_pool_size)
- (innodb_data_home_dir)
- **innodb_log_file_size**
- (innodb_log_group_home_dir)
- **key_buffer_size**
- (lc_messages_dir)
- (log_bin)
- (log_bin_basename)
- (log_bin_index)
- (log_error)
- (log_output)
- **log_slave_updates**
- **master_info_repository**
- **max_binlog_size**
- (max_connections)
- (performance_schema_max_cond_instances)
- (performance_schema_max_mutex_instances)
- (performance_schema_max_rwlock_instances)
- (performance_schema_max_socket_instances)
- (performance_schema_max_thread_instances)
- (pid_file)

(A) デフォルト値から変更されているパラメータ (2/2)



- (plugin_dir)
- (read_buffer_size)
- (read_rnd_buffer_size)
- (relay_log)
- (relay_log_basename)
- (relay_log_index)
- relay_log_info_repository
- relay_log_recovery
- (secure_file_priv)
- (server_id)
- (server_uuid)
- (slave_load_tmpdir)
- (slow_query_log_file)
- (socket)
- (ssl_ca)
- (ssl_cert)
- (ssl_cipher)
- (ssl_key)
- sync_binlog
- (system_time_zone)
- table_open_cache_instances
- (thread_cache_size)
- time_zone
- (tmpdir)
- (version)

※括弧付きは今回説明しないもの

binlog_format



	MySQL 5.5.31	MySQL 5.6.12
default	STATEMENT	STATEMENT
t1.micro	MIXED	MIXED

- バイナリログのフォーマットを指定します。STATEMENTはSQL文を記録する方式、ROWはレコードの変更内容を直接記録する方式、MIXEDはSTATEMENTで記録しつつSTATEMENTでは取り扱えない変更内容をROWで記録する方式です。
- STATEMENTには一部の非決定的な更新処理を正しく伝播できないという欠点があります。一方ROWにはバイナリログの容量がSTATEMENTよりも増えてしまうという欠点があります。一長一短ありますが、STATEMENTは今後使用しないことをおすすめします。
- またトランザクション分離レベルがREAD COMMITTEDの場合、STATEMENTを使用することはできません。エラーとなります。

explicit_defaults_for_timestamp



	MySQL 5.5.31	MySQL 5.6.12
default	(N/A)	OFF
t1.micro	(N/A)	ON

- MySQL 5.6の新規パラメータです。TIMESTAMP型を現在時刻が入る従来までの動作(OFF)にするか、明示的に指定しない限り入らない新しい動作(ON)にするかを指定します。OFFの場合はエラーログに以下の警告が出力されます。

```
[Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --
explicit_defaults_for_timestamp server option (see documentation for more details).
```

- MySQL 5.6からはDATETIME型に同じ機能を付けられるようになりました。TIMESTAMP型は将来のバージョンで廃止される予定です。

```
mysql> CREATE TABLE t1 (id INT PRIMARY KEY,
      dt DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP);
mysql> INSERT INTO t1 (id) VALUES (1);
mysql> SELECT * FROM t1;
+----+-----+
| id | dt                |
+----+-----+
|  1 | 2013-07-29 09:54:49 |
+----+-----+
```

innodb_log_file_size



	MySQL 5.5.31	MySQL 5.6.12
default	5,242,880 (5MB)	50,331,648 (48MB)
t1.micro	134,217,728 (128MB)	134,217,728 (128MB)

- InnoDBトランザクションログのファイルサイズを指定します。これとinnodb_log_files_in_groupを掛け算したものが実際のログファイルサイズになります。
- 値を増やすとInnoDBのデータ書き込み頻度を減らせるため、更新性能が向上します。MySQL 5.5までのデフォルト値5MBは小さすぎます。
- ただしinnodb_buffer_pool_sizeより大きくしても意味がないので、それ以下の値で調節します。MySQL 5.5までは2,047MB×2が最大で、上限いっぱい設定するケースもありました。
- 一方、値を増やすとチェックポイント発生時の性能低下が目立つ、mysqldの停止により長い時間がかかる、クラッシュリカバリにより長い時間がかかるといった副作用があります。RDSではMulti-AZにおけるフェイルオーバー時間やストレージの使用容量などを鑑みて、少し控えめな値にしていると推測されます。

key_buffer_size



	MySQL 5.5.31	MySQL 5.6.12
default	8,388,608 (8MB)	8,388,608 (8MB)
t1.micro	16,777,216 (16MB)	16,777,216 (16MB)

- MyISAMのインデックスをメモリ上にキャッシュしておく容量を指定します。
- RDSではデフォルト値の2倍に設定されていますが、理由は不明です。いずれにせよ非常に小さい値であり、RDSでMyISAMを使用することはあまり想定されていないことが分かります。

log_slave_updates



	MySQL 5.5.31	MySQL 5.6.12
default	OFF	OFF
t1.micro	OFF	ON

- バイナリログをスレーブでも出力するかどうかを指定します。レプリケーションを多段で構成する場合や、GTIDを用いたスイッチオーバー、フェイルオーバーを行う際に使用します。
- RDSではMySQL 5.6よりmysqlbinlogを用いたバイナリログアクセスを提供しています。オンプレミス環境とのデータの同期や、監査ログ、分析、レプリケーションエラーのデバッグといった用途が想定されています。これをスレーブに対しても提供するために、本パラメータが有効化されていると考えられます。

master_info_repository/relay_log_info_repository



	MySQL 5.5.31	MySQL 5.6.12
default	(N/A)	FILE
t1.micro	(N/A)	TABLE

- レプリケーションにおけるマスタへの接続情報とリレーログの実行情報について、ファイルに保存するかテーブルに保存するかを指定します。MySQL 5.6の新規パラメータです。
- スレーブをクラッシュセーフにするためには、これら2つのパラメータをいずれもTABLEに設定しておく必要があります。

relay_log_recovery



	MySQL 5.5.31	MySQL 5.6.12
default	OFF	OFF
t1.micro	OFF	ON

- レプリケーション開始時に、I/Oスレッドが取得したがSQLスレッドがまだ実行していないリレーログをいったん破棄し、再度I/Oスレッドに取得させるパラメータです。
- これはスレーブがクラッシュした際、リレーログの末尾が壊れている可能性を考慮したものです。スレーブをクラッシュセーフにするためには、本パラメータをONにする必要があります。

max_binlog_size



	MySQL 5.5.31	MySQL 5.6.12
default	1,073,741,824 (1GB)	1,073,741,824 (1GB)
t1.micro	134,217,728 (128MB)	134,217,728 (128MB)

- バイナリログのファイルサイズを指定します。innodb_log_file_sizeとは異なり、このパラメータは直接性能に影響するものではありません。
- バイナリログについては、扱いやすさを求めてファイルサイズを小さくすることがあります。例えば何かトラブルがあってバイナリログを直接確認しなければならなくなったとき、1GBのバイナリログをmysqlbinlogに通すと3GB程度になるため、出来のよいエディタでないとうまく取り扱えません。
- RDSはバイナリログを5分ごとにローテートしてS3にバックアップするアーキテクチャになっています。一つ一つのバイナリログがそれほど大きくなることを前提として、バックエンドの設計がなされているのではないのでしょうか。
- Linuxのext3ファイルシステムには大きなファイルの削除に時間がかかるという特性があり、バイナリログを小さくすることで負荷を分散できます。この特性はext4である程度改善されています。

sync_binlog



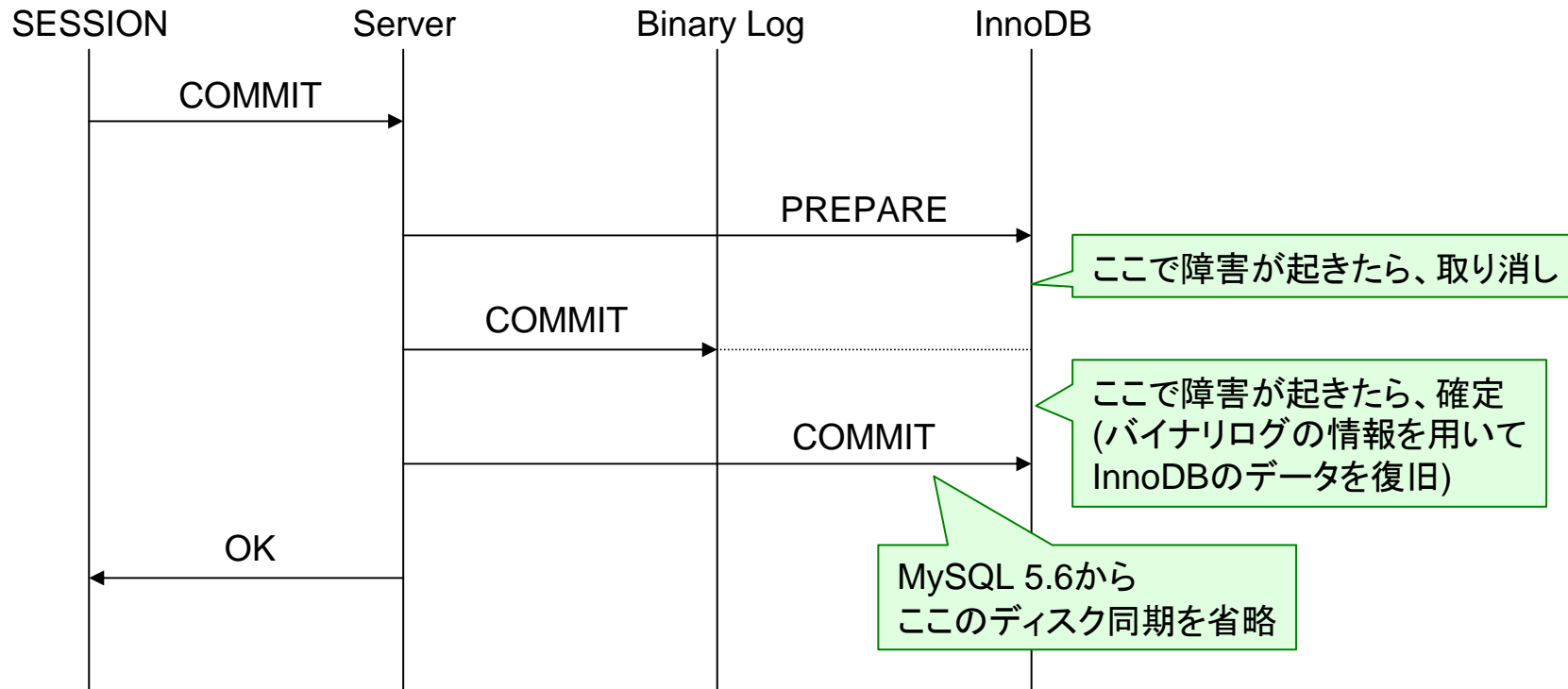
	MySQL 5.5.31	MySQL 5.6.12
default	0	0
t1.micro	0	1

- トランザクションをコミットした際に、バイナリログをディスクに同期するかどうかを指定します。
- RDSではMySQL 5.6から1に設定されており、さらにパラメータグループでの変更が不可となりました。理由は二つ考えられます。
 - コミットされたデータを保護するためにsync_binlog = 1が必須となったため。
 - MySQL 5.6においてsync_binlog = 1での性能が改善されたため。

2フェーズコミット



- バイナリログが有効な場合、InnoDBとバイナリログの整合性をとるために2フェーズコミットが行われています。

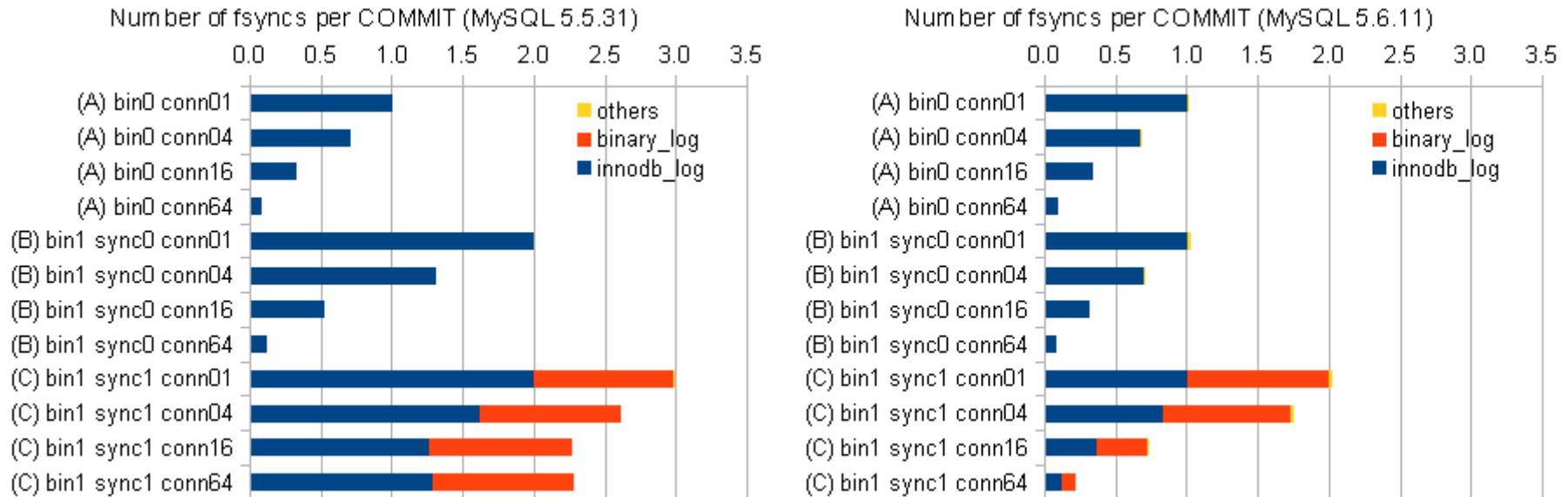


- トランザクションをコミットするたびに、ディスク同期が3回行われます。
- MySQL 5.6から、最後の1回が省略されてコミットあたり2回となりました。

コミットあたりのディスク同期回数



- バイナリログ、sync_binlog、多重度の設定を変えながら更新性能を測定し、SystemTapを用いてコミットあたりのディスク同期回数を数え上げました。



- (C)のところでコミットあたりのディスク同期回数が2回に減っていること、また多重度が高い場合にグループコミットが効いていることが分かります。
- MySQL 5.6で(B)のようにsync_binlog = 0に設定すると、バイナリログ、InnoDB双方のトランザクション確定情報がディスクに同期されず、障害発生時に最終トランザクションの内容を失います。当面この設定はしないでください(Bug#69162)。

table_open_cache_instances



	MySQL 5.5.31	MySQL 5.6.12
default	(N/A)	1
t1.micro	(N/A)	16

- table_open_cacheの分割数を指定します。分割することでDDL文以外の並列性が向上します。MySQL 5.6の新規パラメータです。
- リファレンスマニュアルに「CPUコアが16あるいはそれ以上のシステムでは、8または16を推奨します」と書かれており、RDSの値もこれにもとづくものです。
- 数千、数万といった、テーブル数の非常に多いシステムを見越して追加されたパラメータとのことです。

time_zone



	MySQL 5.5.31	MySQL 5.6.12
default	SYSTEM	SYSTEM
t1.micro	UTC	UTC

- 接続してきたクライアントの初期タイムゾーンを指定します。my.cnfで設定する場合のパラメータ名は、time_zoneではなくdefault_time_zoneです。
- RDSではUTCに設定され、変更不可となっています。UTC以外に変更したい場合は、init_connectパラメータで接続時に設定を上書きするワークアラウンドが用いられているようです。

(B) MySQL 5.5から変更されているパラメータ



(B) MySQL 5.5から変更されているパラメータ



- innodb_additional_mem_pool_size
- (innodb_autoextend_increment)
- innodb_buffer_pool_instances
- (innodb_concurrency_tickets)
- (innodb_data_file_path)
- innodb_flush_method
- innodb_old_blocks_time
- (innodb_open_files)
- (innodb_purge_batch_size)
- (innodb_purge_threads)
- (innodb_stats_on_metadata)
- (innodb_version)
- (join_buffer_size)
- (max_allowed_packet)
- max_connect_errors
- (old_passwords)
- (open_files_limit)
- (optimizer_switch)
- (performance_schema)
- (performance_schema_max_file_instances)
- (performance_schema_max_table_handles)
- (performance_schema_max_table_instances)
- (query_cache_size)
- query_cache_type
- (secure_auth)
- sort_buffer_size
- sql_mode
- (sync_master_info)
- (sync_relay_log)
- (sync_relay_log_info)
- (table_definition_cache)
- table_open_cache
- (version_comment)

innodb_additional_mem_pool_size



	MySQL 5.5.31	MySQL 5.6.12
default	8,388,608 (8MB)	8,388,608 (8MB)
t1.micro	2,097,152 (2MB)	8,388,608 (8MB)

- InnoDBがデータディクショナリなどの内部構造を保持するためのメモリ容量を指定します。
- `innodb_additional_mem_pool_size`は`innodb_use_sys_malloc`がOFFのときのみ指定する意味がありますが、現在`innodb_use_sys_malloc`はデフォルトONであり、このメモリ領域は自動割り当てされます。本パラメータはMySQL 5.6からDeprecatedとなっており、将来のバージョンで廃止される予定です。

innodb_buffer_pool_instances



	MySQL 5.5.31	MySQL 5.6.12
default	1	8
t1.micro	1	8

- InnoDBバッファプールの分割数を指定します。分割することでmysqld内部での並列性が向上します。
- MySQL 5.5ではデフォルト値が1に設定されていましたが、MySQL 5.6から8に変更されました。

innodb_flush_method



	MySQL 5.5.31	MySQL 5.6.12
default	(なし)	(なし)
t1.micro	O_DIRECT	(なし)

- InnoDBがファイルをopen(2)するときのフラグを指定します。Linuxの場合、このパラメータでOSのページキャッシュをバイパスするDirect I/Oの使用有無を決めます。InnoDBはディスク上のデータをキャッシュする機構を備えているため、デフォルト設定ではOSのページキャッシュとの二重管理になります。Direct I/Oを使用することでこの無駄を排除できます。
- ところが、RDSはMySQL 5.6からDirect I/Oの使用をやめたようです。
- Direct I/Oによる性能向上は最大で5%程度あります。一方OSのページキャッシュを使わないことから、mysqldを再起動した際にキャッシュのウォームアップに時間がかかるといったデメリットもあります。RDSがどのあたりを重視してデフォルト設定に戻したのかは分かりません。
- MySQL 5.6ではinnodb_buffer_pool_dump_at_shutdownとinnodb_buffer_pool_load_at_startupを使用して自動的にウォームアップを行わせることができます。RDSはこの機能を使用していません。

innodb_old_blocks_time



	MySQL 5.5.31	MySQL 5.6.12
default	0	1,000
t1.micro	0	1,000

- InnoDBバッファプールはOLD、NEWという2つの領域に分けられています。ディスクから読み込まれたデータはまずOLD領域に格納され、もう一度アクセスされるとNEW領域に移動します。
- 本パラメータを設定すると、指定した時間はデータがNEW領域に移動しないようになります。ごくまれにしかアクセスしないデータによって、よくアクセスされるデータが押し流されないようにする機能です。単位はミリ秒です。
- 意図的にInnoDBバッファプールをウォームアップするときにはかえって邪魔になりますので、必要に応じて設定を変更してください。

max_connect_errors



	MySQL 5.5.31	MySQL 5.6.12
default	10	100
t1.micro	10	100

- 特定ホストからの連続した接続エラーを許容する回数を指定します。この回数を超えると以後そのホストからの接続を拒否するようになります。解除するにはFLUSH HOSTSコマンドを使用します。
- DoS攻撃や、全然関係ないプログラムが設定誤りによりTCP 3306番に接続してくることに備えて設けられているパラメータです。MySQL 5.5ではデフォルト値が少なく暴発することがまれにあったのですが、MySQL 5.6で緩和されました。ただ、程度問題という気はします。

query_cache_type



	MySQL 5.5.31	MySQL 5.6.12
default	ON	OFF
t1.micro	ON	OFF

- クエリキャッシュの使用有無を指定します。
- MySQL 5.5までのデフォルト値はクエリキャッシュ有効&クエリキャッシュサイズ0でしたが、MySQL 5.6ではクエリキャッシュ無効に変更されました。クエリキャッシュを無効にした方が、mysqld内部での並列性が向上します。
- クエリキャッシュサイズの変更はオンラインでできますが、無効から有効への変更はmysqldの再起動を伴います。MySQL 5.6では基本的にクエリキャッシュは使わないことになったという理解でよいかと思えます。

sort_buffer_size



	MySQL 5.5.31	MySQL 5.6.12
default	2,097,152 (2MB)	262,144 (256KB)
t1.micro	2,097,152 (2MB)	262,144 (256KB)

- ソート用に確保できるメモリ容量を指定します。
- MySQL 5.6でデフォルト値が縮小されました。これまでのデフォルト値である2MBはほとんどのケースで大きすぎ、ソートのために2MBものメモリを割り当てることがかえってオーバーヘッドになっていたとのことです。
- デフォルト値は縮小されましたが、実際に大量データのソートが必要な場合はセッション単位でsort_buffer_sizeを設定することをおすすめします。ソート用のメモリが足りなかった場合はステータス変数Sort_merge_passesがカウントアップされるので、これを設定変更の目安にします。

sql_mode



	MySQL 5.5.31	MySQL 5.6.12
default	(なし)	NO_ENGINE_SUBSTITUTION
t1.micro	(なし)	NO_ENGINE_SUBSTITUTION

- SQLの動作モードを指定します。他のDBMSとの互換性を向上させるものなど、20個強のモードを組み合わせて指定することができます。
- MySQL 5.6でデフォルト設定されるNO_ENGINE_SUBSTITUTIONは、現在使用できないストレージエンジンを指定してCREATE TABLEを実行した際にエラーとして扱うモードです。これが指定されていない場合は、デフォルトのストレージエンジンが代わりに使用されます。
- これとは別に、MySQL 5.6インストール直後は/usr/my.cnfに以下の設定が施されています。必要がなければコメントアウトしてください。

```
sql_mode=NO_ENGINE_SUBSTITUTION, STRICT_TRANS_TABLES
```


table_open_cache



	MySQL 5.5.31	MySQL 5.6.12
default	400	2,000
t1.micro	400	2,000

- 同時にオープンできる最大のテーブル数を指定します。
- この数値はセッションごとに消費されるので、実際のテーブル数よりも大きな値が必要です。MySQL 5.1におけるデフォルト値は64で、ほとんどの環境で足りていませんでした。5.5、5.6とバージョンアップするごとにデフォルト値が増やされています。

(C) インスタンスサイズによって異なるパラメータ



(C) インスタンスサイズによって異なるパラメータ



- back_log
- host_cache_size
- (hostname)
- innodb_buffer_pool_size
- max_connections
- (performance_schema_max_cond_instances)
- (performance_schema_max_mutex_instances)
- (performance_schema_max_rwlock_instances)
- (performance_schema_max_socket_instances)
- (performance_schema_max_thread_instances)
- thread_cache_size

back_log



	MySQL 5.5.31	MySQL 5.6.12
default	50	80
t1.micro	50	56
t1.small	50	74

- mysqlがlisten(2)する際の接続キュー長を指定します。クライアントからの接続はmysqlによってaccept(2)されるまでキューの中で待たされます。キューがあふれるとクライアントにエラーが返されます。
- mysqlではコネクションマネージャスレッドが接続処理を行っています。コネクションマネージャスレッドは接続を受け付けるとコネクションスレッドを立ち上げて認証以降の処理を引き継ぎ、自身はすぐに次の接続をaccept(2)します。よほど高負荷な環境でない限り、back_logがあふれることはないと思います。
- MySQL 5.6ではback_logの値は $50 + \text{max_connections} \div 5$ という式で自動計算されるようになりました。RDSの値もこれにもとづくものです。
- back_logがある程度大きくなる場合、Linuxではカーネルパラメータnet.core.somaxconnをあわせて調節する必要があります。デフォルト値は128です。

host_cache_size



	MySQL 5.5.31	MySQL 5.6.12
default	(N/A)	279
t1.micro	(N/A)	160
t1.small	(N/A)	249

- ホスト名とIPアドレスの組をキャッシュするエントリ数を指定します。MySQL 5.6の新規パラメータです。
- デフォルトでは以下の式で自動計算され、RDSの値もこれにもとづくものです。MySQL 5.5までは128でハードコーディングされていました。
 - $128 + \text{max_connections}$ ($1 \leq \text{max_connections} \leq 500$)
 - $628 + (\text{max_connections} - 500) \div 20$ ($501 \leq \text{max_connections} \leq 27,940$)
 - 2,000 ($27,941 \leq \text{max_connections}$)
- MySQL 5.6から、Performance Schemaで中身を確認できるようになりました。

```
mysql> SELECT ip, host, count_host_blocked_errors, count_handshake_errors FROM host_cache;
+-----+-----+-----+-----+
| ip          | host  | count_host_blocked_errors | count_handshake_errors |
+-----+-----+-----+-----+
| 192.168.1.39 | k09s16 | 19 | 101 |
+-----+-----+-----+-----+
```

innodb_buffer_pool_size



	MySQL 5.5.31	MySQL 5.6.12
default	134,217,728 (128MB)	134,217,728 (128MB)
t1.micro	326,107,136 (311MB)	309,329,920 (295MB)
t1.small	1,179,648,000 (1,125MB)	1,145,044,992 (1,092MB)

- InnoDBがテーブルやインデックスのデータをメモリ上にキャッシュしておく容量を指定します。
- インスタンス自体のメモリサイズはt1.microで613MB、t1.smallで1.7GBとなっています。innodb_buffer_pool_sizeはメモリサイズの半分強を割り当てるのが一般的であり、RDSもそれにもとづいた設計となっています。
- MySQL 5.6ではPerformance Schemaがメモリを大量に消費するため、その分innodb_buffer_pool_sizeとsort_buffer_sizeが減らされていると考えられます。

```
mysql> SHOW ENGINE performance_schema STATUS;
+-----+-----+-----+
| Type          | Name                | Status      |
+-----+-----+-----+
...
| performance_schema | performance_schema.memory | 416620152 | (397MB)
+-----+-----+-----+
```

max_connections



	MySQL 5.5.31	MySQL 5.6.12
default	151	151
t1.micro	34	32
t1.small	125	121

- mysqlへ同時に接続できるクライアント数の上限値を指定します。
- デフォルト値の151は、かつてApacheのMaxClientsが150だったことから決められた値です(Bug#23883)。今は256になっているので、実はもう意味がありません。
- RDSではinnodb_buffer_pool_sizeにほぼ比例した設定となっています。
- それぞれの構成でinnodb_buffer_pool_size+key_buffer_size+max_connections×(sort_buffer_size+read_buffer_size+binlog_cache_size)+max_connections×thread_stackを計算すると、以下のようになります。
 - MySQL 5.5.31 t1.micro:413MB
 - MySQL 5.5.31 t1.small:1,457MB
 - MySQL 5.6.12 t1.micro:336MB
 - MySQL 5.6.12 t1.small:1,202MB

thread_cache_size



	MySQL 5.5.31	MySQL 5.6.12
default	0	9
t1.micro	0	8
t1.small	0	9

- 使い終わったコネクションスレッドを次の接続のために残しておく数を指定します。ApacheのMinSpareServersと似たパラメータです。
- MySQL 5.6から、デフォルトでは自動計算されるようになりました。RDSの値もこれにもとづくものです。
 - $8 + \max_connections \div 100$ ($1 \leq \max_connections \leq 9,200$)
 - 100 ($9,201 \leq \max_connections$)

(D) マスタ/スレーブで異なるパラメータ



(D) マスタ/スレーブで異なるパラメータ



- (binlog_format)
- (hostname)
- (log_bin)
- (log_bin_basename)
- (log_bin_index)
- **max_allowed_packet**
- (read_only)
- (server_id)
- (server_uuid)
- (version)

max_allowed_packet



	MySQL 5.5.31	MySQL 5.6.12
default	1,048,576 (1MB)	4,194,304 (4MB)
t1.micro	1,048,576 (1MB)	4,194,304 (4MB)
t1.micro slave	1,073,741,824 (1GB)	1,073,741,824 (1GB)

- BLOB、TEXT型について送受信可能な最大データサイズを指定します。MySQL 5.6からデフォルト値が4MBに引き上げられました。
- RDSではスレーブのみ最大値の1GBに引き上げられています。マスタで設定を変更したあと、スレーブでの設定漏れを防ぐ意図があるのではないかと考えられます。read_onlyのスレーブにはマスタにINSERTされたよりも大きなLOBデータは存在しないため、マスタと同じ値に絞っておく必要はありません。

(E) デフォルト値から変更されていないパラメータ



character_set_server



	MySQL 5.5.31	MySQL 5.6.12
default	latin1	latin1
t1.micro	latin1	latin1

- サーバのデフォルトキャラクタセットを指定します。
- アプリケーションの要件によりますが、特別な事情がなければ今後はutf8mb4に設定することをおすすめします。

gtid_mode



	MySQL 5.5.31	MySQL 5.6.12
default	(N/A)	OFF
t1.micro	(N/A)	OFF

- グローバルトランザクションIDの使用有無を指定します。MySQL 5.6の新機能です。
- MySQL 5.6の目玉機能ですが、RDSでは使用していません。高可用性はMySQLのレプリケーションに頼らずMulti-AZで実現していること、gtid_modeではいくつかのSQLが使えなくなってしまうことが要因ではないかと考えています。

innodb_io_capacity



	MySQL 5.5.31	MySQL 5.6.12
default	200	200
t1.micro	200	200

- InnoDBが更新済みページを書き出す速度をIOPSで指定します。
- リファレンスマニュアルに「per buffer pool instance」と記載されていますが、[storage/innobase/buf/buf0flu.cc](http://storage.innobase.com/buf/buf0flu.cc)のbuf_flush_list()を見る限り、全インスタンスの合計値で指定するものです。
- MySQL 5.6ではAdaptive Flushingアルゴリズムが強化されており、チェックポイントが遅れると書き出しIOPSがinnodb_io_capacity_maxを上限として自動調節されます。そのためinnodb_io_capacityの設定値はMySQL 5.5から引き継がず、innodb_io_capacity_max、innodb_adaptive_flushing_lwmとあわせて再検討されることをおすすめします。

slow_query_log



	MySQL 5.5.31	MySQL 5.6.12
default	OFF	OFF
t1.micro	OFF	OFF

- スロークエリログの出力有無を指定します。
- 管理の手間は増えますが、基本的に出力しておくことをおすすめします。
- RDSではlog_outputパラメータがTABLEに設定されており、スロークエリはmysqlデータベースのslow_logテーブルに出力されます。

宿題



1. ご自身の関わっているシステム、サービスで`explicit_defaults_for_timestamp`を有効にしてアプリケーションの挙動に問題がないか、確認してください。
2. ご自身の関わっているシステム、サービスで`sort_buffer_size`を256KBに変更して性能上の問題がないか、確認してください。
3. ご自身の関わっているシステム、サービスで`sync_binlog`を1に設定して性能上の問題がないか、確認してください。
4. MariaDBの最新版はバイナリログの設計がMySQL 5.6と異なっています。コミットされたデータを保護するために`sync_binlog = 1`が必須かどうか、調べてみてください。

