



PostgreSQL杂志-中文版

最权威的Postgres社区中文资讯

数据库性能优化： 飞越性能极限！

00期 特刊



人物专访：Bruce Momjian

案例研究：法国社会保障系统

最新版本：PG9.1 新功能预览

社区观点：PostgreSQL vs. NoSQL

技巧分享：使用pgFouine进行SQL性能分析



May 2011

00期

About PGMag

PostgreSQL Magazine is a media edited by and for the PostgreSQL Community.

Editor : Damien Clochard

Writers : Greg Smith, Frank Wiles, Hubert Lubaczewski, Jim Mlodgenski, Matt Tescher

Reviewers :

Tools : Scribus 1.4 / Gimp 2.6

Licence :

The articles contained in this magazine are released under the Creative Commons Attribution-Share Alike 3.0 Unported license. This means you can adapt, copy, distribute and transmit the articles but only under the following conditions: You must attribute the work to the original author in some way (at least a name, email or URL) and to this magazine by name ('postgresql magazine') and the URL (pgmag.org). You cannot attribute the article(s) in any way that suggests that they endorse you or your use of the work. If you alter, transform, or build upon this work, you must distribute the resulting work under the same, similar or a compatible license.

Disclaimer :

PostgreSQL Magazine is an independent media. The views and opinions in the magazine should in no way be assumed to be endorsed by the PostgreSQL Global Development Group.

This magazine is provided with absolutely no warranty whatsoever; neither the contributors nor PostgreSQL Magazine accept any responsibility or liability for loss or damage resulting from readers choosing to apply this content to theirs or others computers and equipment.

Photo Credits :

Front : © expressmonorail (flickr) ¹ / p4 : © Magnus Hagander / p5 : Vardion (wikipedia) / p6 : © loveliness.com ² / p8 : © Greg Stein / p11 : © Christine Momjian / p13 : © dirkjankraan (flickr) ² / p17 : © Gage Skidmore (flickr) ² / p18 : © BULL / p19 : © Sebastiaan Zwartz (flickr) / p23 : © Geek and Poke ³

¹ : Creative Commons BY-NC-SA

² : Creative Commons BY-SA

³ : Creative Commons BY-ND

编者按

这是一个尝试，用一本出版物尝试揭开PostgreSQL的神秘面纱。您现在正在阅读的这本杂志即将完整的为您介绍PostgreSQL——全世界最先进的开源数据库。

在2011年创设这本杂志似乎是一件不可思议的事情.....虽然PostgreSQL是迄今为止历史最悠久的开源项目，并且有着广泛的用户群和应用领域，但在它25年的发展历程中却没有任何一种媒介形式来对其进行完整详细的报道.....

这本杂志的目的之一就是让大众明白谁是“站在大象背后的人”（注：谚语，即是PostgreSQL的幕后人物）。如专访Bruce Momjian（PostgreSQL社区的创始人之一），一起探讨PostgreSQL项目的过去和未来。（见第8页）

几个星期后，伴随着最新9.1版本发布，PostgreSQL的前途也会越来越光明。这一重要的版本会包括许多令人惊叹的功能（如同步复制、非日志表、外部数据管理，等等）。而且在这个版本上，我们将继续致力于研发“Extensions”——一种非常简单的方法来对你的PG数据库来进行模块封装和插件安装。

首先，我们尝试着解答一下大家对DBA们的疑问。很多人问“DBA能够保证数据传输的更快吗？的确，性能优化是一个广泛的话题。你将会在下文中找到满意的答案，如一般性的建议（第12页），操作系统调节的技巧（第16页）和查询分析方法（第24页）。我们衷心的希望这些文章能够帮助您在以后的工作中更好的了解和使用PostgreSQL。

通过这本杂志，我们想创设一个开放的媒介来关注PostgreSQL用户群。如果您喜欢使用PostgreSQL，我们也希望您能同样喜欢这门杂志。

当然，如果您有任何意见和想法可直接给我们发送邮件，邮箱为contact@pgmag.org!欢迎您来信交流。

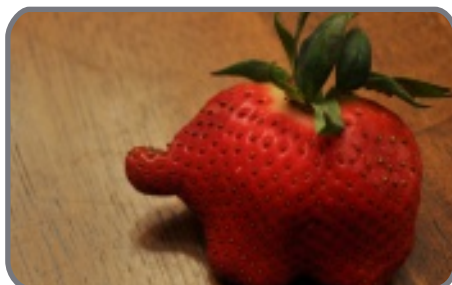
Damien Clochard

PostgreSQL中文杂志 第00期



近期会议及事件

4



最新产品发布

6

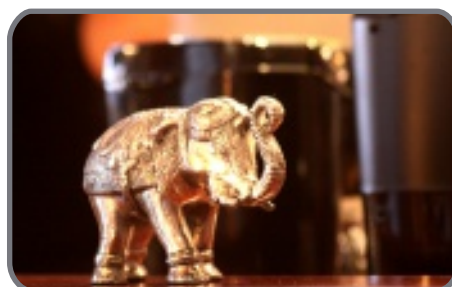
Bruce Momjian专访
为PostgreSQL社区贡献超过
15年的幕后英雄

8



如何进行性能调整

12

如何针对PostgreSQL调整
Linux系统的性能

16

法国社会保障系统
PostgreSQL应用案例

18

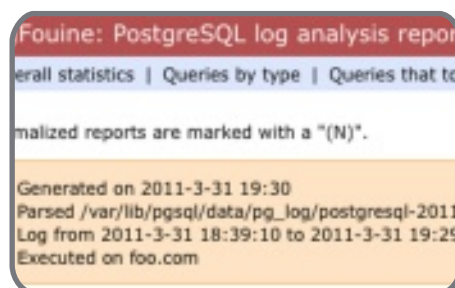
PostgreSQL 9.1
新功能预览

20



理清NoSQL在数据库领域的角色

22

使用pgFouine进行
SQL性能分析

24

How To Contribute ?

The magazine is an open project. You can contribute in various ways : writing, editing, translating, proof-reading, etc. To participate, send us a message at contact@pgmag.org

全球会议及事件



PGCon 2011年5月17-18日
加拿大渥太华

PGCon是为PostgreSQL的开发者和使用者举行的年度国际会议。在这个会议上开发者和用户能够会面、讨论、交朋友、学习到有价值的见解，而且能够交流彼此工作中使用PostgreSQL的心得体会。

<http://www.pgcon.org/2011/>

Postgres Open 2011年9月
美国芝加哥

Postgres Open是讨论关于北美地区PostgreSQL信息的顶级商务会议，这个新会议欢迎所有PostgreSQL数据库技术相关的用户、供应商和投资者来参加。

<http://Postgressopen.org>

Pg Conf Colombia
2011年8月4-5日
哥伦比亚布卡拉曼加

PG Conf Colombia 2011
于2011年8月4-5号在桑坦德工业大学
举行并于8月2-3号开设私人课程培训

<http://www.pgconf.org/>

PGBR
2011年11月3-4日
巴西圣保罗

PGBR（在巴西PGCon业内知名的会议）是美国PostgreSQL最大型的会议。在2007年，这个会议吸引了超过200名IT界的专家参与，2008和2009年，参会的专家超过300人。在2011年，会议开设3个分会，分别举行专题报告，讲座和高层次对话。

<http://pgbr.PostgreSQL.org.br/>

For more informations on local and international PostgreSQL user conferences, check out the wiki !

会议资讯 5

<http://wiki.postgresql.org/wiki/Events>

CHAR(11)
2011年7月11-12日
英国剑桥

CHAR (11) 是一个讨论集群技术、高可用性解决方案、复制技术的会议。参会者可以通过信用卡或者在Paypal平台支付345英镑的费用（税后）来进行预订。

<http://www.char11.org/>

PostgreSQL Session #2
2011年6月23日
法国巴黎

会议在巴黎举行，并向所有人免费开放。每一场会议都围绕一个特定的主题和嘉宾来举行一整天的讲座和研讨会。接下来的议题将围绕PostGIS来展开。

<http://www.postgresql-sessions.org/en/>

PostgreSQL Conference Europe
2011 2011年10月18 - 21日
荷兰阿姆斯特丹

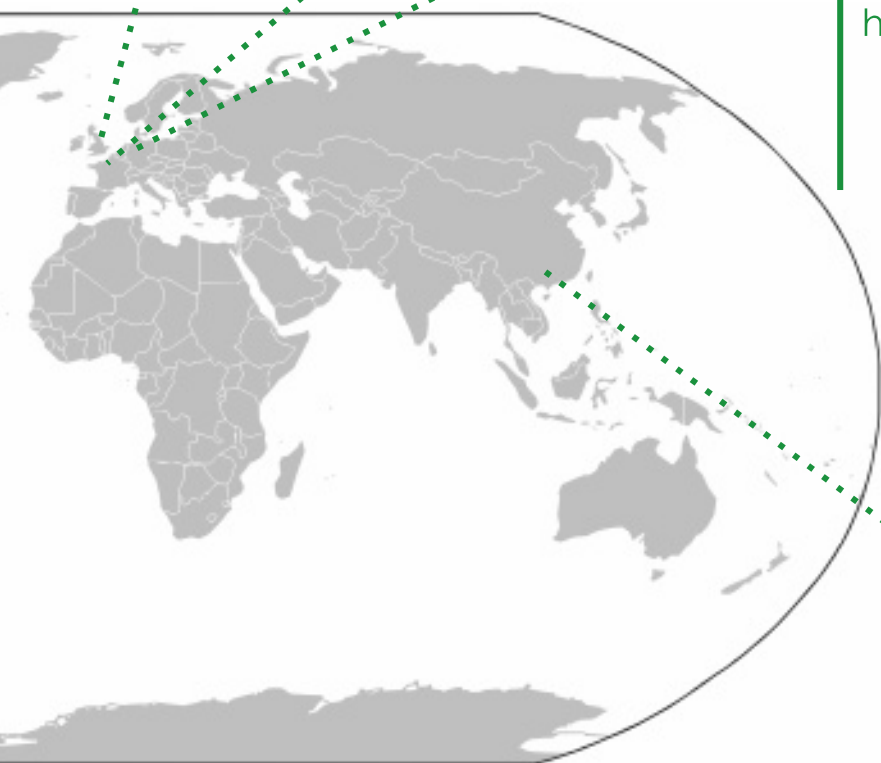
欧洲PostgreSQL会议（通常被称为PG Day.EU）阿姆斯特丹的Casa400旅店举行。会议的第一天将是实训日，接下来的三天将举行常规的会议。这个会议上能够使用英语、荷兰语、德语和法语进行交流。

<http://2011.pgconf.eu/>

China PostgreSQL Conference
2011 2011年7月16 - 17日
中国广州

中国PostgreSQL用户组织（CPUG）会议将在7月召开，为期两天。演讲者有David Fetter 和Tatsuo Ishii等人。

联系CPUG2011@gmail.com



新品发行公告



Pgwatch, 一个新的PostgreSQL监控工具

Cybertec Schönig & Schönig GmbH 在二月, 发布了pgwatch, 一个简单的监控工具, 它支持PostgreSQL9.0或更高版本。这个项目的目标是研发一个PostgreSQL的专业监测工具, 而且它能够提供开箱即用的功能。

pgwatch提供了丰富的预先设定好的图表和功能, 如: 大量配置简单可直接使用的图表, 快速系统检查工具, 自动搜集统计信息, 交互式Flash图表, 集成 SQL 工作表等。

Pgwatch 是开源的, 在遵守Creative Commons License下可以自由使用。

<http://www.cybertec.at/en/pgwatch/>

repmgr 1.1.0 已经发布

repmgr 是 2ndQuadrant 研发的, 支持Postgres 9.0的一个项目。它能够进行大量数据的复制管理。通过提供一个开箱即用的后台系统进程来使得多节点运算更为协调。repmgr把很多节点融合为一个集群, 从而更容易安装和管理。repmgr目前主要是针对UNIX系列系统, 它要求用户利用PGXS接口来进行PostgreSQL相关代码的编译以及高效节点的同步备份。

<http://projects.2ndquadrant.com/repmgr>

Dumbo 0.50, 一个新的客户终端

Dumbo是一个新的全屏控制并继承pgAdmin风格的控制台。它是由python语言结合urwid程序包编写而成, 可以在Linux平台上运行。

<https://bitbucket.org/mixmastamyk/dumbo/>

PostgreSQL Query Cache 发布

Satoshi NAGAYASU 三月份，发布了一个软件包，通过在后台利用查询缓存来改善查询性能（提高10倍—100倍）。PostgreSQL的缓存通过一个特殊的端口等待与用户的连接，它的运行机制就是委托后台程序（就像是一个代理人）和缓存进行匹配，进而产生出正确的查询结果。同时它还能够管理查询缓存的生命周期。

<http://code.google.com/p/pqc/>

PL/Proxy 2.2 已经发布

PL/Proxy是使用过程化语言的数据库分区系统。这个最新版本包含一个新的TARGET语句来指定利用不同的函数去访问远程端。

<http://pgfoundry.org/projects/plproxy/>

德语PostgreSQL文档翻译项目启动

这项工程的目标即是把PostgreSQL文档翻译为德语。目前已经召集了一小部分志愿者来翻译校对，使得翻译后的工具和语言更容易被接受和理解。其它地区仍然提供英文版本的软件。德语翻译会陆续公布在相应的网站上。

<http://doc.postgres.de>

Google Summer of Code 2011

PostgreSQL项目已经被Google Summer of Code 2011认可。学员们可以在3月28日到4月8日之间上交提案书。进一步研究工作在5月23号到8月15号期间展开。

http://wiki.postgresql.org/wiki/GSoC_2011

帮助日本和JPUG成员走出灾难的阴影

PostgreSQL 用户：

我们的项目和日本有着特殊的关系，但是现在这个国家却在饱受灾难的折磨。日本PostgreSQL用户组（JPUG）是全世界大型用户组的代表，它的成员数量占世界总用户总量的20%以上。大概1/4的代码贡献者来自日本，这些志愿者包含各个年龄阶层，其中有许多知名的人士，当然也有许多我们至今也没有听说过的。

因此，我们呼吁更多PostgreSQL的支持者去帮助日本摆脱这次灾难。众人拾材火焰高，救助越多，日本也就能更快的摆脱灾难带来的影响。海滨的日本人需要食物、医疗救助、房子和一切日本政府不能及时提供的救助。

我们推荐通过下列组织来捐赠：

International Federation of The Red Cross/Crescent:

<http://pgmag.org/0007a>

Doctors Without Borders:

<http://pgmag.org/0007b>

Josh Berkus

PostgreSQL Core Team



SHU L. J. NIK / TREE AT
SOFTEN / R. L. CONES
COLOMBIA 80 LAR

Bruce Momjian

为PostgreSQL社区贡献超过15年的幕后英雄

他是一个真正的先驱者,自从1996开始, Bruce Momjian就活跃于PostgreSQL全球开发小组中。丰富的经历成就了他在PostgreSQL大家庭中独一无二的地位,所以我们邀请他来介绍一下这个世界上最先进的开源数据库的过去、现在和将来。

这个工作很适合我,因为我喜欢做这些,而且这个社区(和我现在工作的EnterpriseDB)也都需要我。除此之外,我喜欢做这些有用的事情(即使它们很枯燥)。所以,这项工作正合我意。

工作的EnterpriseDB)也都需要我。除此之外,我喜欢做这些有用的事情(即使它们很枯燥)。所以,这项工作正合我意。

Bruce Momjian: 我有着亚美尼亚血统,并且一直居住在费城。我本来在大学主修的是历史教学,毕业后却在高中教计算机科学,并且为法律事务所开发数据库应用程序。这是一个很奇特的经历,而现在全身心的致力与开源社区的构建更是一种不可思议的经历。但是我相信,这项事业是很有希望的,而且我也非常的喜欢。

PG杂志: 您第一次听说PostgreSQL是什么时候,又是为什么选择了投身于这项事业?

BM: 作为一个数据库应用程序开发者,在工作的时候我会使用Ingres和Informix。但我希望能够找到一种数据库软件能够在我家里的UNIX电脑上使用。所有的这些商业型的数据库系统都是很昂贵的,而且我还想了解这些数据库内部是怎样工作的。而Postgres正是这样理想的数据库,它既有商业数据库的框架,也有大量开源的编码能够让我一饱眼福。

PG杂志: PostgreSQL这些年的变化很大。您

是怎么看待Postgres的成功呢? 其中是不是有让您意想不到的惊喜,您在一开始的时候预料到今天的成功吗?

BM: 我开始的确不知道它会不会成功。Postgres是个有趣的项目,而且在工作中通过和一些智者(比我更聪明)的交流我也学习到了很多的知识。我也没法设想它的未来到底是什么样,因为那时在开源社区的建设上面Postgres能够借鉴的成功案例太少了。

在早些年开始尝试投入到Postgres的开发中的时候,我正做着按小时计费的工作,当然,为Postgres做工作是没有报酬的。但是我觉着这项工作特别有意义,所以就一直坚持了下来。我相信我所掌握的这些技能总有一天会派上用场。

PG杂志: PostgreSQL可以说是现存的历史最悠久的开源项目之一,它生命力如此旺盛的秘密是什么?

BM: Wow....., Postgres的飞速发展也着实令我大吃一惊。我想其中有一个原因就是小群致力于生产最高质量软件的人激励影响了众多分散在世界上的志愿者,并带领他们一起努力造就了今日的辉煌。

可能会存在很多因素来干扰我们这个团队正常的运行,但是现在我们这个团队在茁壮的成长并且越来越有创造力。融合大家的力量,并且怀着谦卑的心来对待每一位志愿者,让他们感

觉到所做的工作都是受人欣赏的——这就是我们成功的秘密。

PG杂志：那您现在主要在做哪方面的工作？

BM：我喜欢做哪些别人不愿意做的事情。过去我常做一些打补丁方面的工作，但现在已经告一段落。我经常寻找那些需要做的事情，然后不断的努力去完善它。比如说，我做过打补丁，检测漏洞，完善文档资料，写过展示文稿，而且出席会议，等等。

PG杂志：您一天工作多长时间？

BM：有时候6个小时，有时候12个小时，这要看具体是什么事情，假如是在开会，可能一直都不会休息。

如果我要一整天都在工作的话，我会在晚上9点到10点间放下工作去稍微活动一下。但是最近我一直在旅行，所以只能通过EMail来完成这些工作。很多时候，我会帮家里做些家务之类的，当然因为我在家工作做这些很容易，另外通过做这些小事情可以把一天的时间分割开来，能够更为有效的利用。

这个工作很适合我，因为我喜欢做这些，而且这个社区（和我现在工作的EnterpriseDB）也都需要我。除此之外，我喜欢做这些有用的事情（即使它们很枯燥）。所以，这项工作正合我意。

More Momjian...

如果想了解Bruce的更多信息，可以登录他的网站：博客，会议安排，演讲，网络广播，简历，出版的书籍，等等。

<http://momjian.us>

那么，什么才是真正的工作呢？有时候我会整个早上都在跟社区的成员讨论开源项目、漏洞和任务。那也是工作吗？我想是的，当然它同样也是非常有趣的一正是因为这些工作不断的促进文档的修订和代码的完善。那现在我接受采访也是工作吗？我想是的，因为它同样也很有意义。

PG杂志：您现在在PostgreSQL社区的发展中是扮演着什么样的角色呢？

BM：我是Postgres的核心成员，也是它的首席架构师，而且是pg_upgrade工具的主要维护人，这个工具最近变得很流行。同时我也认识大多数的Postgres成员，所以有时候我也会充当一个社区联络员。

PG：就您看来，即将发布的PostgreSQL9.1版本有最大的特色是什么？

BM：下面是一个简短的统计清单：

- *Unlogger table可以更好地控制NoSQL的工作负载
- *同步复制可以提高可靠性
- *SQL/MED（控制外置数据）（平面文件（即没有相对关系结构的记录的文件），其它数据库的资料）
- *字段级编码，支持按列排序的规则
- *安全标签，集成了SELinux
- *真正的基于锁的串行化隔离（同样具备快照隔离）
- *可更新的CTE功能（支持非递归查询）
- *最近相邻索引功能
- *Extensions（插件）命令
- *PL/Python功能修复（PL/Python功能在9.0版本中已经支持）

PG杂志：PostgreSQL通常被认为是“最先进的开源数据库”。你认为在将来它能超过其它非开源的关系型数据库而成为市场上最先进的数据库吗？



BM: 是的, 我想Postgres的开发者们正明确的朝着这个目标前进。我认为最重要的不是你如何去对抗竞争者, 而是你自身要不断的完善。如果你发展的速度远远高出了其他人, 你自然就压倒了他们, 即使这会花费数十年的时间。很明显, Postgres巩固和完善自身的速度已经超过了其他非开源的数据库, 现在我们在一些领域已经处于领先地位。所以假以时日, Postgres最终会成为最先进的数据库。

PG杂志: 你是怎么看待NoSQL运动的呢? 它会对Postgres产生威胁吗?

BM: 我想NoSQL中的一些最好的创意会慢慢地融入到现存的数据库系统中, 而且NoSQL数据库也会逐渐丧失他们的吸引力。从过去的object数据库和XML的数据库的身上, 我们似乎可以看到NoSQL的未来。

PG杂志: 在Postgres社区里, 您是不是有一些想做却又没做好的事情呢?

BM: 在早期我们确实错过了一些事情, 但我们正努力用新的技术弥补这些不完善的地方。从这层面上来看, 错过的那些已经显得不重要了。我们希望将来Postgres有更大的发展, 而不仅仅是弥补之前的损失。有一个好消息, 就是我们的研究和开发水平越来越高, 现在所达到的成果已经不仅仅是完善之前的失去的那些了。这些新成果在9.1版本中完美的体现出来, 而且我们会继续研发9.2版本, 如支持本地使用JSON。

PG杂志: 除了开源和PostgreSQL之外, 您还有什么别的爱好?

BM: 我喜欢读书, 旅游, 喜欢和家人在一起做事情。

如何优化PostgreSQL的性能

PostgreSQL是现今最先进和灵活的开源数据库，但这些功能和灵活性也同样带来相应的问题。PostgreSQL的开发者怎么样去优化每个用户的默认配置？不幸的是，他们还不知道怎么做。

问题是每个数据库的差别不仅仅要体现在设计上，也要同样体现在满足用户的个性化需求上。有些数据库仅仅是用来记录海量的数据，这些数据大部分从来都不被查询。有些本来只能用来存储静态数据的数据库可能突然要求实现查询功能。而且很多系统都会有一些不同的设置，如对读写的等级权限设置。透过这些复杂的外部环境还要进一步深入到你个人独特的表格结构、数据和硬件配置，希望你能想象的到，这个优化过程是多么的困难。

PostgreSQL自带的默认配置是一个稳定的配置，针对每个用户的不同需求和硬件配置来营造一个相对均衡的配置环境。这篇文章的目的在于帮助不同层次的用户更好的了解PostgreSQL的性能优化。

优化常见流程

要学习如何优化你的PostgreSQL数据，首先要了解查询的生命周期。下面是一次查询的过程：

1. 传输查询字符到数据库
2. 查询字符的解析
3. 查询计划
4. 检索硬盘上的数据
5. 传输查询结果给用户

第一步是把查询字符串（你输入的SQL数据库指令或应用用途）传输到数据库后台。在这一步，你并不能做过多的优化；但是，如果你的查询量比较大且不能事先规划好查询命令，它会智能的帮助你把这些指令规划后输入进数据库，从而提高数据传输速率。

一旦这些字符串进入数据库，就会被解析成相应的代码。这一步也可以通过优化存储过程来简化。

查询计划是PostgreSQL真正开始做的一些工作。这一阶段主要是检查查询系统是否已经准备好。如果你的PG版本和客户端支持，它会分析你的SQL，以确定检索数据最有效的方法是什么。我们应该使用索引吗，如果需要使用哪一个索引？这两个表的哈希值是否匹配？这些都是数据库在这个阶段要做的决定。但如果查询是事先已经规划好的，这一步就可以简化掉。

PostgreSQL有一个计划，就是让数据库系统自主选定一个最好的方法来检索数据，现在也是时候实现这个计划了。有一点需要指出，优化选项能够提供一些帮助，但实际的效果还是会受您自己硬件配置的影响。

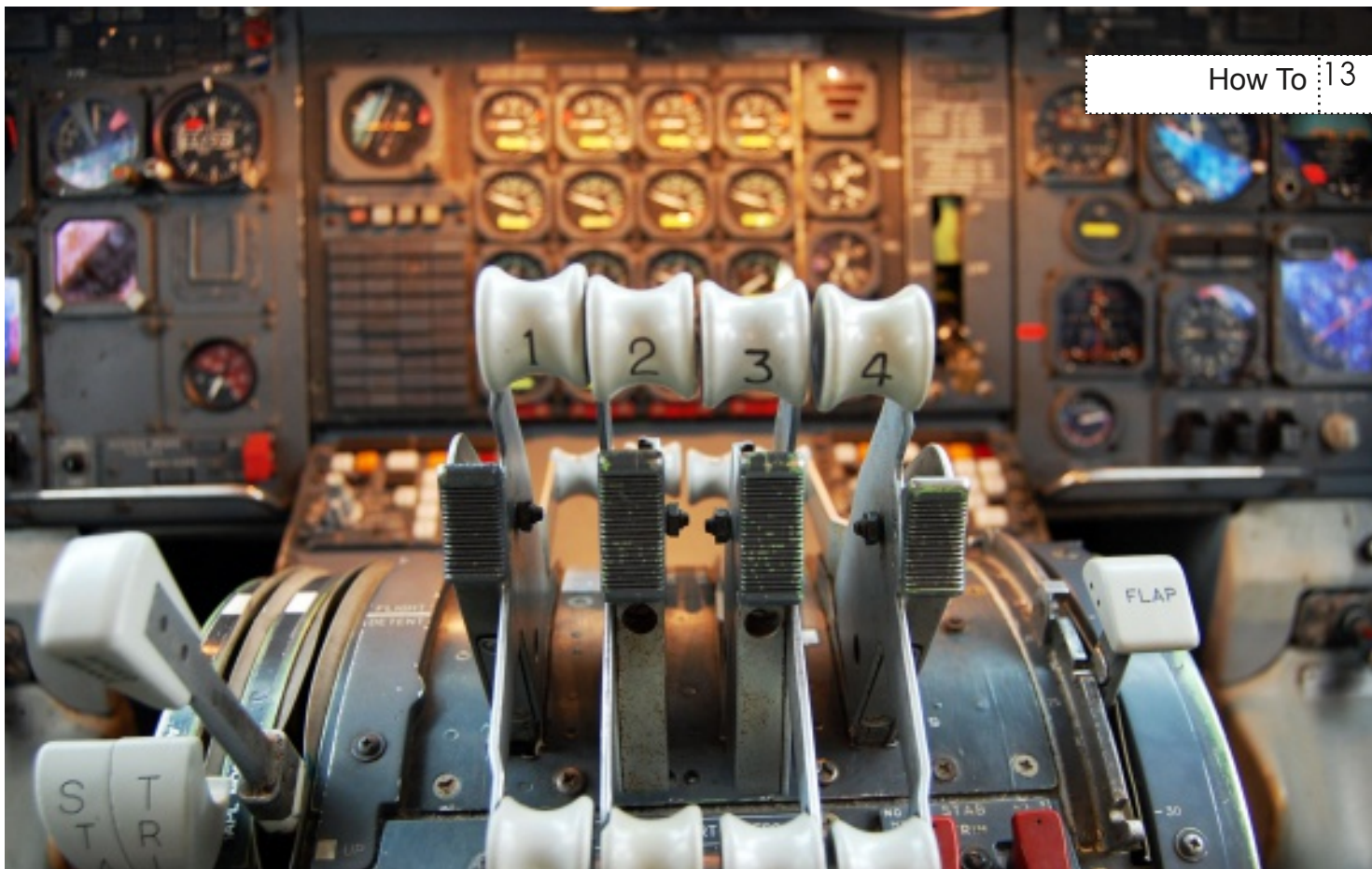
最后一步是将查询结果传送给客户端。虽然这一步不会真正的帮助优化，但你应该知道，返回到客户端的所有数据都是通过硬盘和数据线进行传输的。所以，保证数据行和列的最简化通常能够提高运行的性能。

常规参数优化

通过postmaster(postgresql.conf)中一些选项的设置可以显著的影响PostgreSQL的性能。下面给出一些最常用的选项并且说明它们是如何影响整体性能的：

max_connection:

这个选项用来设置数据库后台同一时间最大连接数。使用此功能能够确保后台运行的程



序量不至于超出负载范围而影响数据库的整体性能。根据你的设置，它完全拒绝所有额外的程序连接，而不仅仅是降低其它程序的运行性能。

shared_buffers:

编辑这个选项是提高数据库服务性能的最简单的方法。这个选项的默认值对大多数当前的硬件配置的都很低，而专家们通常都认为应该在系统上设置25%的可用RAM。在这里要指出一点，像大多选项一样，你应该自己调试一下设定不同的值（高值和低值）后系统运行的状态。大多数人会发现，如果设置超过第三个值的时候，不仅不能提高速度，反而会降低运行的性能。

effective_cache_size:

这个值是用来告诉PostgreSQL的优化者有多少内存能用于高速数据缓存，并帮助它决定是不是使用索引功能。这个数值较大幅度的增长会提高使用索引的可能性。Share_buffers的设置占用的内存加上操作系统的缓存占用的内存构成最终的内存的总使用量，通常情况下，内存使用量要占系统总内存的50%以上。

work_mem:

这个选项是用来控制排序操作和使用哈希表所占用的内存量。在你为应用程序排序的时候，可能会需要增加相应的内存量，但这个操作要采取必要的保护措施。这个参数并不只是局限于某一系统范围内，在所有系统上都是如此。所以，如果一个复杂的查询程序包含多个要求排序的操作，就要在内存中设置多个work_mem内存单元。更不用说，当有多个后台进程同时运行时，也要这样做。但如果这个值太大的话，查询时会引起数据库频繁发生数据交换。这个选项在旧的PostgreSQL版本里被称为sort_mem。

max_fsm_pages:

这个选项有助于控制"free space map"（自由空间）。当一个数据被删除的时候，它并不是立即从硬盘上消失，而是被标记为"free"并将它存放在"free space map"中。这个空间里的数据可以被重新利用，你可以通过新的insert操作将数据库写入表中。如果你的操作中删除和插入率很高，可能就需要增加必要的值，以避免表格过度膨胀。注意一下，max_fsm_page_s功能在PostgreSQL 8.4及以后的版本中被

移除了。

fsync:

这个选项决定了你的WAL页面在事务提交前是否会被写回硬盘。开启这个功能后系统会变得更加安全，但同时也降低了写入的性能。如果fsync被关闭，那么你的数据损坏后可能无法恢复。当然，关不关闭这个功能还要你自己权衡。

Commit delay and commit siblings:

这些选项是通过并行事务写入的迅速提交来优化提高性能。如果有commit_siblings出现在后台进程中，就会被commit_delay延迟执行并集中进行编译，当然这个延迟时间也仅仅是以微秒来计算的。

random_page_cost:

这个选项是用来控制PostgreSQL无序磁盘中的读取方式。设置的值越高，分配的内存就越多，从而使得磁盘的扫描变得更为快速。

请注意：这些选项会占用共享内存，它可能会需要你增加系统的共享内存去匹配这些选项的设置。

与硬件相关的性能问题

显然，数据库服务器使用的硬件类型和质量也会大大的影响数据库的性能。这里有一些小提示，供你为数据库服务器购买硬件时使用（按重要性排序）

RAM:

内存越大，你将会有更多的磁盘高速缓存。因为内存I/O速度是硬盘I/O速度的上千倍，所以增加内存会极大的提高性能。

Disk Types (磁盘类型):

显然，快速SAS磁盘是你最好的选择，然而高端的SATA磁盘也是不错的选择。使用SATA磁盘要便宜好多，在相同的预算下，使用SATA磁盘要比SAS磁盘性价比更高。

DISK Configuration (磁盘配置):

最好的配置是尽可能多的将磁盘配制成RAI

D1+0，而且最好让它自主的把所有的事物日志(pg_xlog)都备份存储到一个单独的磁盘(或磁带)里面。使用RAID5来做并不是一个好的选择，除非你有6个以上的磁盘。在新版本的PostgreSQL中你也能使用表空间的选项将不同表格、数据库和索引放在不同的磁盘上以进行性能优化。比如，把你常用的表格放在快速SAS磁盘上，把那些用的较少的表放在稍微慢一点的IDE或SATA磁盘上。

CPUs:

CPU越多越好，但是如果你的数据库并不需要过多复杂功能的话，你的钱最好还是花在改善RAM和磁盘子系统上。因为通常来说，RAM和磁盘空间越大，系统的性能也会越好。这是因为额外的RAM能够帮助你更快的访问磁盘。而磁盘空间变大了，不仅能够提高数据吞吐量并且能够避免磁盘拥堵。

另外一个好办法就是将你的应用程序代码和数据库服务器放到不同的硬件上。这样做不仅能提供更多的硬件支持以提高数据库服务的质量，同时操作系统磁盘高速缓存区也能存放更多的PostgreSQL数据。注意，在这些专属区域内，其它的应用程序和系统数据都不会被存储。

举个例子，如果你有一台Web服务器和一个数据库服务器，你可以使用交叉线来将它们跨接到一个单独的以太网接口上，来处理Web服务器上的数据库网络流量，以减少任何可能发生的网络瓶颈。如果你有多个服务需要访问同一个数据库服务器的时候，你也可以为这个数据库创建一个完全不同的物理网络来确保访问通畅。

应用层性能问题

这些问题通常对有状态应用和无状态应用都会有影响。一个比较好的解决方法就是让服务器对你经常使用的查询行为进行备份。这样通过使用之前的缓存，可以节省总的查询时间，而且把每次的查询行为备份下来，也方便以后使用。

然而，应该注意的是，如果你输入事先准备好一个查询的占位符(如'column_name=?')，系统并不一定会选择最优计划去执行。比如，你的查询中包含一列布尔类型的列‘active’，而且对值为false的使用了部分索引，执行计划将不会使用它，因为不能确定这个值在传递过程中将会是true还是false。

显然，你也可以在这里利用存储过程以减少数据传输和分析量和规划典型查询的生命周期。当然，最好的做法就是分析你的应用程序和常用的查询和数据操作，把它们写进存储过程。

About the article

The original presentation is available here : <http://pgmag.org/0015>



About the author

Frank Wiles is an Open Source consultant specializing in scaling and performance . He founded Revolution Systems (revsys.com) to help businesses take full advantage of all the benefits of the Open Source Software revolution.



如何在Linux系统下优化PostgreSQL性能



对PostgreSQL而言，Linux是最主要的操作系统之一。其中原因之一就是Linux具有良好的性能。但是如果你想要达到最佳性能，这里也有几点可供选择的优化建议。

文件系统

较老版本的Linux默认使用的是ext3的文件系统，该系统比较可靠但是速度却不是最快的。新版本ext4速度较快但还并不是很成熟，因为在配置的时候测试负荷往往会很重。而处于中间的XFS版本，则是平衡了两者的优点，它运行比较稳定而且速度较快。所有的这些文件系统都得益于使用了“noatime”来配置参数，利用它可以排除或更新一些已经访问过的，对数据库而言不是特别重要的信息。

数据预读

PostgreSQL依赖操作系统做一定量的预先读取，而这对于通过序列扫描包含大量数据的表格是非常必要的。Linux具有优秀的预读功能，但是它的默认设置只有很低的block (256)，不过，设置为4096 blocks倒是一个较好的选择。预读的设置块大小如下：

```
/sbin/blockdev --setra 4096 /dev/sda
```

可靠性及快速写入

想要PostgreSQL发挥良好的性能需要选择一个合适的硬件环境，如果硬件设施配置或安装错误，你的数据就很危险了。你可

以在<http://pgmag.org/0017>网站上找到一些有关配置硬件设施和操作系统的文章，来帮助你配置性能稳定、读写能力可靠的数据库。正确选择合适的硬件也很重要，例如你增加一个电池支持系统提供写控制器，就能大大提高写的性能。但是，XFS和ext4文件系统需要“nobarrier”这个挂载参数才能使其很好的工作。

写缓存

在写入性能这方面，PostgreSQL期望操作系统能够缓冲正常的写入，只有当后台的checkpoints完成后才强制将数据写进磁盘。Linux 2.6.22核心版本的写入缓存比之前版本增大很多，包括现在仍然流行的Red Hat Enterprise 5。你可以用以下的方法在旧版本上获取新的默认配置：

```
echo 10 > /proc/sys/vm/dirty_ratio
echo 5 > /proc/sys/vm/dirty_background_ratio
```

这些设置允许分配10%的RAM用作缓存，也因此PostgreSQL率先实现了提高5%运行速度的预期目标。当然，这些数值可能对于许多仅有千兆字节的内存系统而言还是太高。从Linux核心2.6.29版本开始，就可以利用新的dirty_background_bytes和dirty_bytes配置把这些数值设置的更小一些。即是当你使用这些新配置之后，就会按照你所设置的值来代替以前的值。但是请注意，在PostgreSQL的一些批量操作中，尤其是VACUUM，如果你Linux的高速写入缓存设置的过低，一些批量操作的速度将会明显减缓。

I/O 调度策略

尤其在一个包含大量的混合读写操作的工作负载时，Linux内核的I/O调度程序的性能会因为系统的繁忙而受到影响。仅仅通过做一些简单的性能测试，很难发现参数设置前后的运行性能的不同，但那些选项已

经被确认可以对性能有很大的影响。即使对于一个复杂的数据库工作负载，调度程序同样有很大的影响力。

CFQ是一个默认的调度程序，它是一个很好的出发点，但是它并不能适用于所有的情况。而且，我们首要考虑的是调度的时限问题，通过它能够给出较合理的响应时延，因为当进行读写双重操作时响应时延是非常重要的。

假如你的磁盘中有一些智能控制模块，例如一个很好的RAID控制器或者外部SAN存储器，那么使用I/O调度就不会有任何的效益，只是增加配置成本而已。在这种情况下，就需要利用NOOP调度来提供最好的性能。NOOP调度只需做少量的工作，大部分事项由磁盘或者控制器做最终决定。

作者介绍

Greg Smith是2ndQuadrant US的首席顾问，同时也是“高性能PostgreSQL 9.0”的作者。网站地址为<http://www.2ndquadrant.com/books>，Greg Smith 的博客地址为

<http://blog.2ndquadrant.com>

该博客会定期更新Linux和PostgreSQL的调整策略。



PostgreSQL在法国社会保障系统 每天超过10亿SQL

在2010年法国社会保障系统(CNAF)决定将系统中一个重要的组成部分转移到PostgreSQL上。这是一个相当成功的案例，因为它表明了对于大规模的管理，高负荷的工作量以及大规模的数据库而言，PostgreSQL都是最佳的选择。

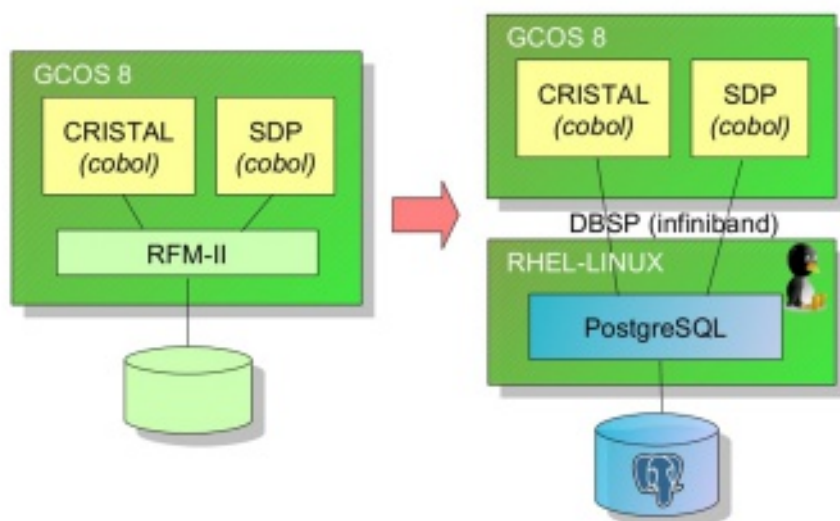
法国社会保障系统(CNAF)已经将他们的Bull Interel - RFM2数据库移到了PostgreSQL上，这一决定是CANF承诺走向开源解决方案的重申。也因此CANF每个月可以通过PostgreSQL项目获得3亿欧元的利润。该项目正在全面运行中，其中最主要一部分是改变这种高灵敏度组织的信息系统。

在有限的成本预算下，为了提高其服务质量，CANF将重点放在寻找更好的性能和更强大的系统上面。为了达到该目的，它决定改变对 CRISTAL和 SDP (CRISTAL和SDP一共有20万条的应用软件代码) 的策略，加快其项目的现代化进展，当然这样做对于加强它项目负责人的地位和获得更高的利益都是至关重要的。就性能和开放性而言，CANF要寻找一个新的数据库来满足其标准，PostgreSQL无疑是一个最佳选择。

Marc Pavie是CANF信息系统管理的副局长，他解释道“选择PostgreSQL是我们承诺走向开源的实际步伐。在BULL作为项目的主

要承包商期间，我们没有遇到任何困难。但是我们从PostgreSQL提供的新功能获取利益，也促进了我们对信息系统的优化。但是最重要的是，解决方案提出的稳定性和高性能的要求在很大程度上给我们的服务质量带来了更高的挑战。”

该项目开始于2008年10月，紧接着是由BULL领导执行了一个可行性研究而且跟CANF团队合作建立了一个标准。到2010年4月，在这九个月的时间里还进行大量的测试，为项目的实施打下良好的基础。





关于CNAF

CANF是法国社会保障系统的一个分支，主要通过技术援助为家庭提供财政补助，进一步支助一些当地的集体性活动，并保障当地的社会生活（如市政厅和托儿所等）。每年CANF向110万申请人提供大约69亿欧元的支助，平均每人可以获得30万。它的信息系统包含高度灵敏的数据，例如法国政府对低收入家庭的最低收入保障是多少。

PostgreSQL所提供的新功能促进了我们信息系统的优化，也让我们获益匪浅

通过统计，我们一共花费了18个月的时间去迁移所涉及到的168个数据库，一共4TB的数据。从那时起，每天数十亿元的SQL查询在进行。现在日常的管理活动已经自动化，系统的监管也是由一个开放软件Nagios来执行的。另外，已经实施的架构上也具有高可用性，并且具有远程商业恢复点和基于本地服务的一系列措施。

一年过去了，所有的实施都已经完成，PostgreSQL数据库清晰的向大家证明了它的稳健性和高性能性。同时，批量处理时间有所减少，事务的响应时延也得到了改善。

Bull France 的首席执行官 Jean-Pierre Barb ris说道：“我们非常荣幸能在这个项目中支持CANF，这在欧洲是空前的。由此带来的好处不仅证明了开放源码工具具有强大力量以及稳健性，而且他们在帮助自己获取经济收益的同时，会以自己的力量去支持更多的政府创新型项目的实施。

CANF目前正在考虑改进PostgreSQL的一些功能特征，如发布PostgreSQL9.x，以追求数据服务的高可用性。

作者介绍

Damien Clochard是Dalibo公司的董事之一，同时他也是 PostgreSQL.fr网站的管理人，并且在法国参与了多个PostgreSQL的宣传项目。



PostgreSQL 9.1

将要带来的 Extension功能

PostgreSQL的新版本将给我们带来一些令人兴奋的新功能。其中Extension是一个关键性的提高，它使得软件供应商为PostgreSQL提供附加功能变得更加方便。

例如，将像ltree这样的contrib模块添加到一个数据库将变得非常简单：

```
$ select '1.1'::ltree;
ERROR: type "ltree" does not exist
LINE 1: select '1.1'::ltree;
          ^

$ create EXTENSION ltree;
CREATE EXTENSION
$ select '1.1'::ltree;
ltree
-----
1.1
(1 row)
```

同样的，要想删除它也非常容易：

```
$ drop EXTENSION ltree;
DROP EXTENSION
```

然而是最棒的是您可以在不同的schema下面装载Extension，并且在装载完之后您可以很方便的修改它。

这个补丁是2月8日，TOM Lane发布的。

Core support for "extensions", which are packages of SQL objects.

This patch adds the server infrastructure to support extensions. There is still one significant loose end, namely how to make it play nice with pg_upgrade, so I am not yet committing the changes that would make all the contrib modules depend on this feature.

In passing, fix a disturbingly large amount of breakage in AlterObjectNamespace() and callers.

Dimitri Fontaine, reviewed by Anssi Kääriäinen, Itagaki Takahiro, Tom Lane, and numerous others

Extension就是将各种数据库对象组合在一起以便于管理操作的一种方式。对于开始使用的人，所有的contrib模块都可以扩展。

这就意味着安装和更新软件变的更加简单。



PostgreSQL的测试版可以说是即将发布的PostgreSQL 9.1专业版本特征的快照。该测试版并不会正式投入使用。目的是想要大家预览和测试一下新版本的功能以尽可能快的得到一些反馈意见。如果你是一个开发者或者是DBA，通过参与测试PostgreSQL的测试版本，不但可以帮助消除漏洞，促使正式版本早日发布而且可以为新版本增光添彩。

<http://www.postgresql.org/developer/alpha>

让我们一齐来看看它是如何工作的：

```
$ create extension ltree;
CREATE EXTENSION
$ create table test (z ltree);
CREATE TABLE
```

假设，我们有一张表，表中有一列数据库类型为ltree。但是我不希望与ltree的相关功能影响到public模式，因此我决定将它移动到一个指定的模式中：

```
$ create schema ltree;
CREATE SCHEMA
$ alter extension ltree set schema ltree;
ALTER EXTENSION
```

表格同样也得到了改变：

```
$ \d test
Table "public.test"
Column | Type          | Modifiers
-----+-----+-----
z       | ltree.ltree   |
```

文章简介

本文原著刊登在：
<http://postgr.es/p/-6>



虽然这似乎没什么大不了的，但其实际上它非常重要。

将所有的对象组合在一起，意味着你可以将它们升级，并且转储/恢复操作也是比较简单的。转储可以备份所有的函数、表格、类型的定义，但转储也可以只包括：

```
# pg_dump | grep ltree
-- Name: ltree; Type: SCHEMA; Schema: -;
Owner: depesz
CREATE SCHEMA ltree;
ALTER SCHEMA ltree OWNER TO depesz;
-- Name: ltree; Type: EXTENSION; Schema: -;
Owner:
CREATE EXTENSION ltree WITH SCHEMA ltree;
-- Name: EXTENSION ltree; Type: COMMENT;
Schema: -; Owner:
COMMENT ON EXTENSION ltree IS 'data type for
hierarchical tree-like structures';
z ltree.ltree
```

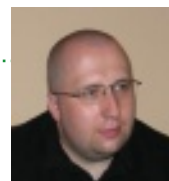
任何一个尝试过将PostgreSQL的一些数据转储加载到另一个PostgreSQL服务器上的人，都会非常欣赏这一点的。还有一点，当加载扩展模块时，路径会自动的更改，但发布的新版本模块中没有该功能（一些新的或者旧的路径会消失）。

编写Extension看起来并不复杂，所以我想，过不了多久我们就能看到pgFoundry项目的Extension方案了。

作者简介

Hubert "Depesz"

Lubaczewski在OmniTI是一个数据库架构师，他的博客专注于PostgreSQL新版本的发布，博客地址为：<http://www.depesz.com/>



让PostgreSQL掌管一切

目前，在数据库领域的一个新话题是关于NoSQL的使用，如MongoDB，Cassandra和Couchbase。

到处可见一些文章宣扬关系型数据库都要死掉，所有的数据最终将保存在灵活的数据储存单元中。最近，在 PG East 有一个关于MongoDB的交流会，所以我有机会和更多的人交流。我发现，无论是已经使用NoSQL的人还是准备使用它的人，他们使用NoSQL的理由都是完全不同的。

有些人有正当的理由选择使用NoSQL，而不使用像PostgreSQL这样的传统关系型数据库。关系型数据库存在关系原理中固有的限制，当我们使用文档存储结构时，关系型数据库就不行了。当我们插入亲子关系时，例如一个web应用用户他拥有2个地址，3个电话号码，那么使用MongoDB比使用PostgreSQL处理起来要快很多。使用PostgreSQL，我们会将用户的信息分解然后插入到表格中，其中两行插入到地址表格，三行插入到电话号码的表格，这里我们忽略那些麻烦的外键约束导致效率降低带来的影响。

但是使用MongoDB，我们只需插入一个BSON文档就能够涵盖所有的信息。需要具有极高吞吐率的应用程序还是比较适合选择NoSQL模式，所以在此也很难分辨非关系模型数据库性能的优劣。

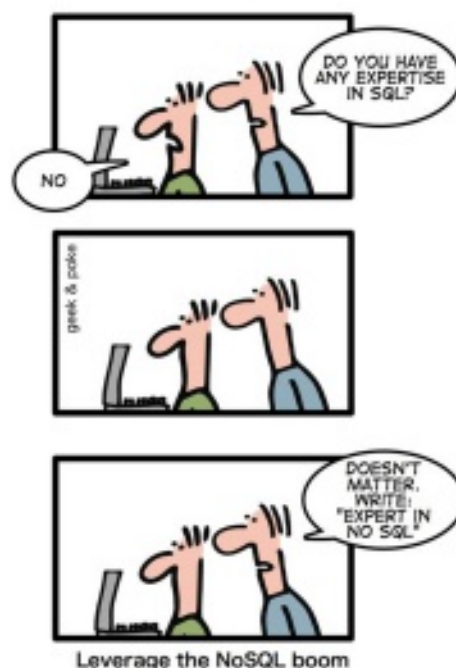
然而，许多人并不会遇到这种特殊的使用情况，而是仅仅把NoSQL作为一个持久的数据存储库来使用。对于许多这样的用户而言，PostgreSQL是一个更好的选择，但是不幸的是对于网页开发者而言，SQL语言就如同一种外语。

大多数的应用程序都会涉及到许多表格信息

，这是PostgreSQL相对于NoSQL所具有的优势。比如，为了显示上个月纽约用户的应用程序中的所有销售清单，如果使用NoSQL的MongoDB就需要从纽约找到所有的用户，然后在列表中输入相关信息，然后进一步查询他们的销售成绩。

而使用PostgreSQL的开发者只需要简单的加入一个销售表，该销售表只是专一的用来查询用户的地址表。然后PostgreSQL将对查询结果进行优化，如在纽约着手实施了一个包含10万用户的方案，但最后只有10个销售额。根据表格数据的统计情况，PostgreSQL将自主选择一种更有效的方法，将所有的销售和使用情况作为标准以检测出最后的结果。但使用MongoDB，开发商还必须了解哪一个订单是最有效的，并且知道哪一个是最经常出错的。

HOW TO WRITE A CV



Leverage the NoSQL boom

应用程序结构方式的不断发展，意味着关系型数据库一统天下的局面已经过去，但是关系型数据库并不会淡出历史的舞台。

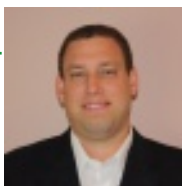
想要继续保持关系型数据库的主导地位，进一步推广PostgreSQL的发展，最重要的是探索怎样和新的数据存储方式交互融合，和平共处。

PostgreSQL 9.1版本的新增功能SQL/MED就是关键性的起步。结合MongoDB外部数据的包装器，它将允许MongoDB文档以PostgreSQL行的方式出现，开发者也可以将NoSQL的数据和关系型数据组合在一起使用。想象一下，仅仅通过一个简单的查询语句，你就可以搜寻到MongoDB实例的数据，甚至能自由的提取你存储在Oracle数据库里的财务信息，最后还能方便的把这些查询到的结果放入你自己的数据库中。

加入SQL/MED功能的PostgreSQL把发展方向定位为：要让PostgreSQL统治并融合所有其它的数据库。

作者简介

Jim Mlodgenski是Cirrus Technologies, Inc.公司的创始人，该公司作为一个顾问和产品开发的组织，致力于提供可扩展的数据架构。



什么是NoSQL?

NoSQL是非关系型数据库的统称，它打破了历史悠久的关系型数据库和ACID协议一统天下的局面。NoSQL类型的数据库数据结构定义比较松散，并不要求使用固定的表格模式，也尽量避免参与到操作中。该术语从2009年早期开始流行。

NoSQL数据库的目的是支持大规模网络访问和抵制关系型数据库模式（核心理念，文档，图形等）。NoSQL有特有的优势，但其功能比SQL数据库少的多。而且它并不是各方面都很强，它主要倾向于案例使用这一方面。

SQL/MED的来临

SQL/MED（外部数据的管理）是对SQL标准的一个扩展，SQL/MED定义了外部数据的包装和数据链路的类型，允许DBMS的把数据集成存储在数据库外。

这个规范早在PostgreSQL 8.4版本已经得到实施，同样在即将发布的PostgreSQL 9.1版本中也将继承并进一步延伸。例如第一个外部数据包装器—file_fdw，它可以用来访问服务器文件系统中的数据文件。

<http://wiki.postgresql.org/wiki/SQL/MED>

使用pgFouine对SQL查询进行分析

pgFouine是一个相当好的Postgres查询记录分析工具。和pqa一样它能够从一个Postgres记录文档生成一个html的网页报告，通过Postgres的日志文件可以看出哪些查询运行的较慢，哪些查询运行的最快，以及其他统计数据。

下面是安装和使用pgFouine的基本步骤：

1、如果你还没有pgFouine，你可以通过yum或者手动安装php；

```
yum install php
```

2、安装最新版本的pgFouine；

```
wget http://.../pgfouine-1.2.tar.gz
tar -zxvf pgfouine-1.2.tar.gz
```

3、在pgFouine.conf中你必须更改默认的前缀，例如记录的目的地址是stderr而不是syslog；

```
log_destination = 'stderr'
redirect_stderr = on
log_line_prefix = '%t [%p]: [%l-1] '
log_min_duration_statement = 0
```

设置log_min_duration_statement为0，将会记录所有的SQL语句

4、重新加载Postgres配置文件：

```
sudo -u postgres psql
select pg_reload_conf();
```

5、你可能必须使Postgres日志更新（这样就会生成一个新的日志的报告文档）：

```
select pg_rotate_logfile();
```

6、现在在你的应用程序中做一些事情（即执行应用程序的用户可能会做的一些事情），捕捉一些应用程序运行时所执行的SQL语句。

7、生成一个报告，在命令行运行pgFouine.php,使用最新的Postgres日志；

```
./pgfouine.php -logtype stderr \
               -file postgresql.log \
               > report.html
```

最终的报告结果看起来是这个样子的：

pgFouine: PostgreSQL log analysis report

Overall statistics | Queries by type | Queries that took up the most time (N) | Slowest queries

Normalized reports are marked with a "(N)".

- Generated on 2011-3-31 19:30
- Parsed /var/lib/pgsql/data/pg_log/postgresql-2011-3-31_183840.log (356,800 lines) in 50s
- Log from 2011-3-31 18:39:10 to 2011-3-31 19:29:52
- Executed on foo.com

Overall statistics ^

- Number of unique normalized queries: 433
- Number of queries: 35,178
- Total query duration: 44.0s
- First query: 2011-3-31 18:39:10
- Last query: 2011-3-31 19:29:52
- Query peak: 6,073 queries/s at 2011-3-31 19:17:19

Queries by type ^

Type	Count	Percentage
SELECT	5,954	16.9
INSERT	26,538	75.4
DELETE	1,606	4.6

Queries that took up the most time (N) ^

Rank	Total duration	Times executed	Av. duration (s)	
1	17.0s	281	0.06	COMMIT;
2	4.7s	122	0.04	INSERT INTO foo SELECT DISTINCT Bar

文章简介

本文原著刊登在：
<http://pgmag.org/0024>



作者简介

Matt Tescher 是美国旧金山Elastra的一个软件工程师。

