

The Need For Speed

leads to PostgreSQL

Dimitri Fontaine `dimitri@2ndQuadrant.fr`

28 Mars 2013

2ndQuadrant France PostgreSQL Major Contributor

- pgloader, prefix, skytools, debian, ...
- CREATE EXTENSION
- CREATE EVENT TRIGGER
- *Bi-Directional Réplication*
- *Partitionnement*



Performances





u13074820 [RF] © www.visualphotos.com

Quelques expériences d'optimisations

- Optimisation de requêtes
- Migration Oracle, de 1h30 à 5mins
- *prefix, GiST indexing*
- pgloader
- preprepare



Extract Month From date, THE HORROR

Certaines requêtes SQL sont assez faciles à optimiser.

```
select f1, f2, f3, d
  from t
 where  extract('month' from t.d)
        = extract('month' from now())
```



Extract Month From date, THE HORROR

Ah, tiens, un index !

```
select f1, f2, f3, d
  from t
 where      t.d >= date_trunc('month', now())
 and t.d <   date_trunc('month', now())
           + interval '1 month'
```



Performance Club

<http://fetter.org/optimization.html>

- 1 The first rule of Optimization is, you do not talk about Optimization.
- 2 The second rule of Optimization is, you **DO NOT** talk about Optimization.
- 3 If your app is running faster than the underlying transport protocol, the optimization is over.
- 4 **One factor at a time.**
- 5 No marketroids, no marketroid schedules.
- 6 Testing will go on as long as it has to.
- 7 If this is your first night at Optimization Club, you have to write a test case.



Performance Club

<http://fetter.org/optimization.html>

- 1 The first rule of Optimization is, you do not talk about Optimization.
- 2 The second rule of Optimization is, you **DO NOT** talk about Optimization.
- 3 If your app is running faster than the underlying transport protocol, the optimization is over.
- 4 **One factor at a time.**
- 5 No marketroids, no marketroid schedules.
- 6 Testing will go on as long as it has to.
- 7 If this is your first night at Optimization Club, you have to write a test case.





Premature optimization is the root of all evil

Donald Knuth, Structured Programming with go to Statements

“Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered. We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil. Yet we should not pass up our opportunities in that critical 3%.”



La performance : quand s'y intéresser ?

Premature optimization is the root of all evil

- Jamais trop tôt
- Avant qu'il ne soit trop tard
- Préparation d'une phase de croissance
- Réduction des dépenses énergétique
- Réduction de la facture d'hébergement
- Meilleur service aux utilisateurs



Diminishing Returns

Loi des rendements décroissants

- *Plus tu pédales moins vite, moins plus vite tu avances*
- Maîtriser l'effort d'optimisation
- Difficulté de savoir quand on est allé assez loin
- Prendre du recul sur ce que l'on fait



La performance dans quel but

<http://www.copyblogger.com/website-speed-matters/>

A one second delay in page-load can cause 7% loss in customer conversions.

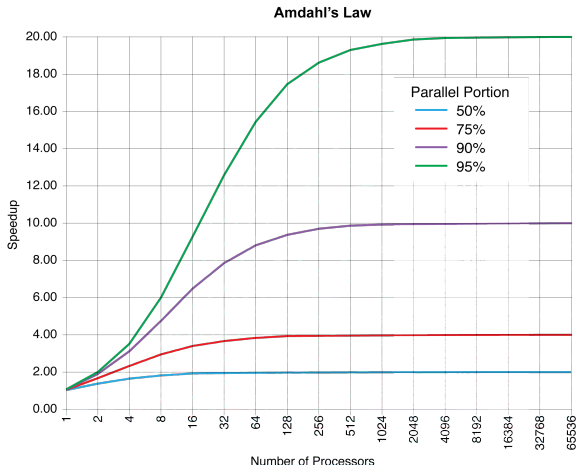
If you make \$1,000 a month from your site — that's seventy bucks a month you are losing — and \$840 a year. Can you afford to just throw away \$70 a month? \$840 a year?



La performance c'est quoi ?

Améliorer les performances ne peut se faire qu'après avoir *profilé*

- Loi d'**Amdahl**
- *profiling*
- *metrologie*



La performance c'est quoi ?

Mesurer les performances

- EXPLAIN
- EXPLAIN (VERBOSE, BUFFERS, ANALYZE)
- `\timing`



La performance c'est quoi ?

Attention aux allers retours entre client et serveur

- Round-trip
- Bande passante (*bandwidth*)



Comment améliorer les performances

Premier réflexe : **ne pas** exécuter la requête !

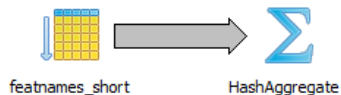
- Traitements par lots (*batch*)
- Traitements hors lignes (*asynchrone*)
- PGQ
- LISTEN et NOTIFY



Outils d'analyse des performances des requêtes 1/2

Analyser les performances

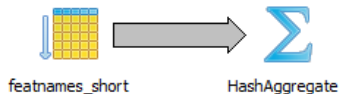
- EXPLAIN
- (ANALYZE, VERBOSE, BUFFERS)
- INSERT, DELETE, UPDATE
- (*ne pas oublier de ROLLBACK*)



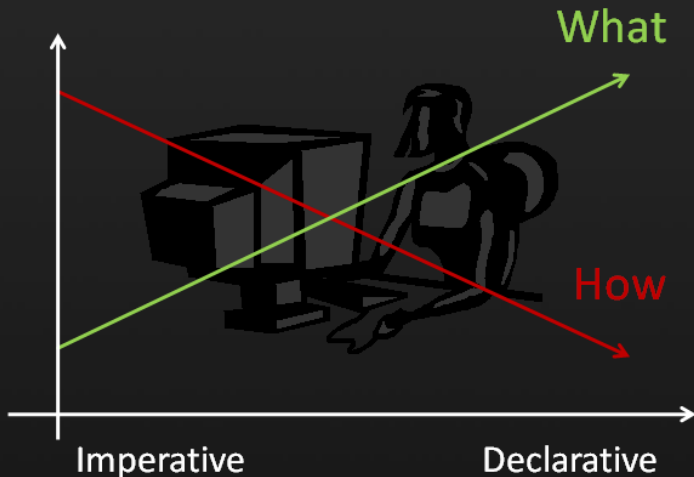
Outils d'analyse des performances des requêtes 2/2

Analyser les performances

- Un mot sur les Nested Loops
- <http://explain.depesz.com/>
- `SELECT * FROM pg_locks;`
- `pg_activity`

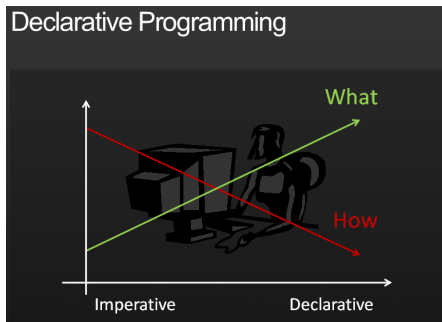


Declarative Programming



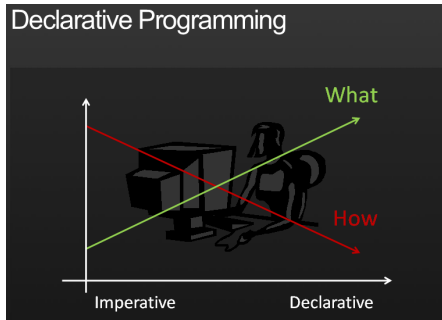
Les approches essentielles

- Identifier les requêtes
- Réécrire les requêtes
- *Traiter moins de données*
- Revoir les **Types de données**
- Revoir l'utilisation des indexes
- Utiliser des tableaux
- WITH et OFFSET



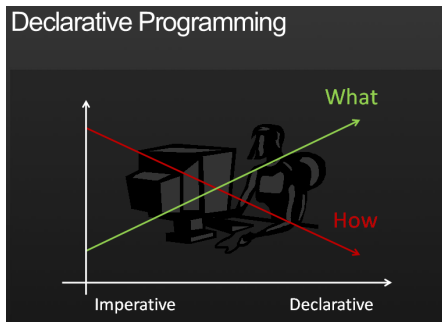
Les approches essentielles

- Identifier les requêtes
- Réécrire les requêtes
- *Traiter moins de données*
- Revoir les **Types de données**
- Revoir l'utilisation des indexes
- Utiliser des tableaux
- WITH et OFFSET



Les approches essentielles

- Identifier les requêtes
- Réécrire les requêtes
- *Traiter moins de données*
- Revoir les **Types de données**
- Revoir l'utilisation des indexes
- Utiliser des tableaux
- WITH et OFFSET



Conclusion

The **Need** For *Speed*.

