



# libev+OpenSSL =EvServer (PostgreSQL編その後)

Future Versatile Group/MyDNS.JP  
T.Kabu



# ▶ 簡単に振り返り...

- **Postgres用マルチメインDB**の構築方法を考察
  - PgBouncerの内部構造を解析
- libev+OpenSSLで **イベントドリブン**にするよ
  - スケルトンは比較的簡単にできた = EvServer
  - FastCGI対応のhttpdも作った
- まずは**実際のプロトコル**がみたい(デバッグも楽に)
  - PostgreSQLのプロトコルアナライザ  
= PgAnalyzer



# ▶ ざっくり作ってみたら

- ・ 昨年のデモの通り、うまく動いた
- ・ ログの時系列出力がいまいち
- ・ てか、ログの出力自体の負荷がデカイ
- ・ BASE64 関連は本家からのコピーだし...
- ・ SASL 認証 (sha256 認証) はスパゲッチだし...

本音 = 公開するのどうしようかなー



# ▶ ログ出力の見直し

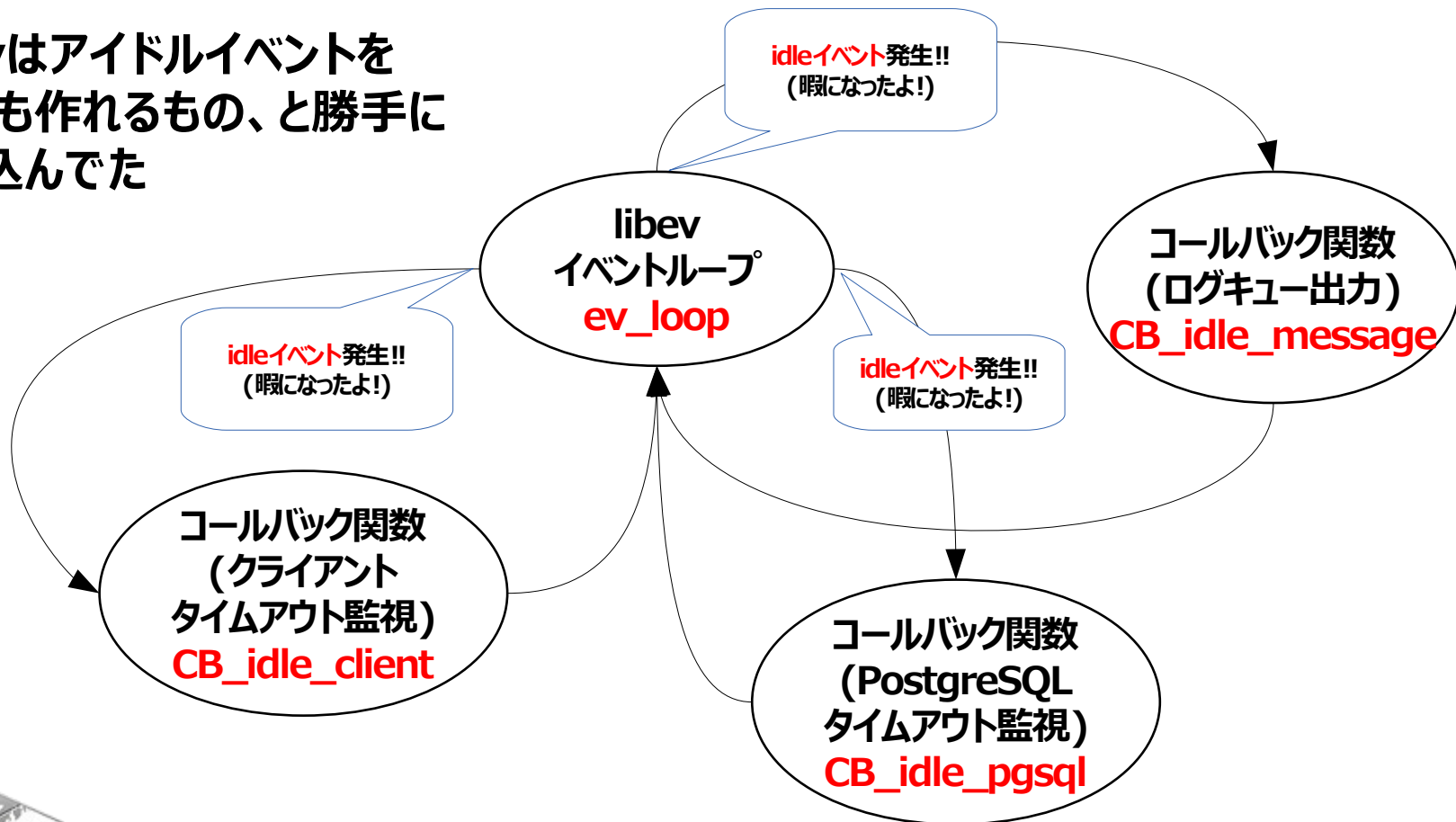
- ・リアルタイム出力だと、オーバーヘッドが...
- ・ログ出力をキューに溜める方式に
  - 処理が暇になったらキューから取り出して出力
- ・ログ出力がデバッグモード(大量)ならスムーズ
- ・でもログ出力が少ないと、出力がいきまいち

もしかしてlibevの使い方間違えてる？



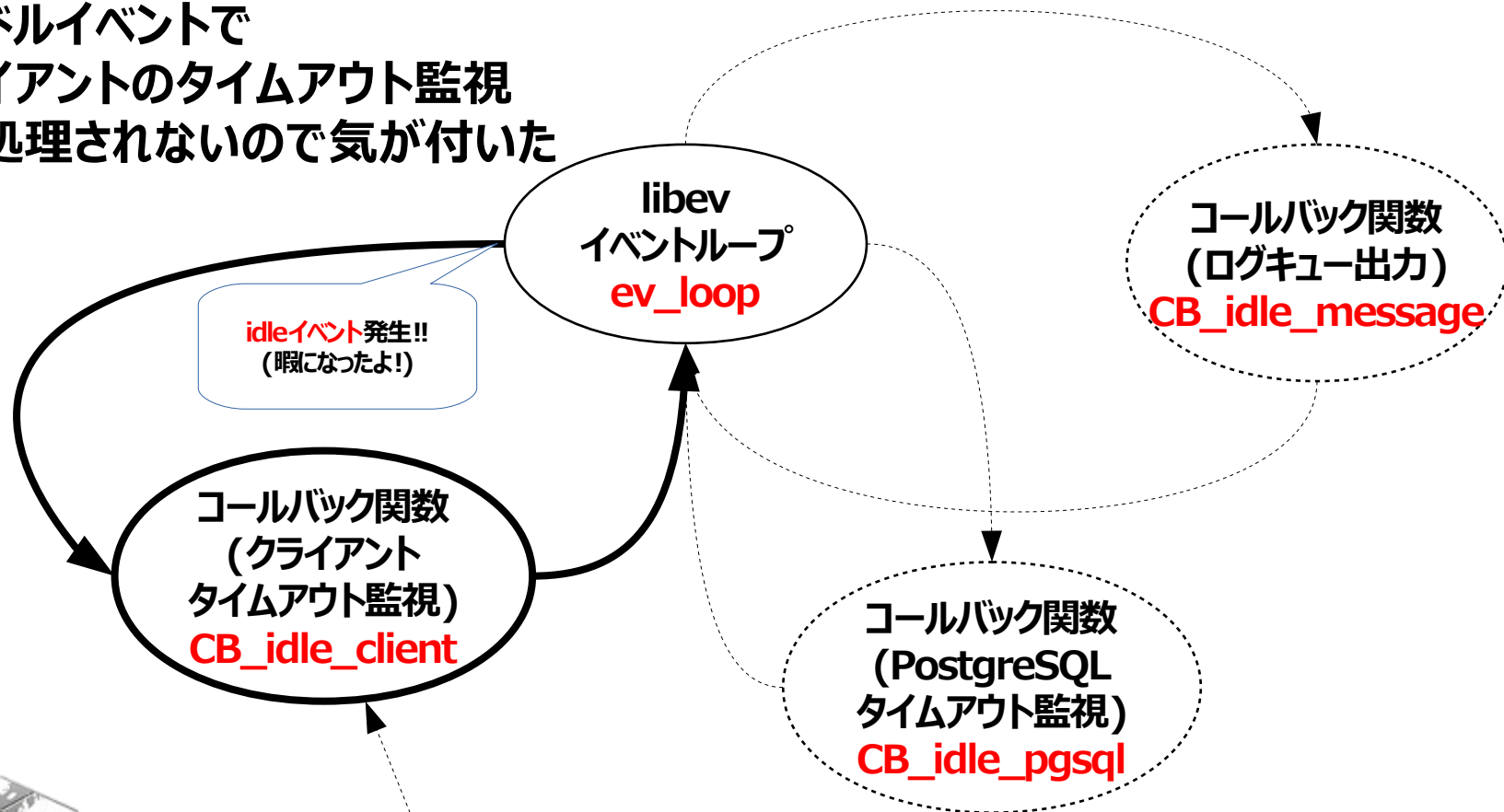
# ▶ libevのイメージ(以前)

libevはアイドルイベントを  
いくつも作れるもの、と勝手に  
思い込んでた



# ▶ libevのイメージ(以前)

アイドルイベントで  
クライアントのタイムアウト監視  
しか処理されないのが気が付いた

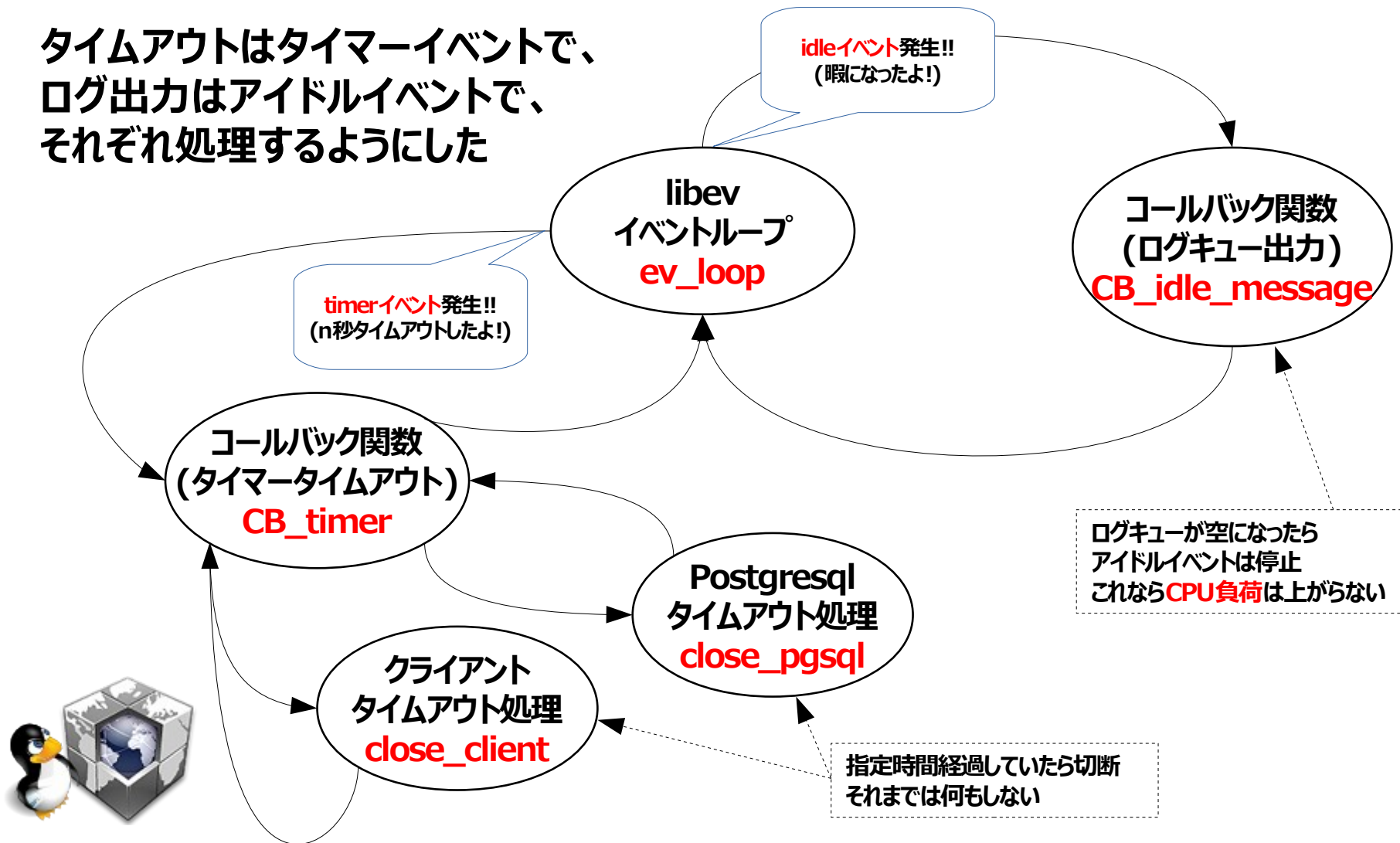


しかもめっちゃCPU負荷が高い!!  
そりゃそうか、暇だから呼ばれまくるし!!



# ▶ libevはこれが正解(?)

タイムアウトはタイマーイベントで、  
ログ出力はアイドルイベントで、  
それぞれ処理するようにした



# ▶そしてGitHubへ

- GitHubで『**PgAnalyzer**』として公開

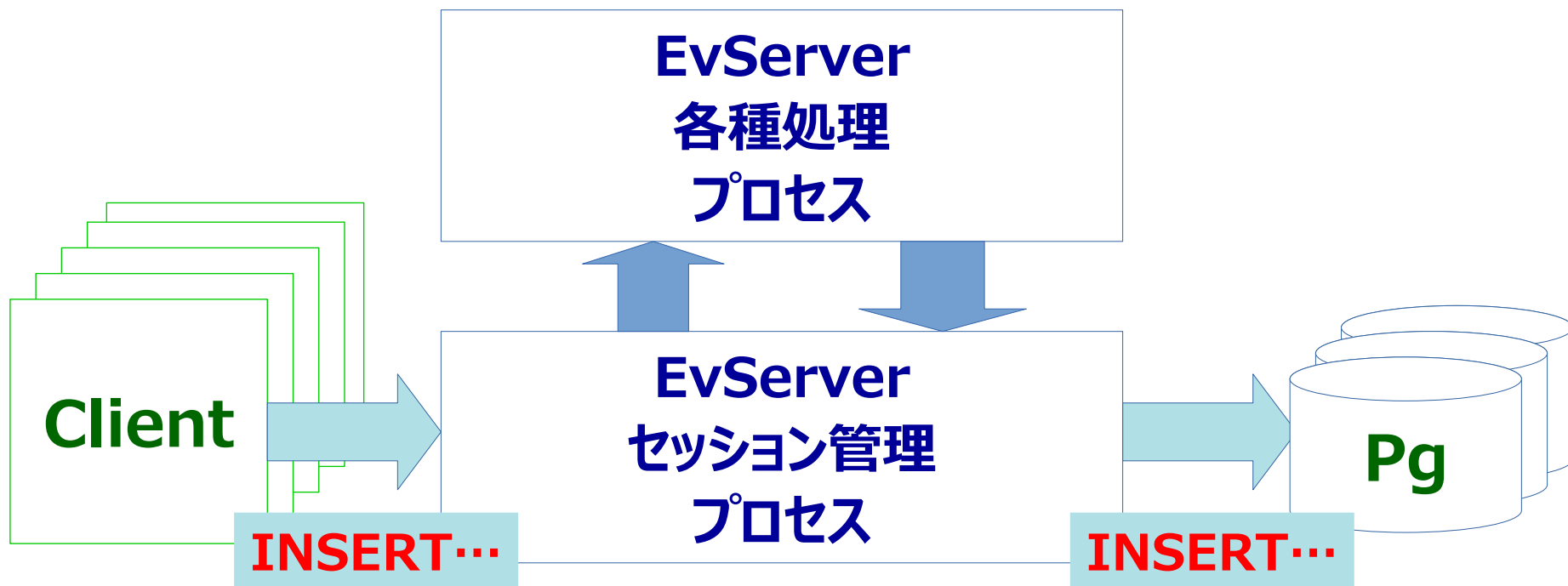
<https://github.com/disco-v8/PgAnalyzer>

- 例によって日本語による、ほぼ一行一コメント
- SASL認証 (sha256認証) はバッチイので削除
- 勉強するにはちょうどいい…のかな？感想歓迎！





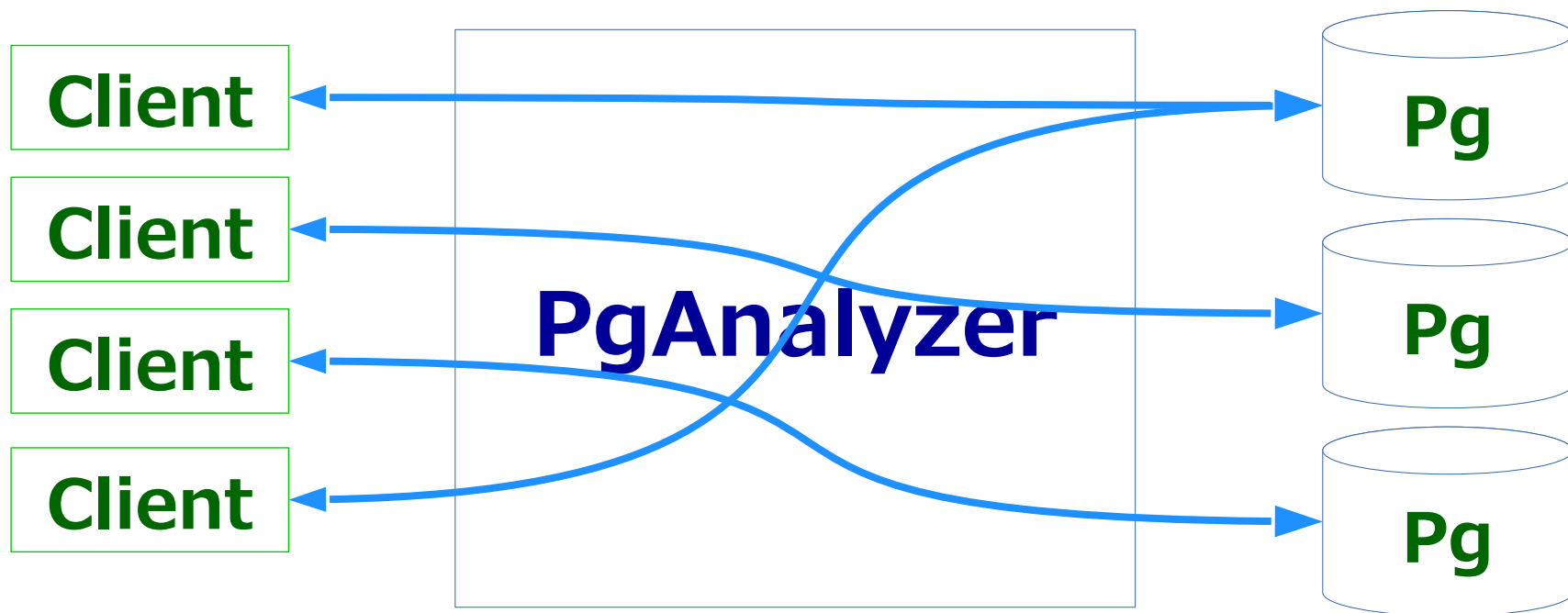
# ▶そして本題



PgAnalyzerは一段落したので、本来のマルチメインDBを。  
Clientからのクエリを受けたら、事前の設定と、クエリの内容により  
PostgreSQLに対してクエリを発行する。



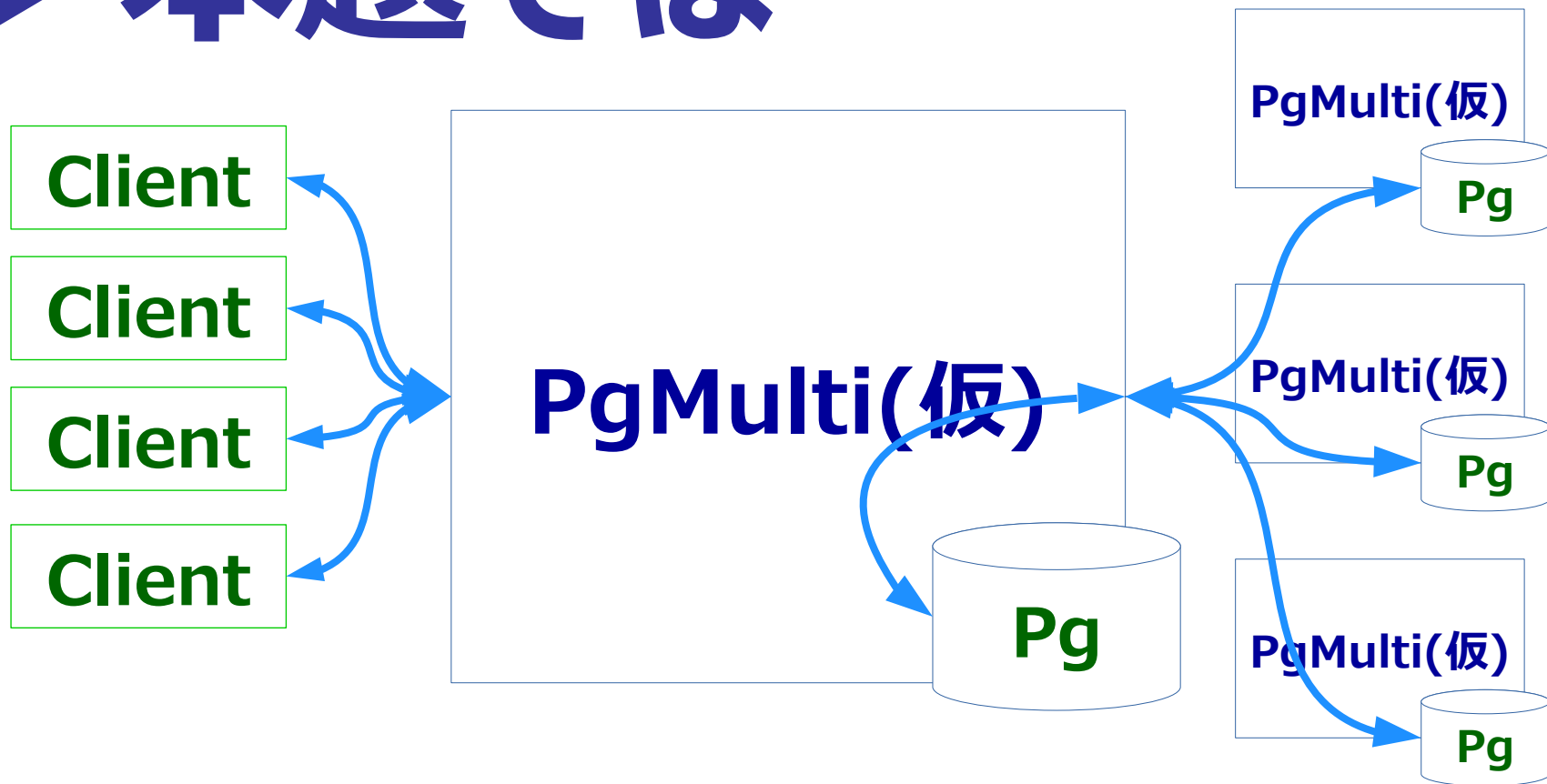
# ▶ PgAnalyzerは...



Clientから接続されると、PostgreSQLへの接続(セッション)情報を  
ひたすら作る。→同一PostgreSQLに対しての接続が複数になるので  
同一PostgreSQL(DB)への時系列の統制が取りにくいかも!?



# ▶ 本題では...



PostgreSQLへの接続(セッション)は、起動時に済ませておく。  
Clientから接続要求が来たら、PostgreSQLの代わりに接続処理。  
Clientからのクエリは、その内容により大きく処理の流れが分かれる。



# ▶マルチメインDB目指して

- PgMultiには必ず対となるPostgreSQLを用意
- 暗示的トランザクションと明示的トランザクション
  - BEGIN;の有無で「完全同期 / 非同期」モードとする
- 完全同期 (**SYNC**) モード
  - 全てのPgで ReadyforQuery(=OK)が返るまで待機
  - 全てのPgから RfQ が来たら、Clientに RfQ を返す
- 非同期 (**ASYNC**) モード
  - 対となるPgで ReadyforQuery(=OK)が返れば OK
  - Clientに RfQ を返す、他のPgの RfQ は待たない
  - でもエラーになったら全てを ROLLBACK?! (検討中)
  - RfQのあとの COMMIT;の結果を見る?! (検討中)



# ▶マルチメインDB目指して

その他にも…

- ・クライアントとPgMultiとPostgreSQLの関係
- ・PgMultiの負荷やデータの流れた考えたプロセス構成
- ・リカバリ(運用中のPgのパーティ、マージをどうするか!?)
- ・データベースの同一性チェック(パーティの判断方法)

…などなど、引き続きまったりと開発を進めます。



ご期待ください!?



# PgAnalyzerからPgMultiへ おしまい

Future Versatile Group/MyDNS.JP  
T.Kabu

