PoWA 3

Optimisations avancées de PostgreSQL

Ronan Dunklau - Julien Rouhaud

Dalibo - www.dalibo.org

24 septembre 2015 - PostgreSQL Session #7

License

DALIBO
L'expertise PostgreSQL

- Create Commons BY-NC-SA
- Vous êtes libre
  - de partager
  - de modifier
- Sous les conditions suivantes
  - Attribution
  - Non commercial
  - Partage dans les mêmes conditions

Auteurs

DALIBO
L'expertise PostgreSQL

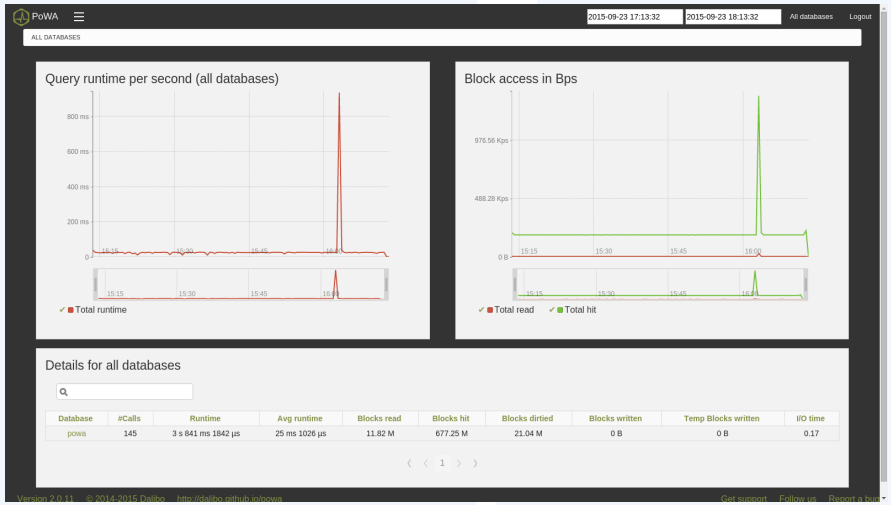
- Ronan Dunklau
  - DBA @ Dalibo
  - Open-Source : Multicorn...
  - Contributeur PostgreSQL (FDWs...)
- Julien Rouhaud
  - DBA @ Dalibo
  - Open-Source : HypoPG, OPM...
  - Contributeur PostgreSQL, PgSQL FR...
- Mais aussi
  - Marc Cousin
  - Thomas Reiss

PoWA ?

Présentation

DALIBO
L'expertise PostgreSQL

POWA



- Outil d’analyse de charge
- Et d’optimisation
- En temps réel

Pile applicative

Présentation

DALIBO
L'expertise PostgreSQL

- pg_stat_statements
- pg_stat_kcache
- pg_qualstats
- powa-archivist
- powa-web

pg_stat_statements

Présentation

DALIBO
L'expertise PostgreSQL

- Contrib officielle PostgreSQL
- Normalisation de requêtes
- Cumul de compteurs (buffers, temps d'exécution...), par
  - utilisateur
  - base
  - requête

pg_stat_statements

Concrètement

DALIBO
L'expertise PostgreSQL

- Nombre d'exécutions d'une requête
- Temps moyen d'exécution d'une requête
- Création de fichiers temporaires
- Accès dans ou hors du cache **de PostgreSQL**

pg_stat_kcache

Présentation

DALIBO
L'expertise PostgreSQL

- Capture de métriques systèmes par requête normalisée
  - Accès physique aux disques
  - Utilisation CPU

pg_stat_kcache

Concrètement

DALIBO
L'expertise PostgreSQL

- "Vrai" hit-ratio (cache PostgreSQL / cache système)
- Identification des requêtes coûteuses en temps processeur

pg_qualstats

Présentation

DALIBO
L'expertise PostgreSQL

- Analyse des prédicats
  - clause WHERE
  - clause JOIN
- Stockage de nombreuses statistiques
  - Sélectivité
  - Échantillon de constantes (+/- filtrantes, fréquentes)
  - Nombre d'exécutions
  - Type de parcours (séquentiel ou index)

pg_qualstats

Concrètement

DALIBO
L'expertise PostgreSQL

- Trouver des index manquants
- Statistiques avancées sur les prédicats
- ...

powa-archivist

Présentation

DALIBO
L'expertise PostgreSQL

- Historiser toutes ces sources de données
- Configurable (rétention, fréquence...)
- Extensible à d'autres sources de données

powa-archivist

Concrètement

DALIBO
L'expertise PostgreSQL

- Quels sont les goulets d’étranglement
- Pour quelle raison
- Y remédier
- En temps réel

Compatibilité

DALIBO
L'expertise PostgreSQL

- PostgreSQL **9.4** et plus
- PoWA 1 compatible 9.3, mais beaucoup plus limité

powa-web

Présentation

DALIBO
L'expertise PostgreSQL

- Interface graphique web
- Contrôle une ou plusieurs instances PoWA
- Drill-down analysis

powa-web

Exemple d’utilisation

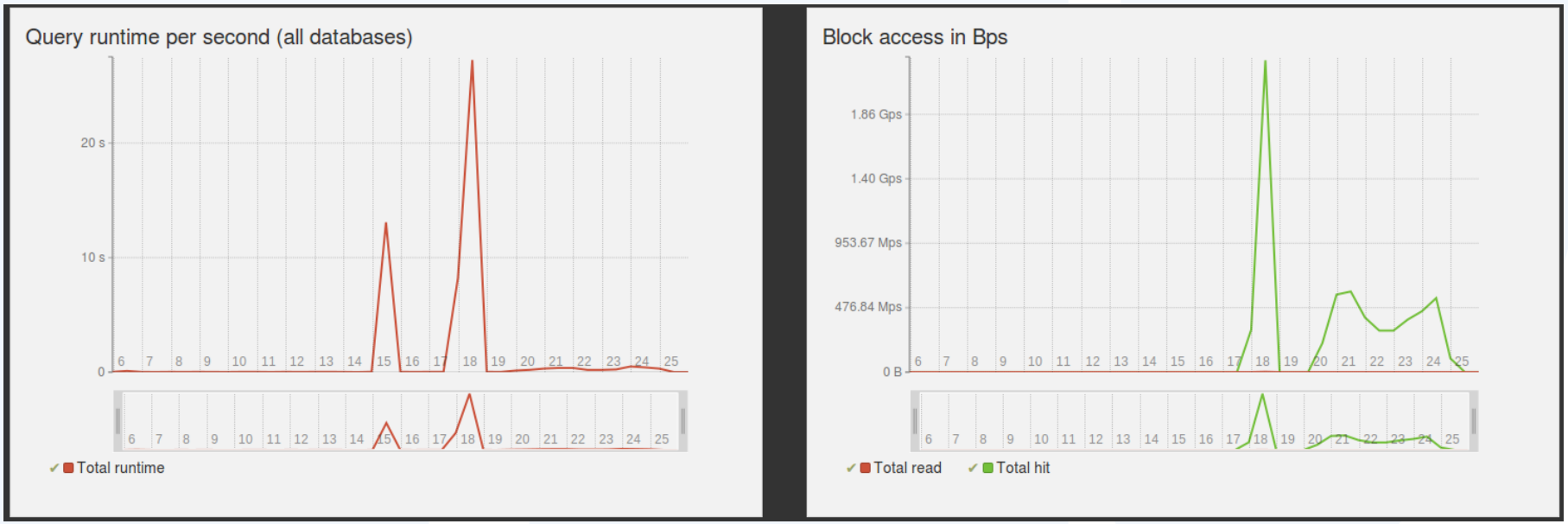
DALIBO
L'expertise PostgreSQL

- Mauvaises performances sur certaines parties d’une application
- Choisir la période d’analyse (dernière heure par défaut)
- Trouver la base concernée

powa-web

Vue d’un cluster 1

DALIBO
L'expertise PostgreSQL



powa-web

Vue d’un cluster 2

DALIBO
L'expertise PostgreSQL

Details for all databases

Database	#Calls	Runtime	Avg runtime	Blocks read	Blocks hit	Blocks dirtied
obvious	3114	11 min 19 s 718 ms 1719 µs	217 ms 1218 µs	79.20 M	154.72 G	79.13 M
powa	1407	8 s 721 ms 1722 µs	5 ms 1006 µs	9.10 M	868.98 M	17.43 M
rjuju	10	26 s 738 ms 1739 µs	2 s 672 ms 1673 µs	0 B	560.00 K	40.00 K
tpc	1368	11 min 32 s 520 ms 1521 µs	505 ms 1506 µs	1.15 M	36.02 G	0 B

1

powa-web

Vue d’une base

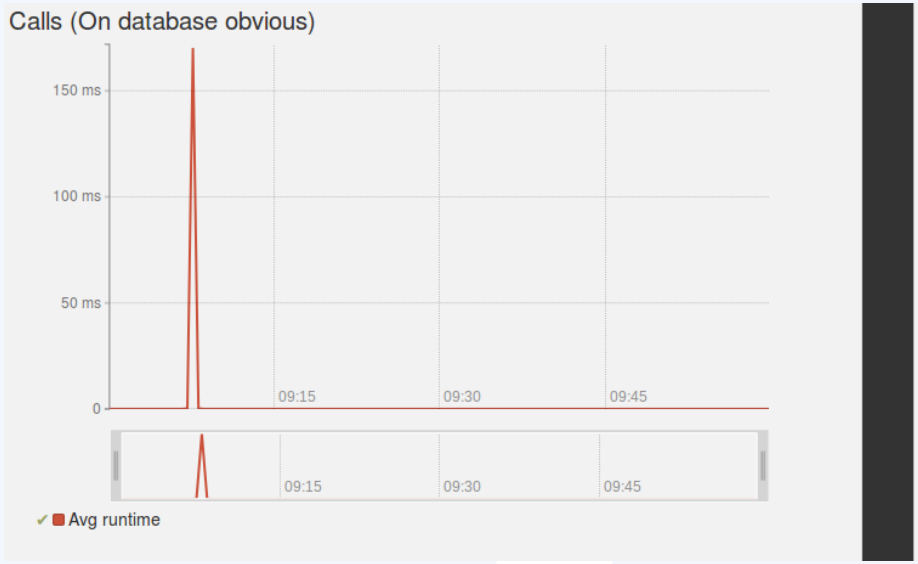
DALIBO
L'expertise PostgreSQL

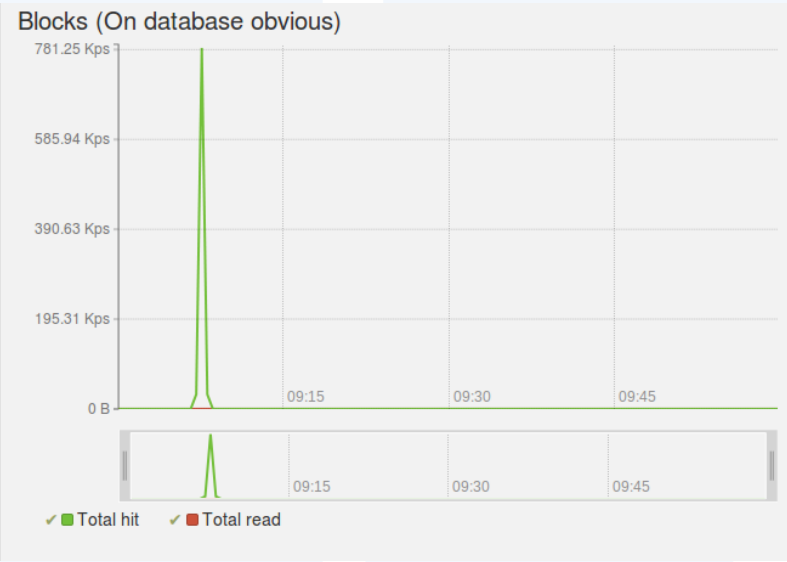
- La base est identifiée
- Trouver les requêtes problématiques

powa-web

Vue d’une base 1

DALIBO
L'expertise PostgreSQL





powa-web

Vue d’une base 2

DALIBO
L'expertise PostgreSQL

Details for all queries

Query	Execution			I/O Time		Blocks				Temp blocks	
	#	Time	Avg time	Read	Write	Read	Hit	Dirtied	Written	Read	Written
SELECT com.id, sum(c_l.price) as total_price FROM command com JOIN com	250	1 min 32 s 757 ms 1758 µs	370 ms 1371 µs	0	0	0 B	13.56 G	0 B	0 B	0 B	0 B
SELECT id,dt FROM command WHERE state = ?;	250	7 s 679 ms 1680 µs	29 ms 1030 µs	0	0	0 B	1.40 G	0 B	0 B	0 B	0 B
select current_schema()	1	59 µs	59 µs	0	0	0 B	16.00 K	0 B	0 B	0 B	0 B
SELECT t.oid, typarray FROM pg_type t JOIN pg_namespace ns ON typnames	1	27 µs	27 µs	0	0	0 B	16.00 K	0 B	0 B	0 B	0 B
select version()	1	18 µs	18 µs	0	0	0 B	0 B	0 B	0 B	0 B	0 B
SELECT extversion FROM pg_extension WHERE extname = ? LIMIT ?	1	17 µs	17 µs	0	0	0 B	8.00 K	0 B	0 B	0 B	0 B
show transaction isolation level	1	13 µs	13 µs	0	0	0 B	0 B	0 B	0 B	0 B	0 B
SELECT CAST(? AS VARCHAR(60)) AS anon_1	2	22 µs	11 µs	0	0	0 B	0 B	0 B	0 B	0 B	0 B
SELECT ? AS some_label	1	7 µs	7 µs	0	0	0 B	0 B	0 B	0 B	0 B	0 B
show standard_conforming_strings	1	5 µs	5 µs	0	0	0 B	0 B	0 B	0 B	0 B	0 B
ROLLBACK	4	5 µs	1 µs	0	0	0 B	0 B	0 B	0 B	0 B	0 B

1

powa-web

Vue à la requête

DALIBO
L'expertise PostgreSQL

- 2 requêtes problématiques trouvées
- inspection de chacune d’elles

powa-web

Première requête - SQL

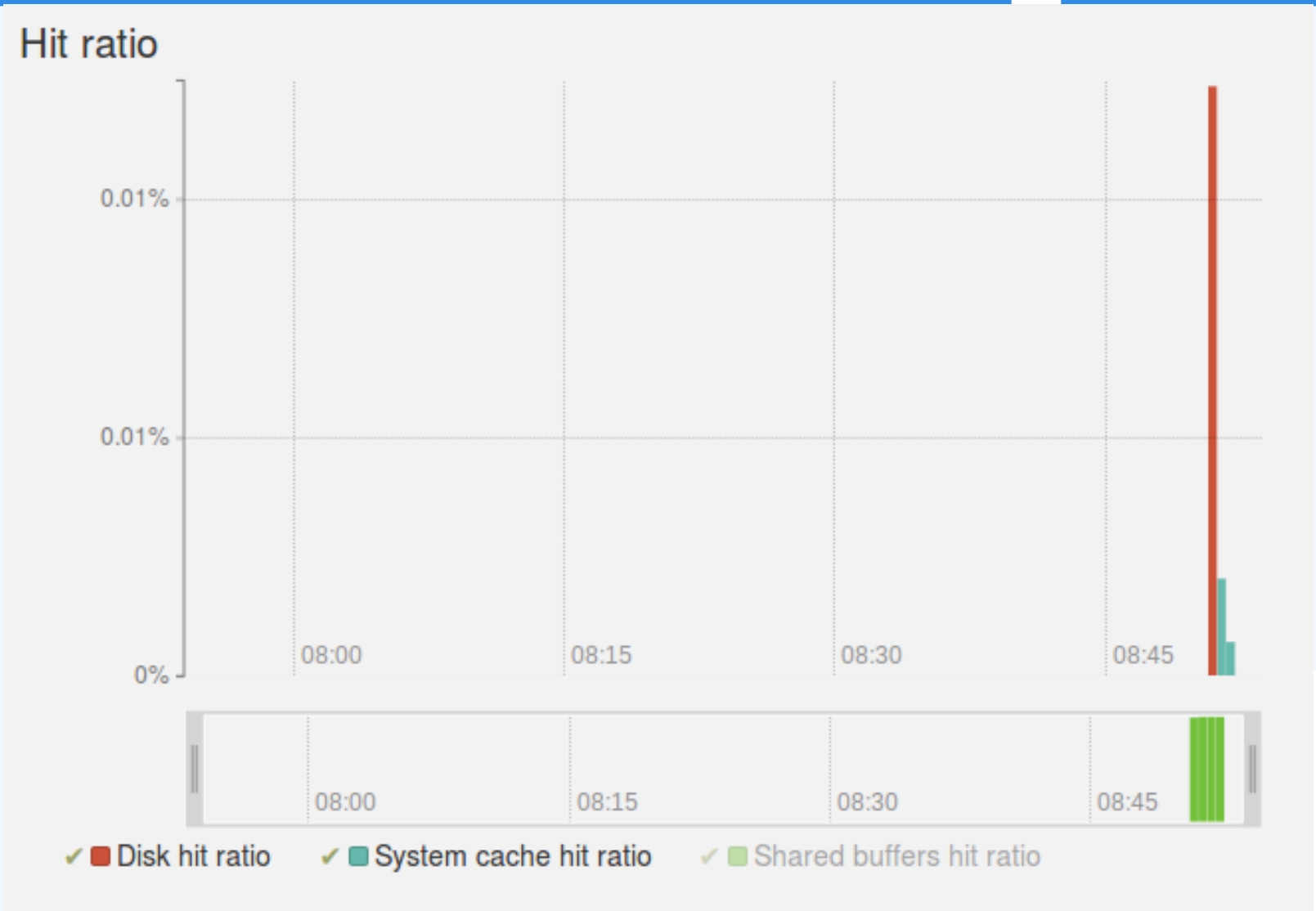
DALIBO
L'expertise PostgreSQL

```
SELECT
  com.id,
  sum(c_l.pric) AS total_price
FROM
  command com
  JOIN command_line c_l ON com.id = c_l.id_command
  JOIN client cli ON cli.id = com.id_client
WHERE
  cli.id = ?
GROUP BY
  com.id
```

powa-web

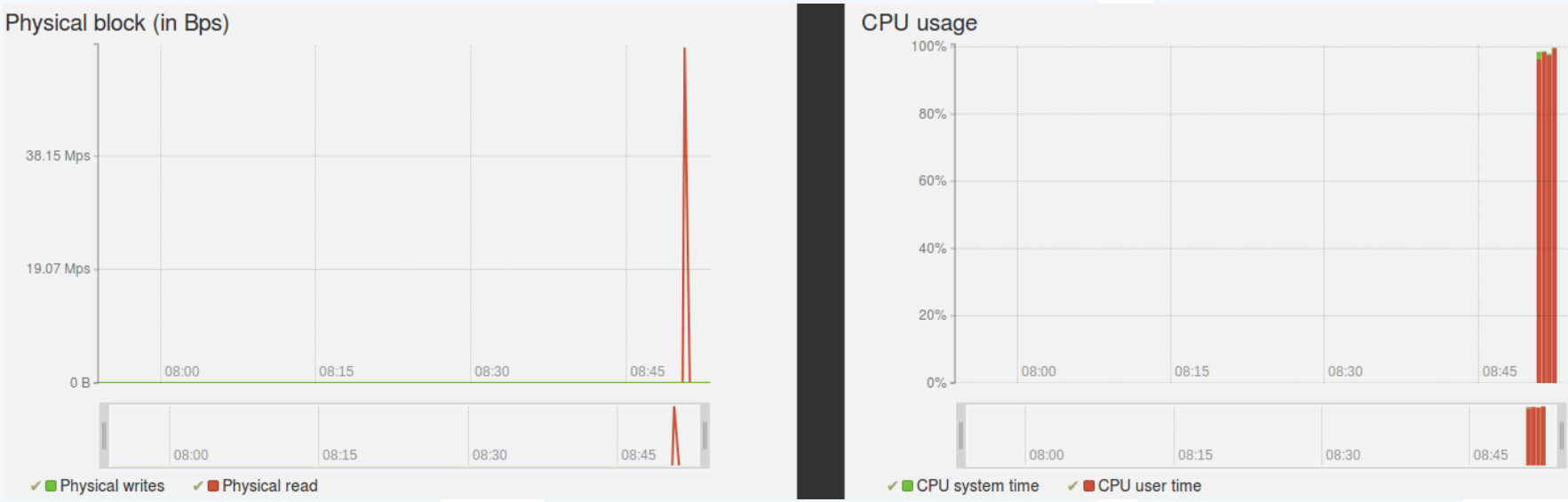
Première requête - cache

DALIBO
L'expertise PostgreSQL



powa-web

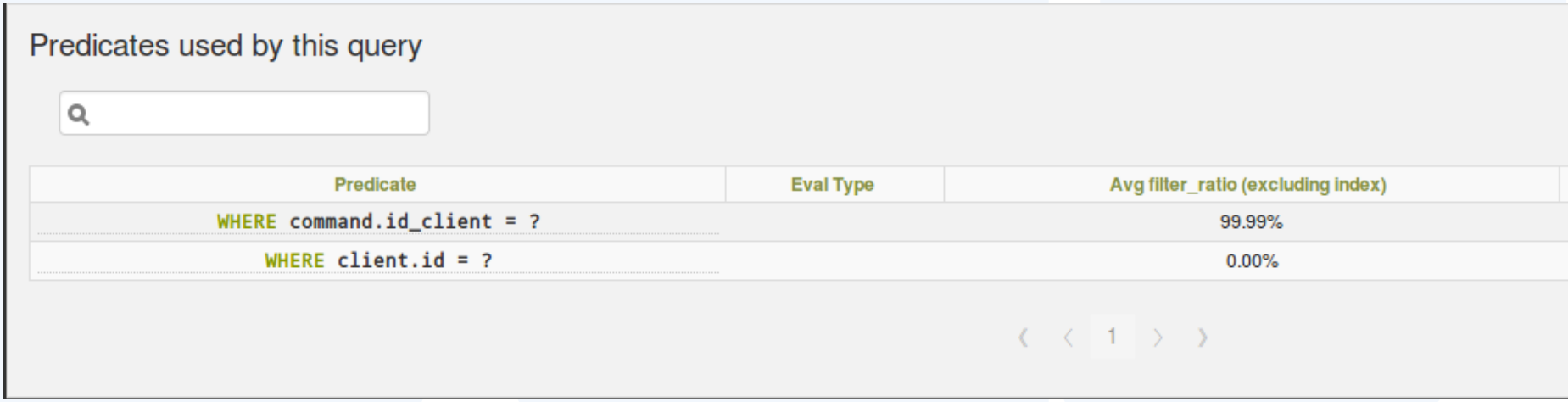
Première requête - CPU



powa-web

Première requête - prédicats

DALIBO
L'expertise PostgreSQL



powa-web

Première requête - index

DALIBO
L'expertise PostgreSQL

Index suggestion

- Possible indexes for attributes present in **WHERE command.id_client = ?**:

  With access method *btree*

  - **Attribute**

    command.id_client

    Data distribution

    approximately **9837** distinct values

  With access method *brin*

  - **Attribute**

    command.id_client

    Data distribution

    approximately **9837** distinct values

powa-web

Deuxième requête - SQL

DALIBO
L'expertise PostgreSQL

```
SELECT id, dt
FROM command
WHERE state = ?
```

powa-web

Deuxième requête - EXPLAIN

DALIBO
L'expertise PostgreSQL

Least Filtering values

Executed:

300000 times

Average filter ratio:

0.1%

Example plan:

```
SELECT id,dt FROM command WHERE state = 'shipped'::text;
```

```
Seq Scan on command  (cost=0.00..1986.00 rows=99907 width=12)
  Filter: (state = 'shipped'::text)
```

Most Executed values

Executed:

20000000 times

Average filter ratio:

99.9%

Example plan:

```
SELECT id,dt FROM command WHERE state = 'returned'::text;
```

```
Seq Scan on command  (cost=0.00..1986.00 rows=93 width=12)
  Filter: (state = 'returned'::text)
```

powa-web

En video

DALIBO
L'expertise PostgreSQL

powa-web

Nouveautés version 3

DALIBO
L'expertise PostgreSQL

- Support de l'extension HypoPG
- Suggestion globale d'index

HypoPG

Présentation

DALIBO
L'expertise PostgreSQL

- Gestion d'index hypothétique
- Création instantanée, aucune ressource consommée
- Pris en compte lors d'un EXPLAIN

HypoPG

Concrètement

DALIBO
L'expertise PostgreSQL

- Mon index sera-t-il utilisé ?
- Quelle taille fera-t-il ?
- Quel gain apportera-t-il ?

HypoPG

En action

DALIBO
L'expertise PostgreSQL

The following indexes would be used:

```
CREATE INDEX ON "public"."command"(state)
```

EXPLAIN plan **without** suggested indexes:

```
Seq Scan on command  (cost=0.00..1986.00
rows=120 width=12)
  Filter: (state = 'returned'::text)
```

EXPLAIN plan **with** suggested index

```
Index Scan using <28731>btree_command_state
on command  (cost=0.04..12.11 rows=120
width=12)
  Index Cond: (state = 'returned'::text)
```

Query cost gain factor with hypothetical index: **99.39 %**

Optimisation globale

Présentation

DALIBO
L'expertise PostgreSQL

- Trouver les index optimaux à créer
  - Satisfaire toutes les requêtes
  - Créer le moins d'index possible
  - Gestion des index multi-colonnes

Optimisation globale

Fonctionnement 1

DALIBO
L'expertise PostgreSQL

- Récupération des prédicats à optimiser
  - Prédicats filtrant en moyenne plus de X lignes
  - Prédicats filtrant en moyenne plus de X %
  - Prédicats utilisés lors d'un parcours séquentiel

Optimisation globale

Fonctionnement 2

DALIBO
L'expertise PostgreSQL

- Validation du résultat
- Si HypoPG est disponible sur la base
  - Valider l'utilisation de chaque index avec HypoPG

Optimisation globale

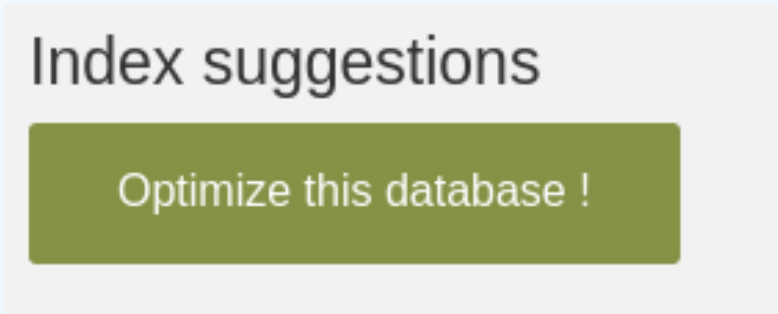
Fonctionnement 3

DALIBO
L'expertise PostgreSQL

- Estimation du gain
- Si HypoPG est disponible sur la base
  - Comparaison du coût avec et sans index hypothétique
  - Rapprochement avec le nombre d'exécutions des requêtes
  - Gain en performance estimé par requête et globalement

Optimisation globale

En action



Optimisation globale

En action

DALIBO
L'expertise PostgreSQL

Index suggestions

Optimize this database !

Done !

Index	Used by	# Queries boosted
CREATE INDEX ON public.commandes USING btree(date_commande,client_id)	WHERE commandes.client_id = ? AND commandes.date_commande >= ? AND commandes.date_commande <= ?	5
CREATE INDEX ON public.pieces_fournisseurs USING btree(cout_piece,quantite_disponible)	WHERE pieces_fournisseurs.quantite_disponible < ? AND pieces_fournisseurs.cout_piece >= ?	2
CREATE INDEX ON public.clients USING btree(solde)	WHERE clients.solde > ?	1
CREATE INDEX ON public.commandes USING btree(client_id)	WHERE commandes.client_id = ? ~~AND commandes.priorite_commande ~~ ?~~	4

Query	Index used	Gain
SELECT count(*) FROM commandes cmd JOIN lignes_commandes lc ON lc.numero_commande = cmd.numero_commande WHERE cmd.client_id = 14776;	✓	99.05%
SELECT numero_commande, etat_commande FROM commandes WHERE client_id = 14776 AND date_commande >= (2009 || '-01-01')::date;	✓	99.79%
SELECT numero_commande, etat_commande FROM commandes WHERE client_id = 14776 AND EXTRACT('year' FROM date_commande) = 2009;	✓	99.79%
SELECT COUNT(*) FROM commandes WHERE client_id = 14776 AND priorite_commande LIKE '3-%%';	✓	99.76%
SELECT count(*) FROM commandes cmd JOIN lignes_commandes lc ON lc.numero_commande = cmd.numero_commande WHERE cmd.client_id = 14776 AND date_commande BETWEEN (2009 || '-01-01')::date AND (2009 || '-12-21')::date;	✓	99.61%
SELECT COUNT(*) FROM commandes WHERE date_commande BETWEEN (2009 || '-01-01')::date AND (2009 || '-12-21')::date;	✓	42.65%
SELECT co.nom FROM clients cl JOIN contacts co ON co.contact_id = cl.contact_id WHERE cl.solde > 494;	✓	27.98%
SELECT COUNT(*) FROM commandes WHERE date_commande BETWEEN (2009 || '-01-01')::date AND (2009 || '-12-21')::date AND priorite_commande LIKE '3-%%';	✓	45.57%
SELECT numero_commande, etat_commande FROM commandes WHERE client_id = 14776 AND date_commande BETWEEN (2009 || '-01-01')::date AND (2009 || '-12-21')::date;	✓	99.83%
SELECT COUNT(*) FROM pieces_fournisseurs WHERE cout_piece >= 949	✓	48.5%
SELECT COUNT(*) FROM pieces_fournisseurs WHERE quantite_disponible < 4239 AND cout_piece >= 949;	✓	54.84%
SELECT numero_commande, etat_commande FROM commandes WHERE client_id = 14776;	✓	99.74%

Optimisation globale

En action

DALIBO
L'expertise PostgreSQL

- vidéo

Liens utiles

DALIBO
L'expertise PostgreSQL

- powa-archivist
  - Site
  - Dépôt
- powa-web
  - Dépôt
  - Démo
- pg_qualstats
  - Dépôt
  - Article
- pg_stat_kcache
  - Dépôt
  - Article
- HypoPG
  - Site
  - Dépôt
  - Article

Questions ?

DALIBO
L'expertise PostgreSQL

- contact@dalibo.com
- powa@dalibo.com
- powa.readthedocs.org