

PostgreSQL 最新情報

Open Source Conference 2024 Nagoya
2024-05-25 10:00 - 10:45 第三会議室 (2)
日本 PostgreSQL ユーザ会 名古屋支部 福島克輝

本日の話題：

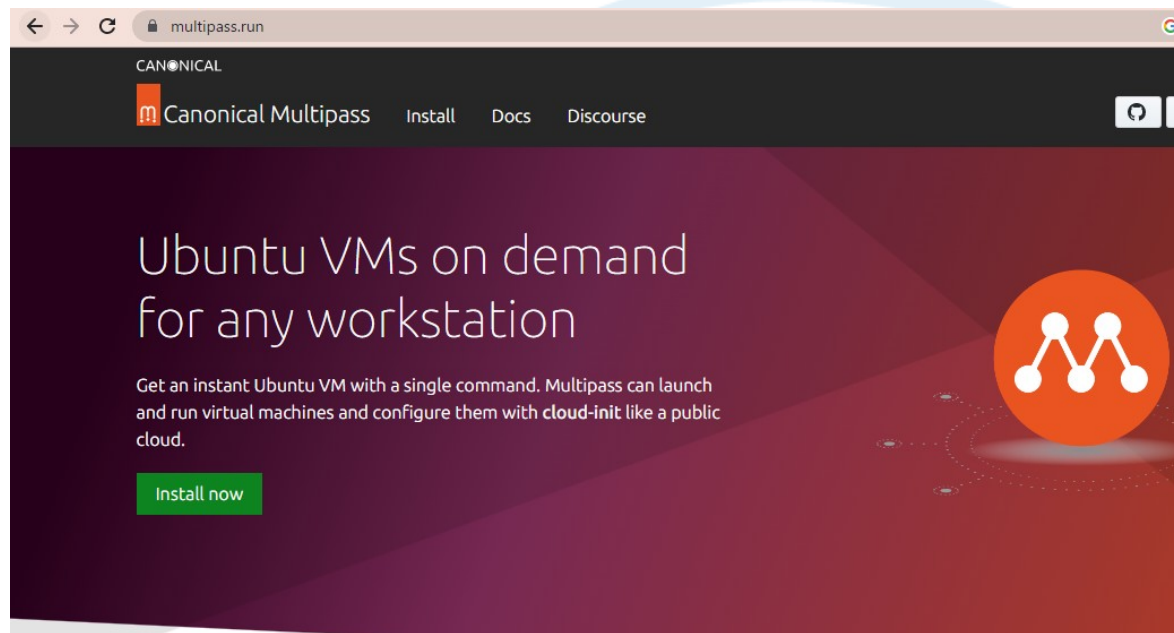
- 最新のスナップショット版を使うには
- Docker で PostgreSQL
- PostgreSQL 15 新機能おさらい
- PostgreSQL 16 新機能
- Etc.

講演者：

- 福島克輝（ふくしまかつあき）
- 日本 PostgreSQL ユーザ会名古屋支部
- 株式会社セイノー情報サービスにて仕事は主に技術支援

スナップショット版を使うには 仮想環境構築からビルド&インストールまで

- 仮想環境設定して、Linux 導入して、ビルド環境作って、ソース落としてきてビルドしてインストール…面倒くさそう…
- multipass を使って Ubuntu 入れて PostgreSQL ビルドするのは意外と簡単です！ <https://multipass.run/>



一連の流れはこんな感じ

- multipass で環境作成 -c CPU コア数 -d ディスクサイズ -m メモリ -n インスタンス名
multipass launch -c 4 -d 20G -m 2G -n pgsq
- ビルド環境構築
sudo apt -y update
sudo apt -y install build-essential bison flex libreadline-dev zlib1g-dev libicu-dev pkgconf language-pack-ja
- postgresql ユーザ作成 (導入先は /opt/pgsql へ)
sudo useradd --shell /bin/bash --create-home --home-dir /opt/pgsql postgres
sudo passwd postgres postgres ユーザにパスワードを割り当てるなら必要
echo "export PGDATA=/opt/pgsql/data" | sudo tee -a /opt/pgsql/.profile
pg_ctl や psql コマンド実行時の DB クラスタ指定のため
- ソースを取得して展開
wget https://ftp.postgresql.org/pub/snapshot/dev/postgresql-snapshot.tar.bz2
tar axf postgresql-snapshot.tar.bz2
- ビルド&インストール
cd postgresql-17devel
./configure --prefix=/opt/pgsql
make
sudo make install

Docker で PostgreSQL を使う

- Docker パッケージインストール

```
sudo apt update
```

```
sudo apt install apt-transport-https
```

```
sudo mkdir -p /etc/apt/keyrings
```

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o ¥  
/etc/apt/keyrings/docker.gpg
```

```
echo ¥
```

```
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
```

```
https://download.docker.com/linux/ubuntu ¥
```

```
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

```
sudo apt update
```

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

- PostgreSQL 実行（実質この1行だけ！）

```
docker run --rm -e POSTGRES_PASSWORD=password -d -p 5432:5432 postgres
```

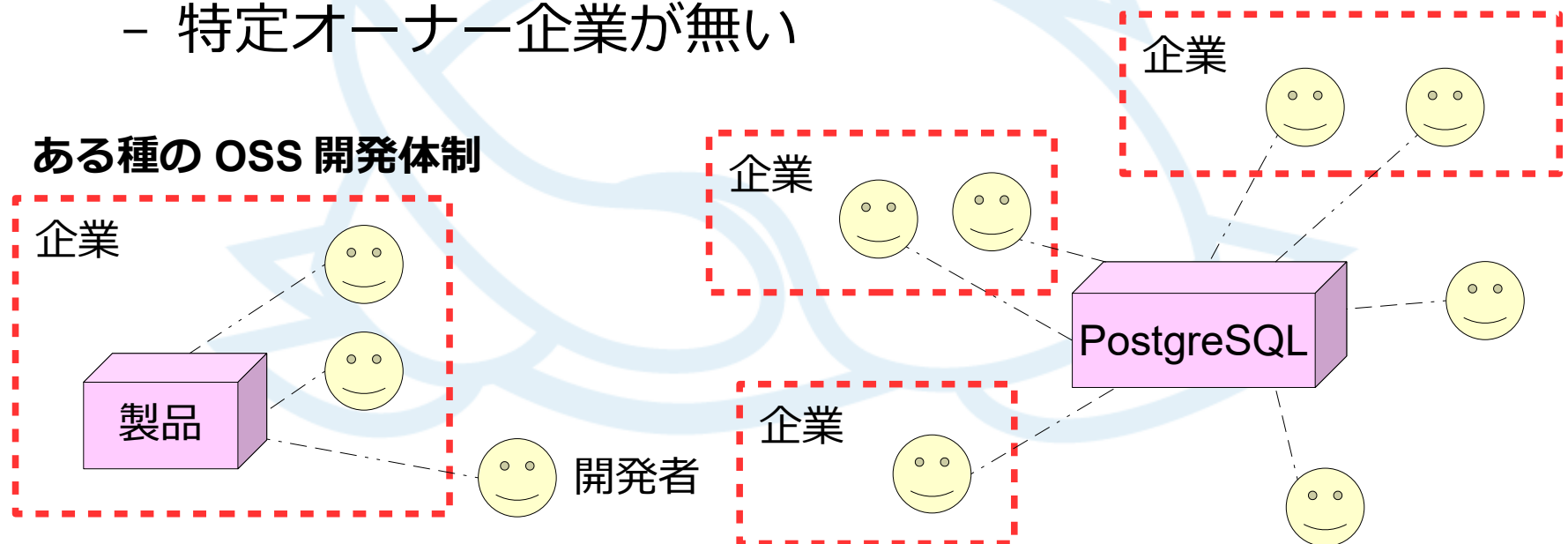
上記の例ではコンテナを終了するとデータも消えてしまうので、データを残したい場合は、オプションでローカルのマウント先を指定する必要があります。

またデフォルトでは最新版が実行されるため、古いバージョンを利用したい場合は指定が必要です。

オプションについては https://hub.docker.com/_/postgres を参照ください。

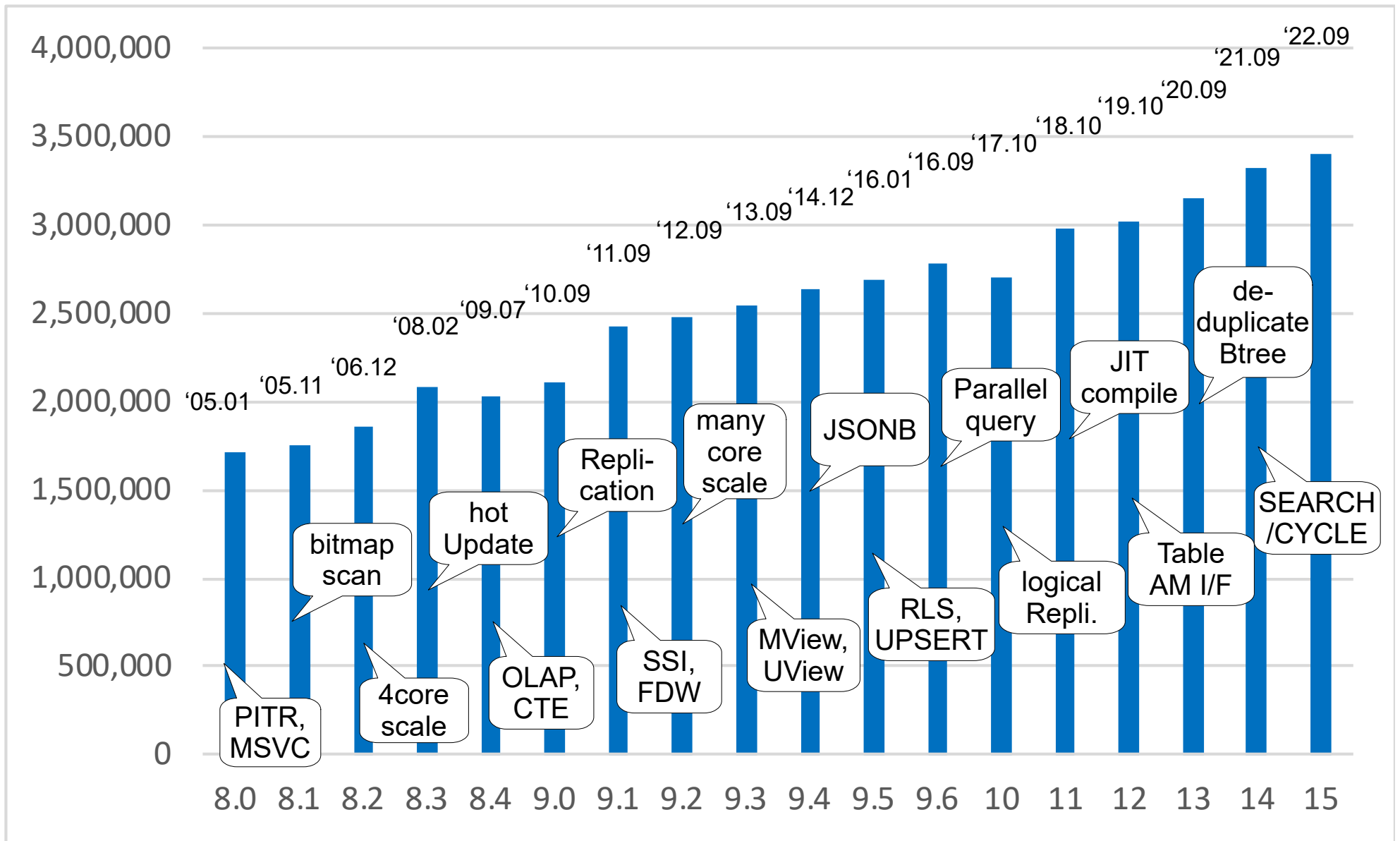
PostgreSQL とは

- 多機能、高性能、かつオープンソースのリレーショナルデータベース管理システム
 - INGRES('70),POSTGRES('80) 由来の歴史
 - BSD タイプのライセンス
 - 特定オーナー企業が無い



PostgreSQL リリース

コード行数



最新のマイナーバージョン

- 2024/5/9 に以下のバージョンをリリース
 - 16.3、15.7、14.12、13.15、12.19
- 12 のサポートは 2024/11/14 まで
- PostgreSQL のサポート期間はイニシャルリリース日から 5 年間です
- 2024/5/23 に PostgreSQL 17 Beta 1 リリース

15 までの機能拡張

- ロジカルレプリケーションとその強化 (10 - 15)
- Table Access Method 拡張用インタフェース (12)
- Btree インデックスの重複除去、肥大化防止 (13 - 15)
- パラレルクエリ対応と強化 (9.6 - 15)
- テーブルパーティショニング対応と強化 (10 - 15)
- 拡張統計情報 (12 - 15)
- 進捗モニタリング `pg_stat_progress_*` (10 - 15)

バージョン 15

- 2022年9月リリース
- 性能向上

- 全体的には劇的な性能アップなし
- ピンポイントで改善

- 外部ソートのアルゴリズム変更
- 先読みによるリカバリ性能改善
- パラレルクエリ SELECT DISTINCT も有効
- ウィンドウ関数性能改善 row_number()、rank()、count()
- NOT IN 性能改善
- ¥copy FROM 性能改善
- 実行時統計情報の共有メモリ格納

バージョン 15

- 2022 年 9 月リリース
- SQL 新機能
 - MERGE 文をサポート
- DDL 新機能
 - ロジカルレプリケーション機能追加
 - 行、列単位での絞り込み構文
- 運用管理
 - 初期状態での public スキーマへの権限変更
 - ICU をグローバルロケールプロバイダとして使用可能

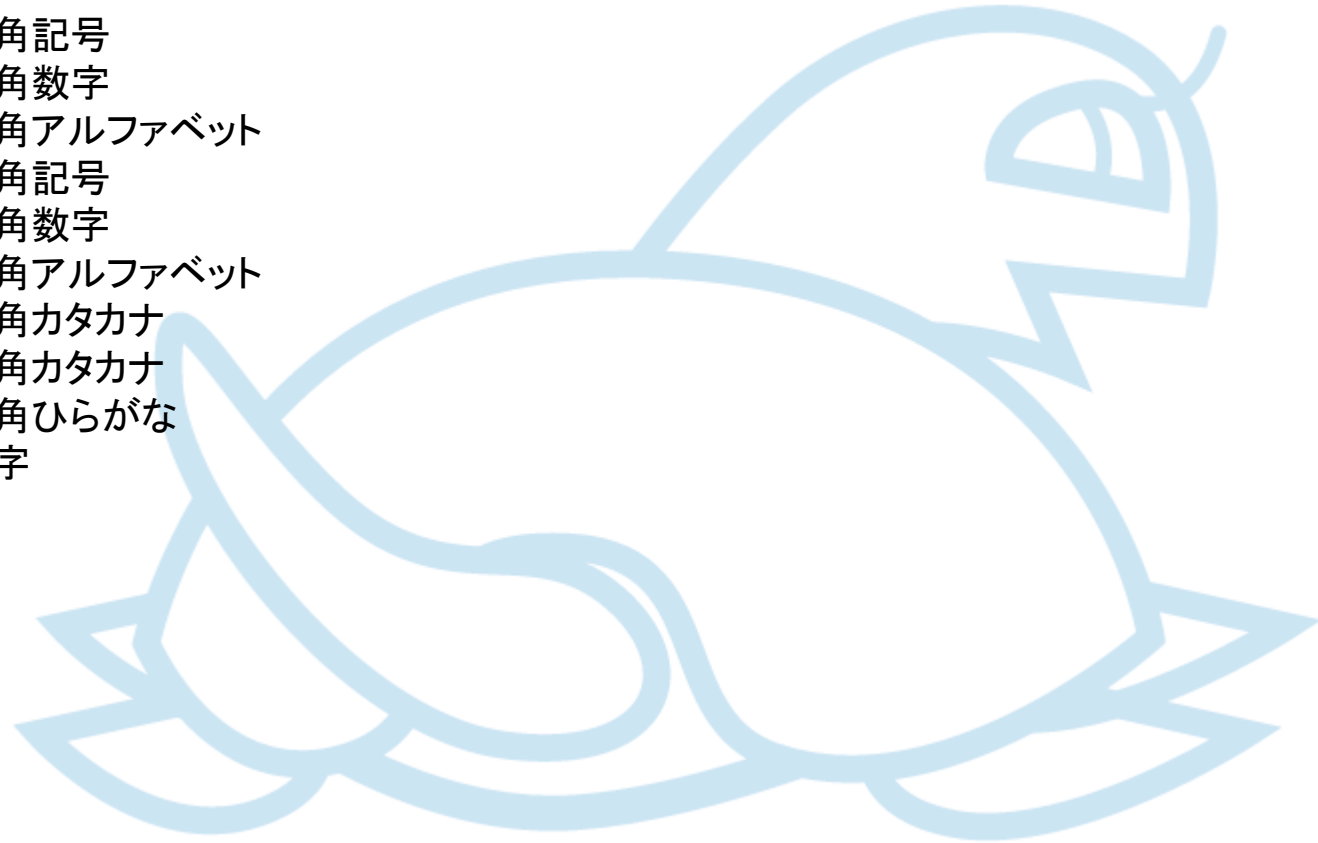
- 意欲的な機能追加
- 実用観点での拡張
- 必要なら仕様変更

PostgreSQL 16 から icu がデフォルトに

ロケールプロバイダによる振る舞い

- 以下のような文字種のソート順が変化するので注意

- 半角記号
- 半角数字
- 半角アルファベット
- 全角記号
- 全角数字
- 全角アルファベット
- 半角カタカナ
- 全角カタカナ
- 全角ひらがな
- 漢字



バージョン 16

- 性能向上
 - WAL レコードの重複排除でプラン凍結
 - 更新処理性能向上
 - ORDER BY 、 DISTINCT 集計のパフォーマンス
 - GROUP BY 、 DISTINCT 最適化
 - 繰り返しカラムのグループ化
 - パーティション化テーブルの一括挿入・更新の効率化
 - 文字列から整数への変換パフォーマンス向上
 - ハッシュインデックス構築の高速化

バージョン 16

- SQL 新機能
 - 整数リテラル：10 進数以外、3 桁区切り
 - '0x30B1'、'-0o0755'、'+0b0011'、'1_500_000'
 - DATE TIMESTAMP 型での +INFINITY が使用可能
 - FROM 句のサブクエリのエイリアスが不要に
 - ANY_VALUE 集約をサポート
 - IS_JSON 述語を追加
 - COPY での FDW への一括挿入をサポート
 - TRUNCATE のトリガーが FDW で許可

バージョン 16

- モニタリング
 - pg_verifybackup で -P オプション追加 (進捗報告)
 - モニタビュー：
 - pg_stat_io (新規追加)
 - pg_stat_all_tables (last_seq_scan 追加)
- 運用機能
 - VACUUM に以下のオプションを追加
 - SKIP_DATABASE_STATS and ONLY_DATABASE_STATS
 - PROCESS_MAIN
 - BUFFER_USAGE_LIMIT

pg_stat_io

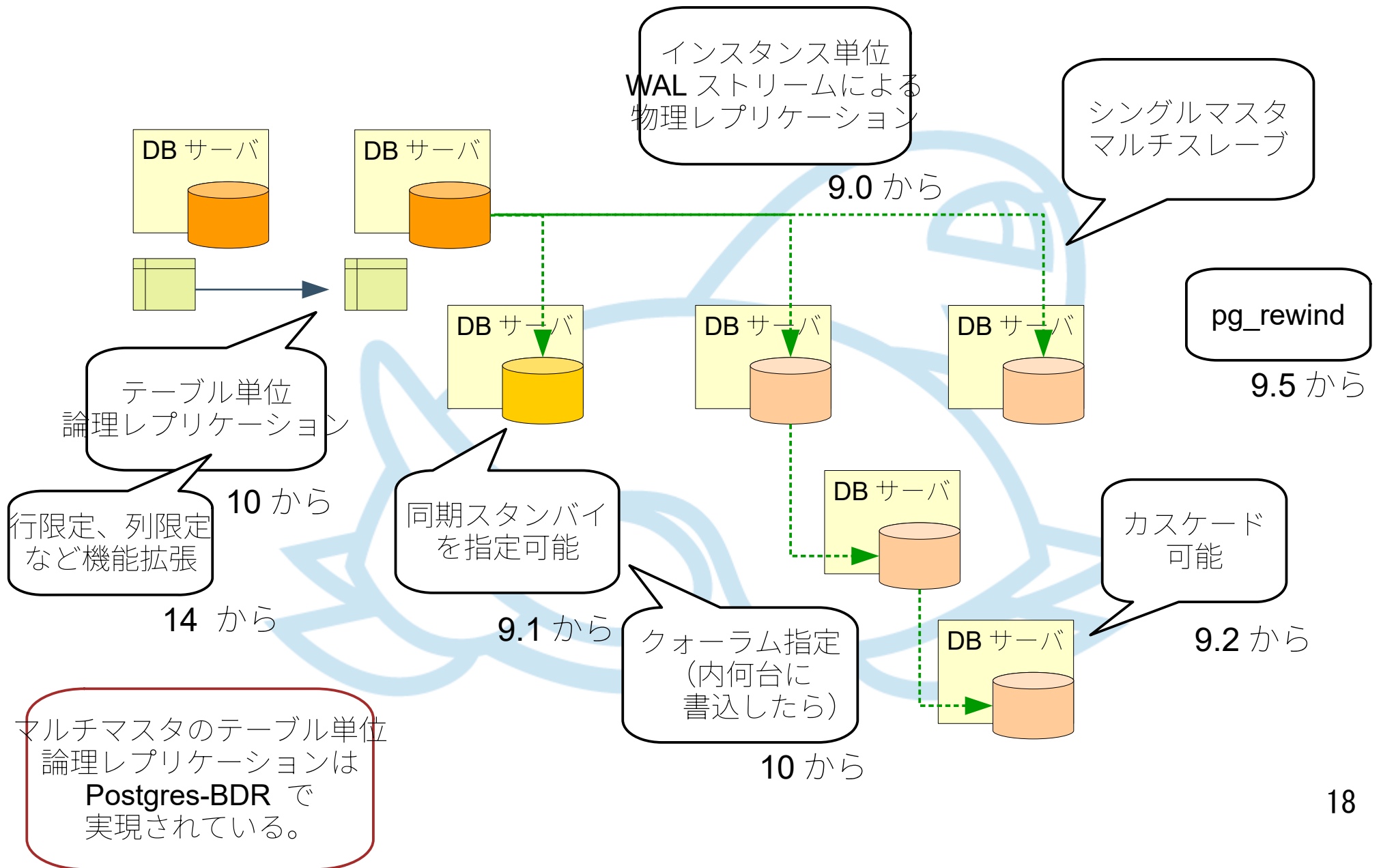
backend_type	io_object	io_context	reads	writes	extends	op_bytes	evictions	reuses	fsyncs	stats_reset
autovacuum launcher	relation	normal	19	5		8192	13		0	2023-02-13 11:50:27.583875-08
autovacuum worker	relation	normal	15972	2494	2894	8192	17430		0	2023-02-13 11:50:27.583875-08
autovacuum worker	relation	vacuum	5754853	3006563	0	8192	2056	5752594		2023-02-13 11:50:27.583875-08
client backend	relation	bulkread	25832582	626900		8192	753962	25074439		2023-02-13 11:50:27.583875-08
client backend	relation	bulkwrite	4654	2858085	3259572	8192	998220	2209070		2023-02-13 11:50:27.583875-08
client backend	relation	normal	960291	376524	159497	8192	1103707		0	2023-02-13 11:50:27.583875-08
client backend	relation	vacuum	128710	0	0	8192	1221	127489		2023-02-13 11:50:27.583875-08
background worker	relation	bulkread	39059938	590896		8192	802939	38253662		2023-02-13 11:50:27.583875-08
background worker	relation	normal	257533	118972	0	8192	256437		0	2023-02-13 11:50:27.583875-08
background writer	relation	normal		243142		8192			0	2023-02-13 11:50:27.583875-08
checkpointer	relation	normal		390141		8192			18812	2023-02-13 11:50:27.583875-08
standalone backend	relation	bulkwrite	0	0	8	8192	0	0		2023-02-13 11:50:27.583875-08
standalone backend	relation	normal	689	983	470	8192	0		0	2023-02-13 11:50:27.583875-08
standalone backend	relation	vacuum	10	0	0	8192	0	0		2023-02-13 11:50:27.583875-08

(14 rows)

バージョン 16

- conf ファイル
 - pg_hba.conf / pg_ident.conf で include をサポート
 - pg_ident.conf で pg_hba.conf とおなじような、リテラルではないユーザ名のサポート
 - pg_hba.conf で正規表現をサポート
- クライアント機能
 - psql: マジック変数追加
 - SHELL_EXIT_CODE、SHELL_ERROR
 - libpq: 接続パラメータ追加
 - sslcertmode、require_auth、load_balance_hosts

PostgreSQL のレプリケーション



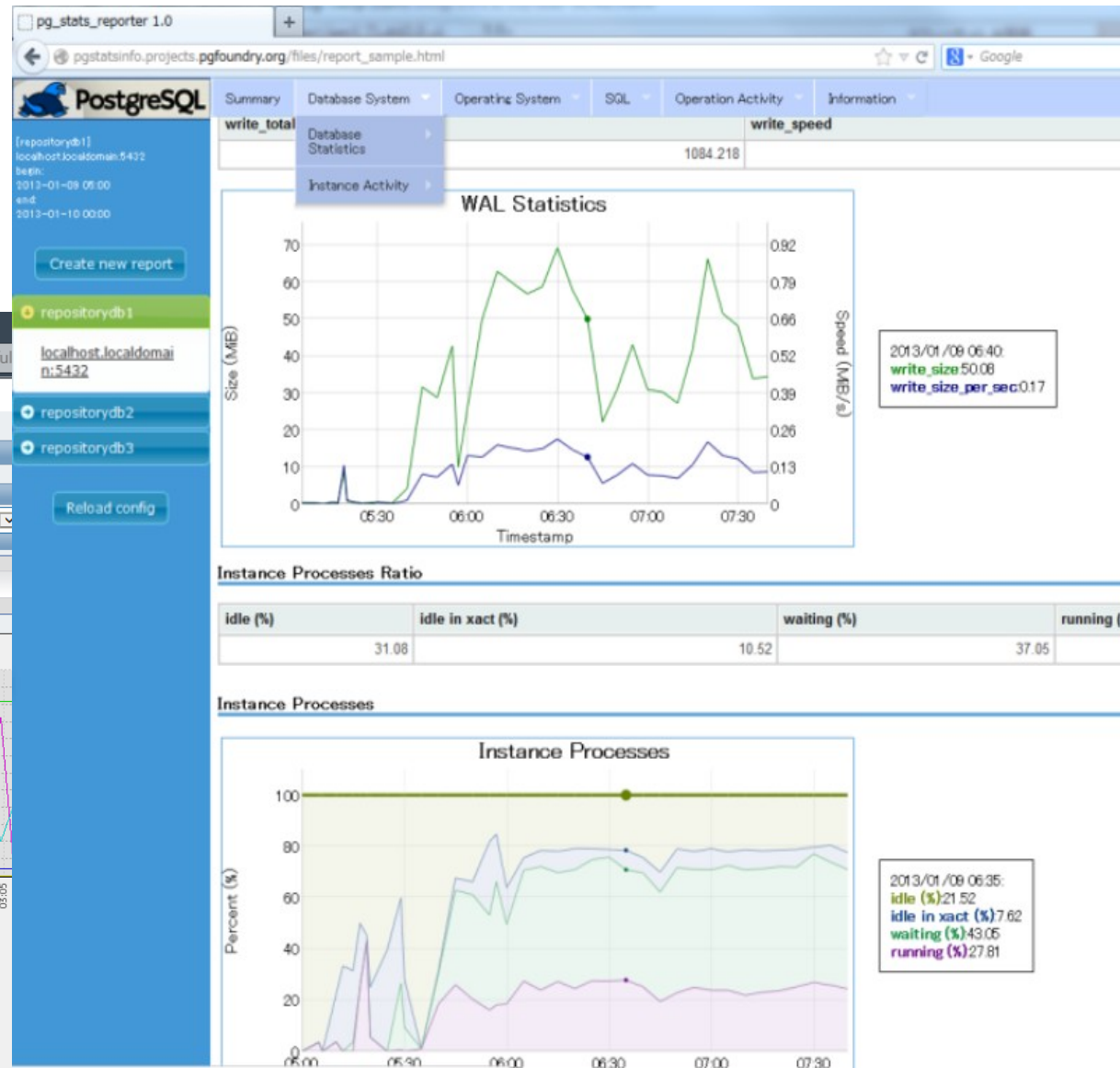
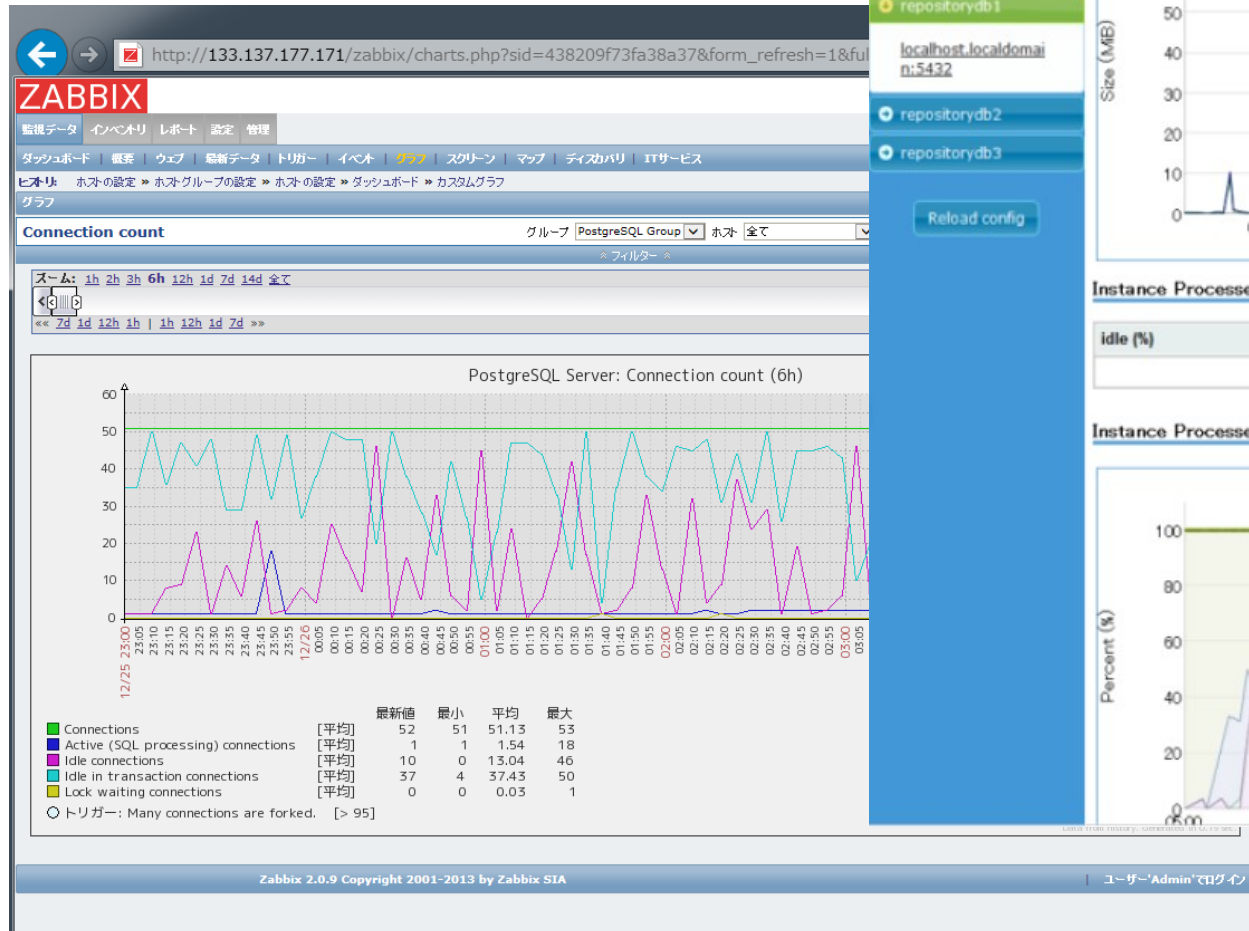
PostgreSQL クラスタ構成

- HA クラスタ
 - Pacemaker 等の各種 HA クラスタソフトで対応
 - k8s と k8s オペレータ
- MPP クラスタ (シャーディング - データ分割格納)
 - Citus (Azure Hyperscale)
 - Postgres-XL
 - GreenPlum
- ロードバランサ
 - Pgpool-II (制限あるが参照用に L3 汎用ロードバランサも有)

PostgreSQL 本体機能でも
外部テーブル+パーティショニングで
相当程度シャーディングが可能

PostgreSQL の運用監視

- pg_statsinfo
- pg_monz (Zabbix)
- pg_badger
- DataDog、Makerel



https://www.postgresql.jp/sites/default/files/2017-01/B1_PGCON_JP_kondo_nttoss.pdf より

PostgreSQL のクライアントツール

The screenshot shows the pgAdmin 4 web interface. The left sidebar displays a tree view of the database structure, including 'pem' and its tables. The main area shows a query editor with the query 'SELECT * FROM pem.agent_heartbeat' and its execution history. The 'Data Output' tab is active, displaying a table with 5 rows of data.

agent_id	last_heartbeat
1	2019-01-07 13:23:41.740671+00
2	2019-01-07 13:23:42.991127+00
3	2019-01-07 13:23:44.500388+00
4	2019-01-07 13:23:45.803882+00
5	2019-01-07 13:23:48.505901+00

<https://www.pgadmin.org/screenshots> より

- pgAdmin 4
- 各種商用製品 PostgreSQL 対応
 - Navicat for PostgreSQL
 - SI Object Browser for Postgres
- A5:SQL Mk-2

PostgreSQL でやや困難なケース

- データ投入性能の限界
 - IOT 方面 / 投入量要件と構成によっては専用製品に
 - WAL に直列的に書く設計であるため
 - OLTP 性能の限界
 - 大メモリや多 CPU コアを活かしきれない場合
 - 遅いストレージ格納を前提とした基本設計に起因
 - スケールアウトが難しい
 - HA クラスタにおける高度要件
 - 障害復旧時間の最小化要件
 - 10 秒以内の切替など
- 改善は続いている。
インスタンス分割で対応。
- 改善は続いている。
現状で不足なら負荷を
他に逃がす対応で。
- トレードオフを許容すれば
実現不可ではない

PostgreSQL クラウド / コンテナ

- クラウドサービス
 - Azure Database / Azure Hyperscale
 - AWS RDS / Aurora
 - GCP Cloud SQL / AlloyDB
- K8s オペレータ
 - KubeDB
 - CrunchyData/postgres-operator
 - Zalando/postgres-operator
 - Cloud Native PostgreSQL (EDB)

PostgreSQL 互換 / PostgreSQL ベース

- PostgreSQL 互換 DB
 - GCP AlloyDB, Spanner
 - AWS Aurora, Redshift
 - Yugabyte
 - CockroachDB
 - 劔
- PostgreSQL ベース DB 製品
 - EnterpriseDB (EDB)
 - Enterprise Postgres (Fujitsu)
 - PowerGres Plus (SRAOSS)
 - GreenPlum

コミュニティと商用サポート

- 開発コミュニティ (Mailing List, Git, Slack)
(PostgreSQL Global Development Group)
- 国内のコミュニティ活動
 - 日本 PostgreSQL ユーザ会 (JPUG)
 - PostgreSQL エンタープライズ・コンソーシアム
(PGECons)
- 国内の商用サポート
 - 提供会社いくつもあり、取り扱い会社多数

ユーザ動向

• JPUG による利用調査 2022 年版



あなたがここ数年で業務で関わったデータベースソフトウェアを選択してください。あてはまるものを全て選択してください。 ※複数回答あり(回答数:2,071/1,000名)

第1位	Oracle	31.4%
第2位	SQL Server	24.3%
第3位	MySQL	20.3%
第4位	PostgreSQL	19.4%
第5位	DB2	10.4%

【その他のDBの集計結果】

[従来型]

• Access	: 10.0%
• MariaDB	: 3.9%
• <u>HiRDB</u>	: 3.4%
• MongoDB	: 1.8%
• SAP Sybase	: 1.5%

[クラウド系]

<Amazon>

• Aurora MySQL	: 4.7%
• Aurora PostgreSQL	: 3.5%
• DynamoDB	: 2.1%
• <u>CosmosDB</u>	: 1.6%

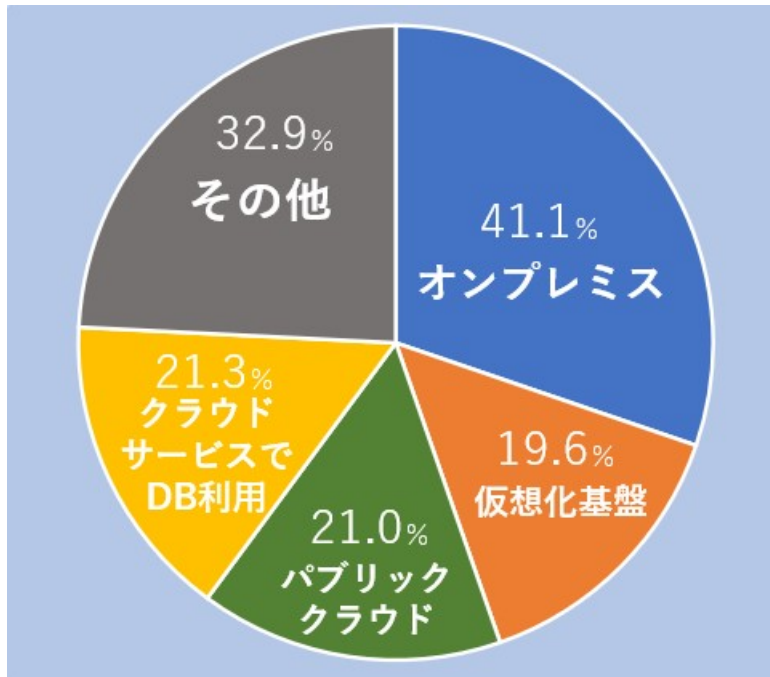
<Google>

• Spanner	: 1.1%
-----------	--------

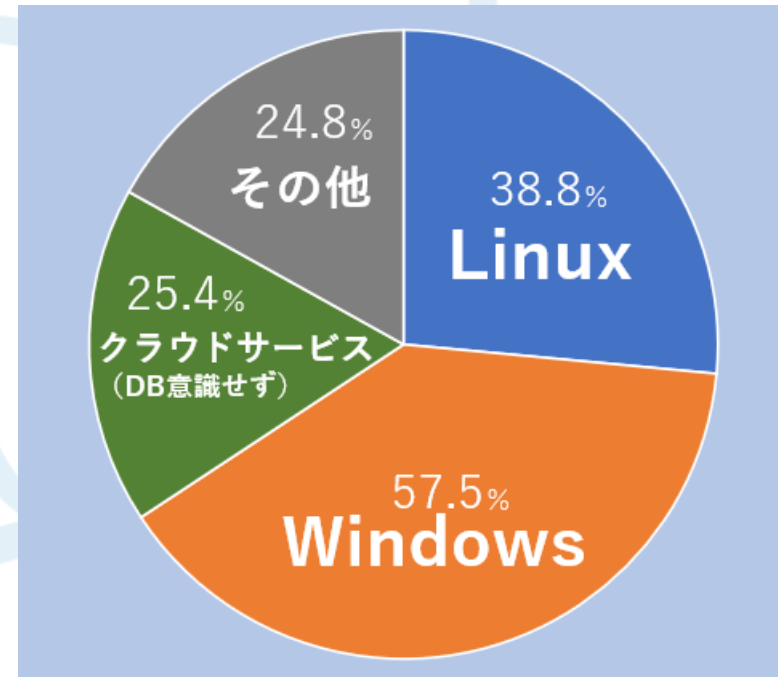
ユーザ動向

• JPUG による利用調査 2022 年版

現在お使いの、あるいは、導入しようとしているデータベースシステムが稼動する環境は？
(複数回答)



現業務でデータベースソフトウェアを使う際の OS を教えてください (複数回答)



ユーザ動向



今後、使用するのを検討していDBは何ですか？

※複数回答(回答数:757/1,000名)

第1位	SQL Server	7.0%
第2位	Oracle	6.8%
第3位	PostgreSQL	6.1%
第4位	Amazon Aurora MySQL	5.5%
第5位	MySQL	5.3%

※第6位(次点):Amazon Aurora PostgreSQL(4.1%)

JPUG の活動

- イベント
 - PostgreSQL アンカンファレンス - オンラインで毎月開催
 - PostgreSQL カンファレンス - オンサイト開催を続けました
 - PostgreSQL 勉強会 - 各地支部で年 1 回～数回
 - 合宿
 - OSC など各種イベントに出展 (通年)
- 文書翻訳 <https://github.com/pgsql-jp/jpug-doc>
- ML 運営 pgsql-jp@postgresql.jp, jpug-users@postgresql.jp
 - 最近では Slack [postgresql-jp](#) が中心
- Web サイト (www.postgresql.jp 、 lets.postgresql.jp)