北京大学 PEKING UNIVERSITY

瀚高 HIGH GO

课程总结

开源开发实践-第十二周

David & Cary

课程介绍 - 重温

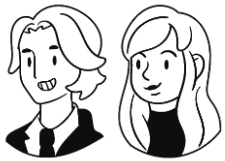
我们学到什么？

- 这是一门围绕PostgreSQL开源数据库的实践课程。
- 本课程侧重于讲解PostgreSQL内部的功能实现逻辑。
- 掌握C语言在Linux平台上的基本编程技能和代码调试工具。
- 学习数据加密和密钥管理的知识基础，应用场景和实践。
- 并把这两种理论知识应用到当前的PostgreSQL开源数据库。
- 了解当前PostgreSQL国际社区的研发及工作模式。
- 与PostgreSQL国际社区核心成员（Bruce Momjian）进行在线交流。
- 个人的课程项目有机会提交给PostgreSQL国际社区成为社区贡献。
- 以后在数据库领域发展和自我的修炼！

What Is Security?

The Big Picture

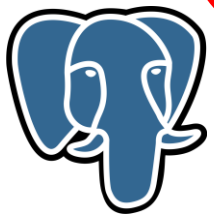
TDE and KMS happen here!



PostgreSQL Users (Client)

Data Security
- TLS / SSH

Firewall



PostgreSQL Server

Vulnerability
- Software bug

Data Security
- Data encryption
- Key management

Hard disk

Authentication Server

User Security
- password check
- User authentication
- Privilege checking

Privilege Settings

Bad Guy

Network Security
- Firewall Rules
- DOS attack

Where is security involved?

Cryptography

Introduction

- Secret-key Cryptosystems
  - ✓ plaintext
  - ✓ ciphertext
  - ✓ encryption
  - ✓ decryption
  - ✓ keys
  - ✓ Alice and Bob, and Eve
- *Public-key Cryptosystems*
  - ✓ *Public key*
  - ✓ *Private key*
  - ✓ *Rivest, Shamir and Adleman*
- Block and Stream Ciphers
  - ✓ DES, 3DES, AES…
- *Message Integrity*
  - ✓ *Message Authentication Codes (MACs)*
  - ✓ *Signature schemes*
  - ✓ *Nonrepudiation*
  - ✓ *Certificates*
  - ✓ *Hash Functions*
- *Cryptographic Protocols*
  - ✓ *Identification scheme*
  - ✓ *Key distribution scheme*
  - ✓ *Secret sharing scheme*
- *Hybrid Cryptography*
  - ✓ *TLS…*

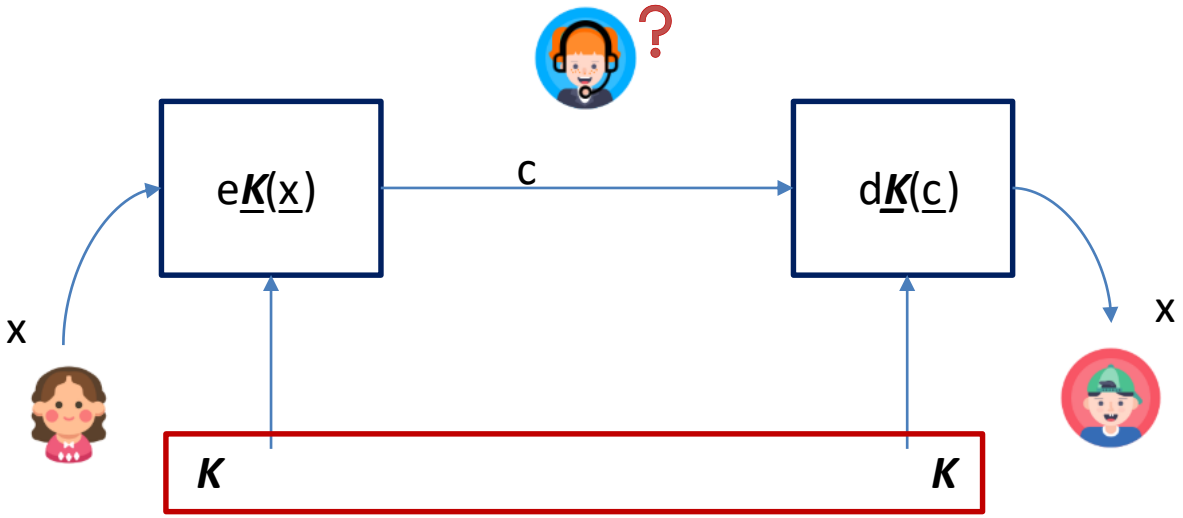
Cryptography

Summary

- Cryptology
  - ✓ Cryptography
    - ➢ Secret-key cryptosystems
      - ▪ Shannon’s theory
      - ▪ Confusion and diffusion
      - ▪ Stream cipher and Block cipher
    - ➢ Public-key cryptosystems
      - ▪ Diffie and Hellman
      - ▪ Factoring Integers and Discrete Logarithm
      - ▪ RSA cryptosystems
      - ▪ Key distribution
  - ✓ Cryptanalysis

Cryptography

Classical Cryptography



C语言基础

基础软件：大型数据库

- Oracle
- PostgreSQL
- MySQL
- MS SQL Server

...

目前排在前10名的大型数据库，有80%都是主要由C或者部分C++编写，尤其是Oracle，PostgreSQL，MySQL，MS SQL Server。另外，Redis，MongoDB，MariaDB，IBM-DB2也是主要靠C语言完成。

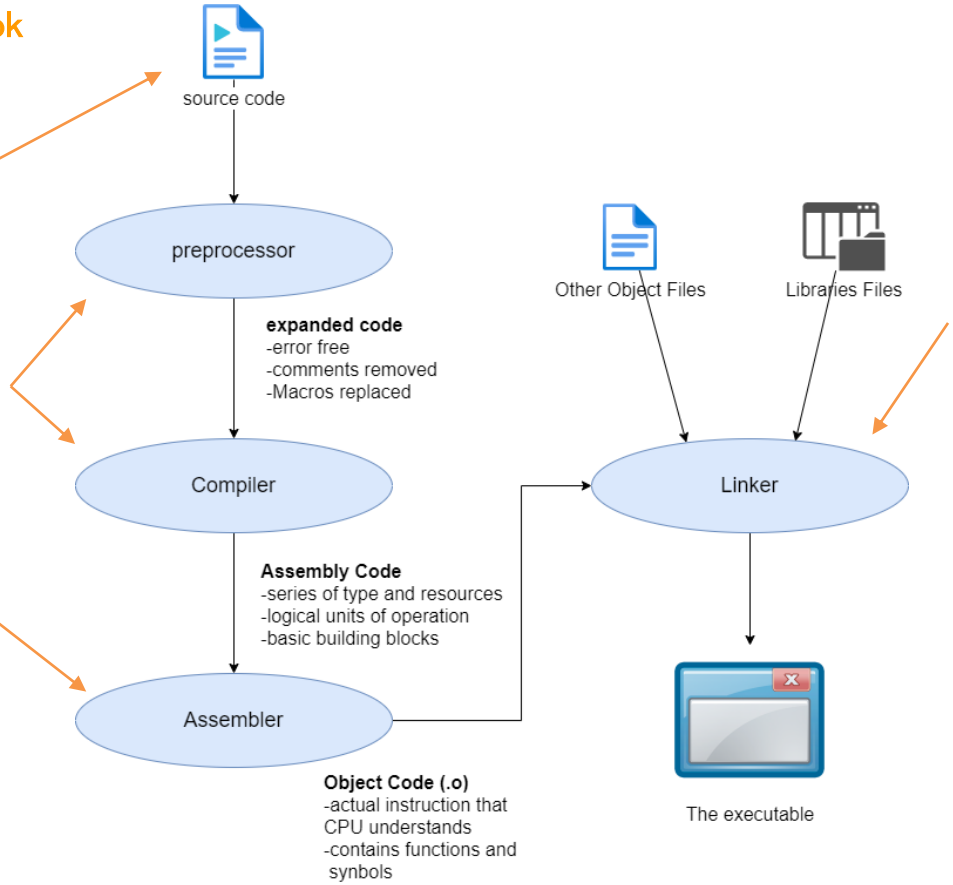
The Compilation Process

A Behind the Scene Look

That is the source file you will be working on. Normally ended with (.c) extension
Ex: bufmgr.c

Preprocessor then checks the syntax and compiler converts the expanded code into an assembly code

Then the assembly converts the assembly code into an object file. Normally every single (.c) file will produce one (.o) file.
Ex: bufmgr.o



Finally, the linker combines all of your object files (.o) including:

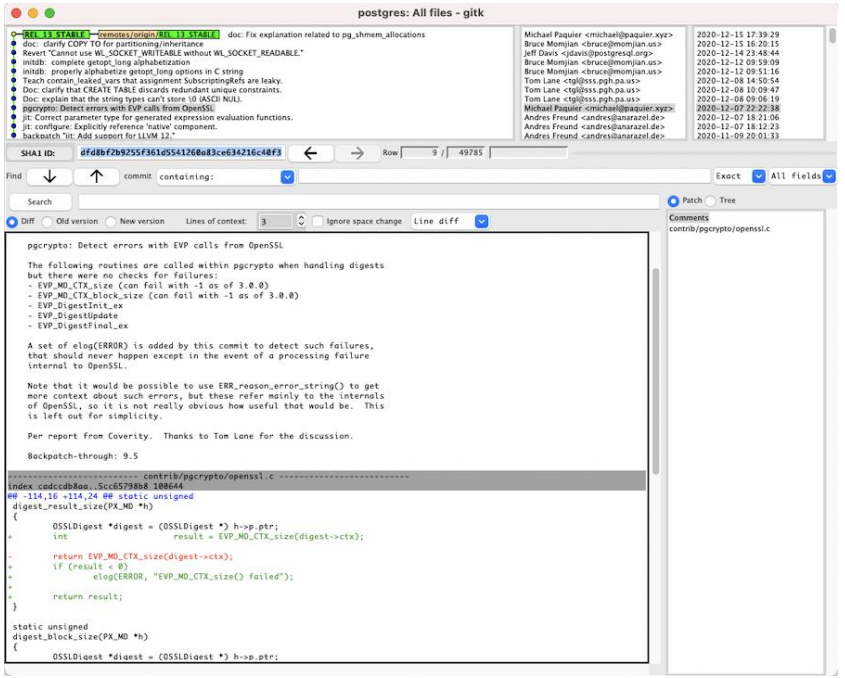
- Ones you compiled
- Ones you introduced to the system
- From shared and static libraries (.a) or (.so) files

Into one executable file. This is the final result of the compilation process.

Git基础

Git图形界面工具

- 目前最流行的源代码管理工具
- gitk, GitHub Desktop,等等



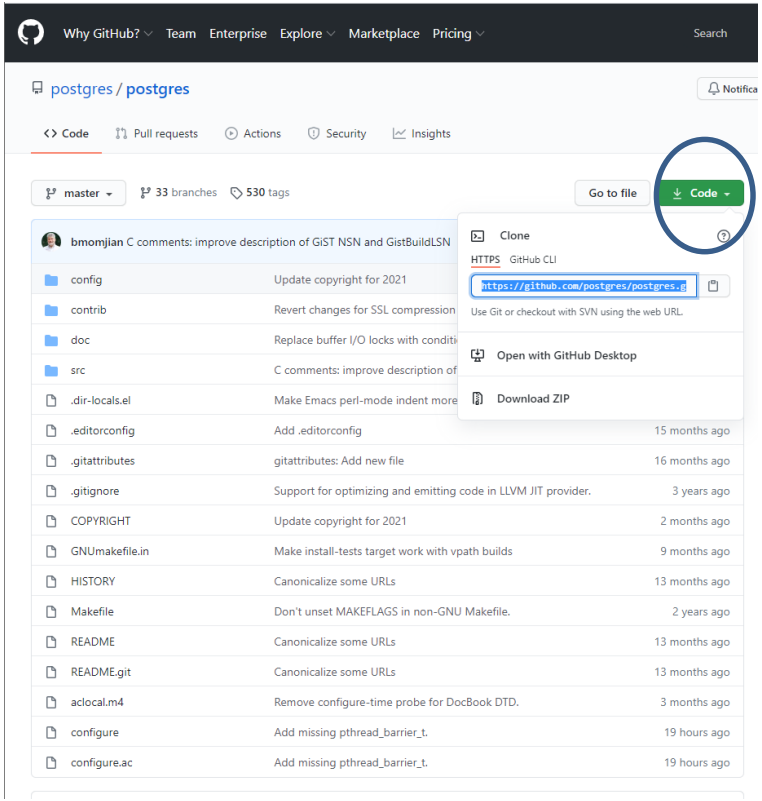
PostgreSQL 编译源码

下载官方PostgreSQL源代码

- 在开始使用PG之前，必须要先安装它
- 我们会再Ubuntu Linux环境下，下载PG源代码，然后编译
- 我们会使用源代码管理工具‘git’去做下载（clone）的动作
- 首先，先到PostgreSQL官方的github页面找到git URL：（https://github.com/postgres/postgres）
- 在Ubuntu Linux上，打开一个会话窗口，执行：

```
git clone https://github.com/postgres/postgres.git
```

- Git 应该会开始下载源代码



使用PostgreSQL

启动PostgreSQL服务以及使用psql客户端

- 当PostgreSQL集群被创建好了，配置文件也准备好了以后，就可以试着启动PostgreSQL服务了
- 可以利用‘pg_ctl’工具来启动或停止PG

```
pg_ctl  -D $name start

pg_ctl  -D $name stop
```

- 可以利用‘psql’工具来访问PG服务

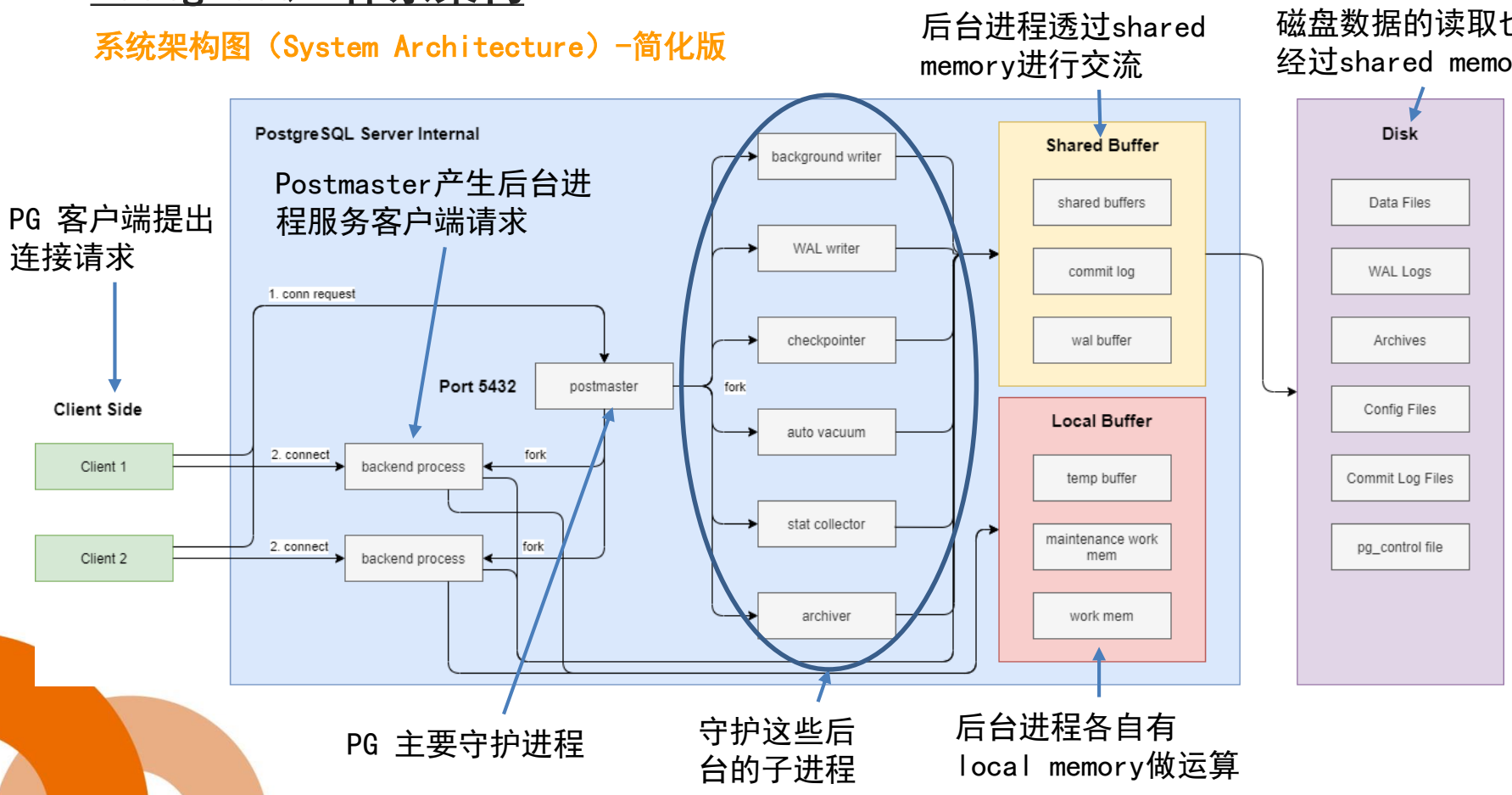
```
psql  -d postgres
```

```
postgres=# create table test (a int, b int);
CREATE TABLE
postgres=# create table test2 (a int, b char(20));
CREATE TABLE
postgres=# create table test3 (a int primary key, b int, c text);
CREATE TABLE
postgres=# \d
       List of relations
 Schema | Name  | Type  | Owner
--------+-------+-------+-------
 public | test  | table | caryh
 public | test2 | table | caryh
 public | test3 | table | caryh
(3 rows)

postgres=# insert into test values ( 55, 55);
INSERT 0 1
postgres=# select * from test;
 a  | b
----+----
 55 | 55
(1 row)

postgres=#
```

```
caryh@HGPC01:~/highgo/git/postgres.community2/postgres$ highgo/bin/pg_ctl -D mydatabase/ start
waiting for server to start....2021-03-11 13:46:56.053 PST [989299] LOG:  starting PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 7.4.0-1
ubuntu1~18.04.1) 7.4.0, 64-bit
2021-03-11 13:46:56.054 PST [989299] LOG:  listening on IPv4 address "127.0.0.1", port 5431
2021-03-11 13:46:56.060 PST [989299] LOG:  listening on Unix socket "/tmp/.s.PGSQL.5431"
2021-03-11 13:46:56.081 PST [989300] LOG:  database system was shut down at 2021-03-11 12:34:24 PST
2021-03-11 13:46:56.086 PST [989299] LOG:  database system is ready to accept connections
 done
server started
```

PostgreSQL 体系架构

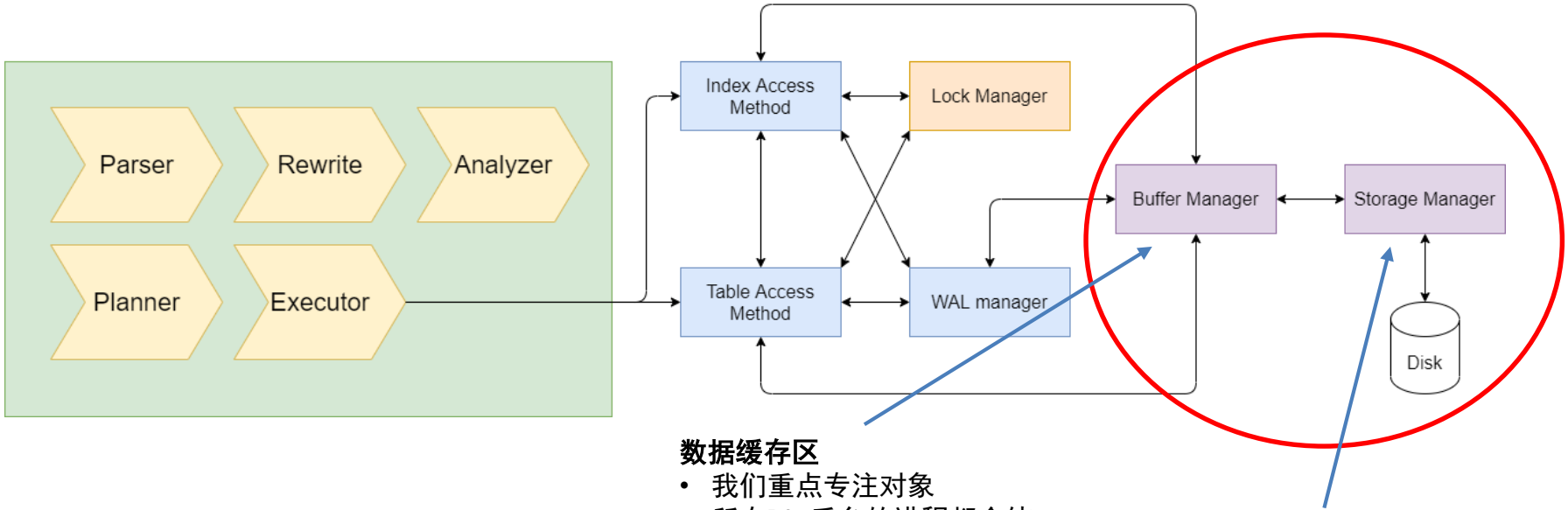
系统架构图（System Architecture）-简化版



使用PostgreSQL

语句处理主要流程

透明数据加密影响到的主要模块！



数据缓存区

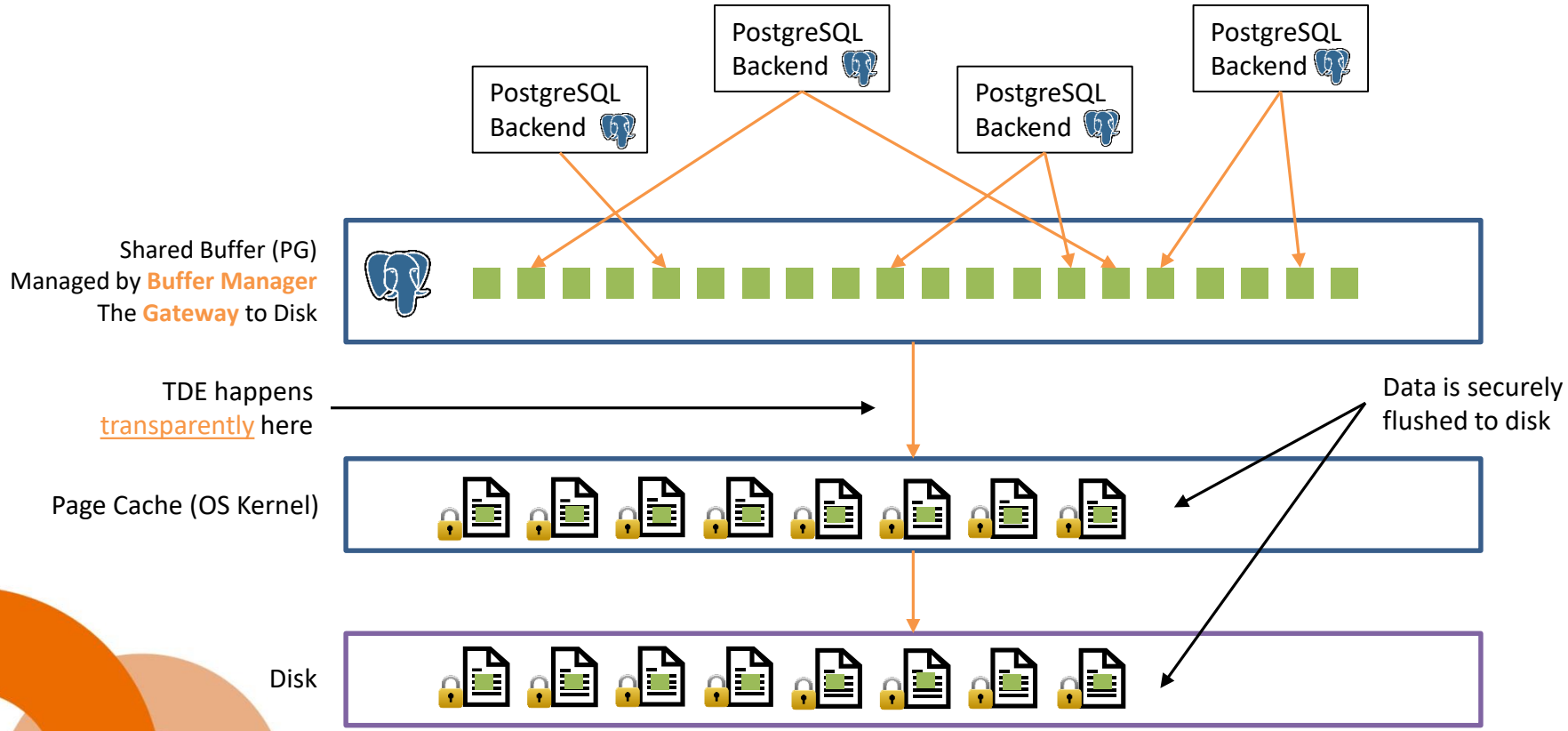
- 我们重点专注对象
- 所有PG 后台的进程都会使用到Buffer Manager的服务
- PG使用了buffer快的概念，一块默认大小8k

存储管理员

- 我们重点专注对象
- 缓冲区里的数据快最终都会被推到磁盘上了
- 目前存储的数据都是明文的

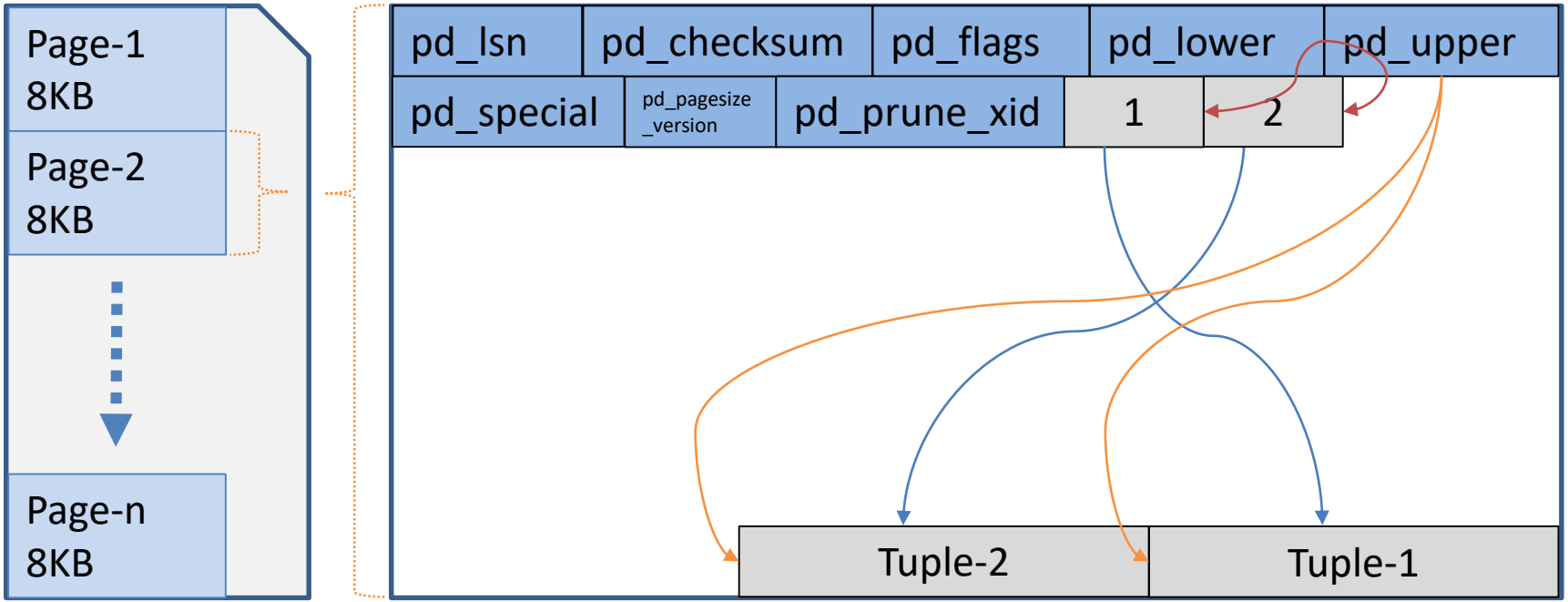
File Level Encryption

Adding TDE into the I/O Picture



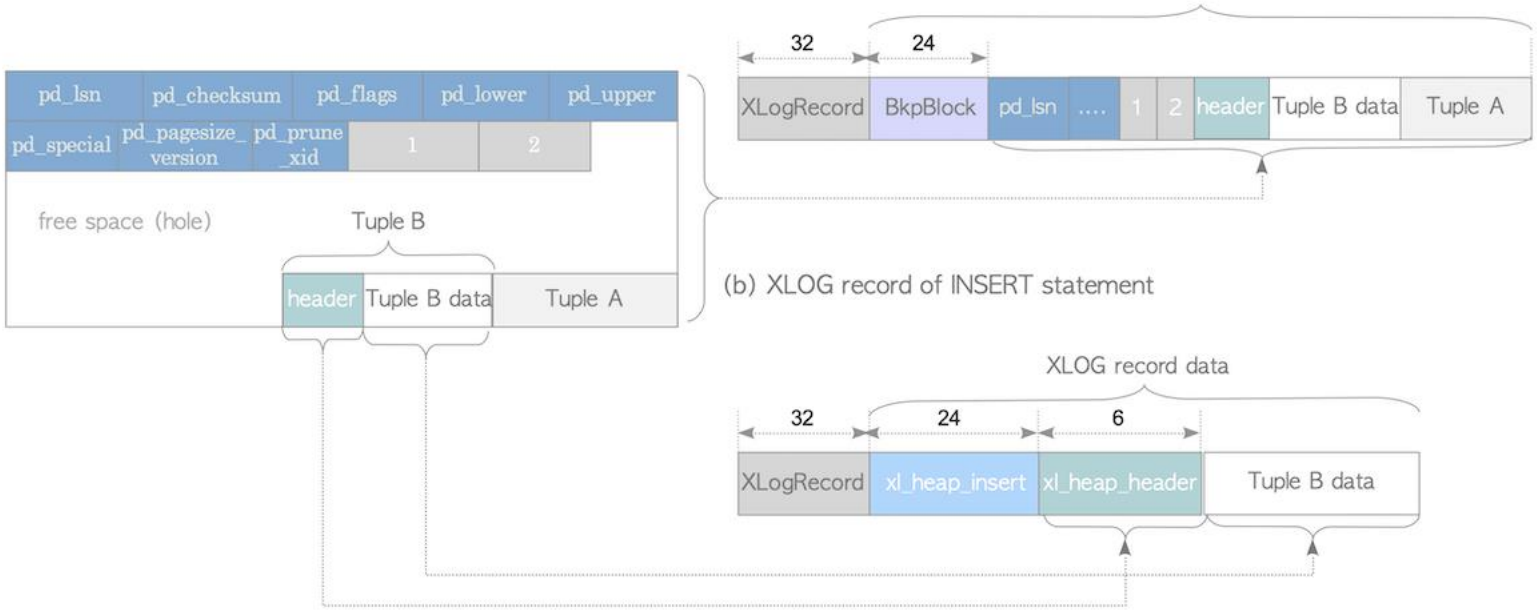
A table page layout

Insert a tuple into a page



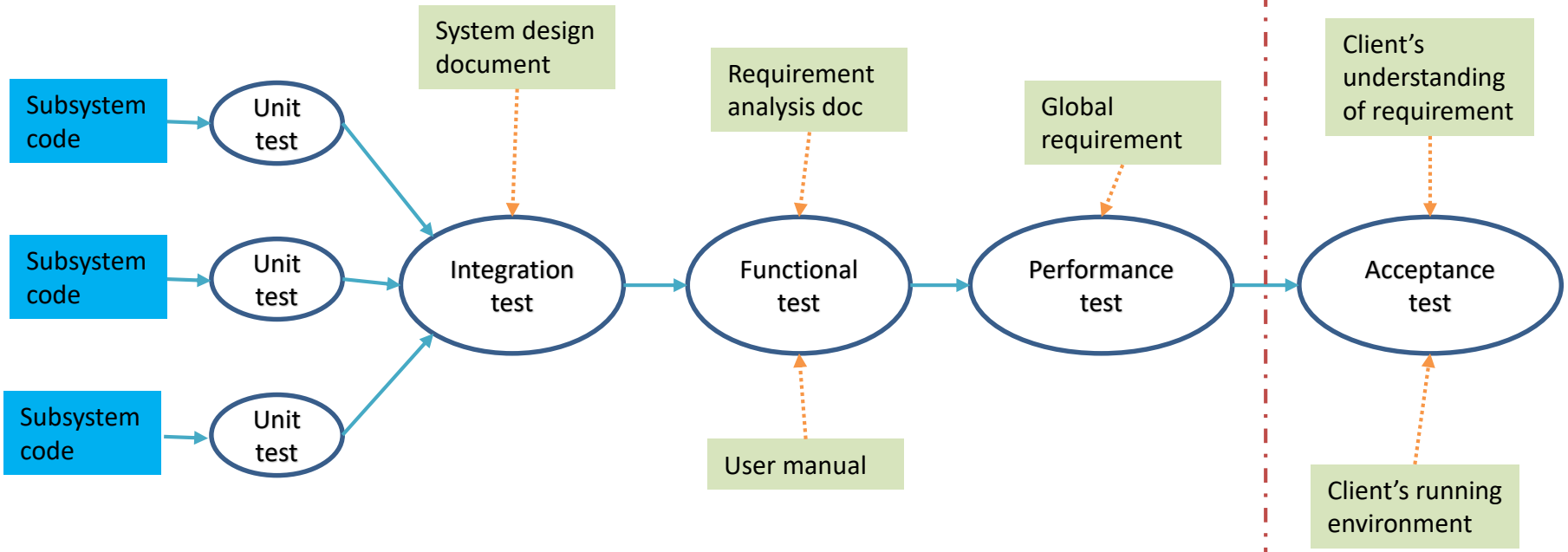
Write Ahead Log

The Layout of XLOG Record



Software Testing

Testing Activities



What Does KMS Do?

Managing the Life Cycle of a Key

Birth → Death

Key Generation

- The creation of a crypto key
- Normally created by a **pseudorandom number generator**
- We want high "quality" of randomness

Key Storage

- How to secure the generated keys before storing on disk.
- Similar idea as TDE
- Normally performs a **wrap** before storing and a **unwrap** after reading from the disk

Key Renewal

- Concerns about the **validity** of the key before it needs a renewal
- If key is **expired**, it cannot be used unless renewed

Key Rotation

- **Regenerate** a new key

Key Revocation

- **Invalidate** a key
- Make a key unusable right away

Key Destruction

- **Remove** the key such that it no longer exists

GDB 调试工具

什么是gdb?

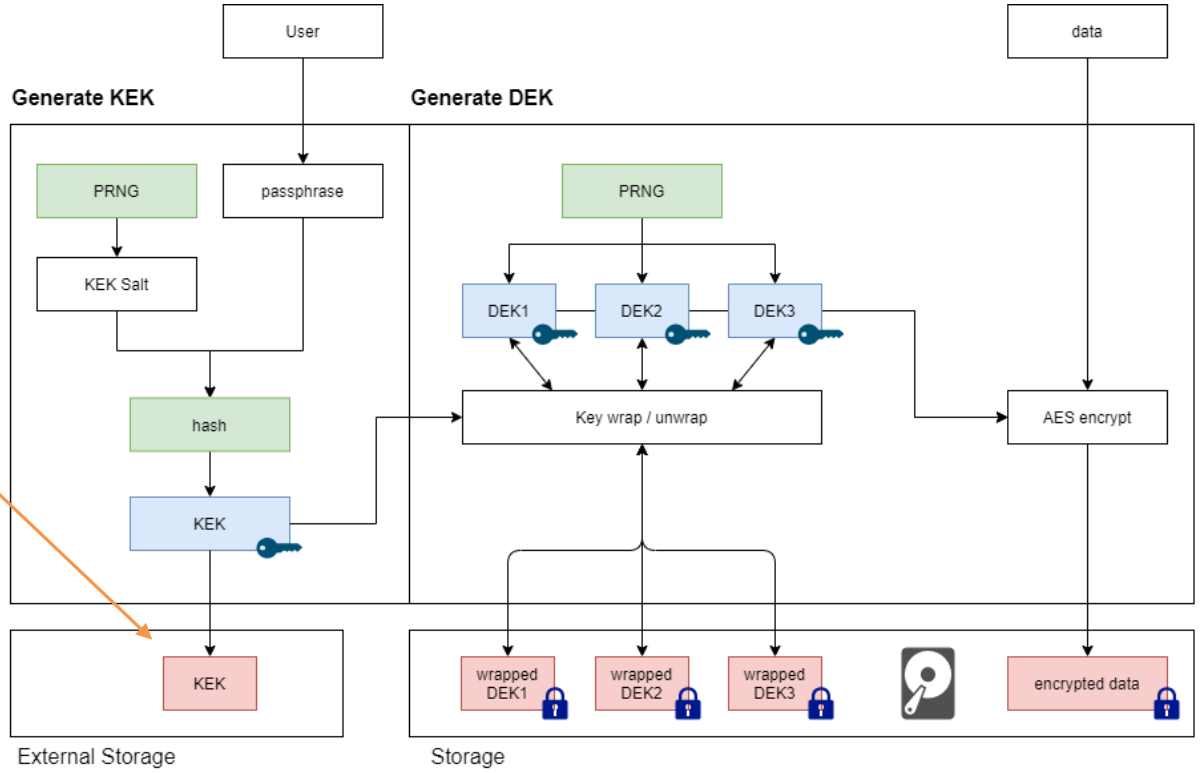
- GDB 全称“GNU symbolic debugger”
- 发展至今，GDB 已经迭代了诸多个版本，实际场景中，GDB 更常用来调试 C 和 C++ 程序。
- Windows 操作系统中，人们更习惯使用一些已经集成好的开发环境（IDE），如 VS，VC，Dev-C++，eclipse 等，它们的内部已经嵌套了相应的调试器。

GDB 的吉祥物：弓箭鱼

Key Storage

2-tier Key Management

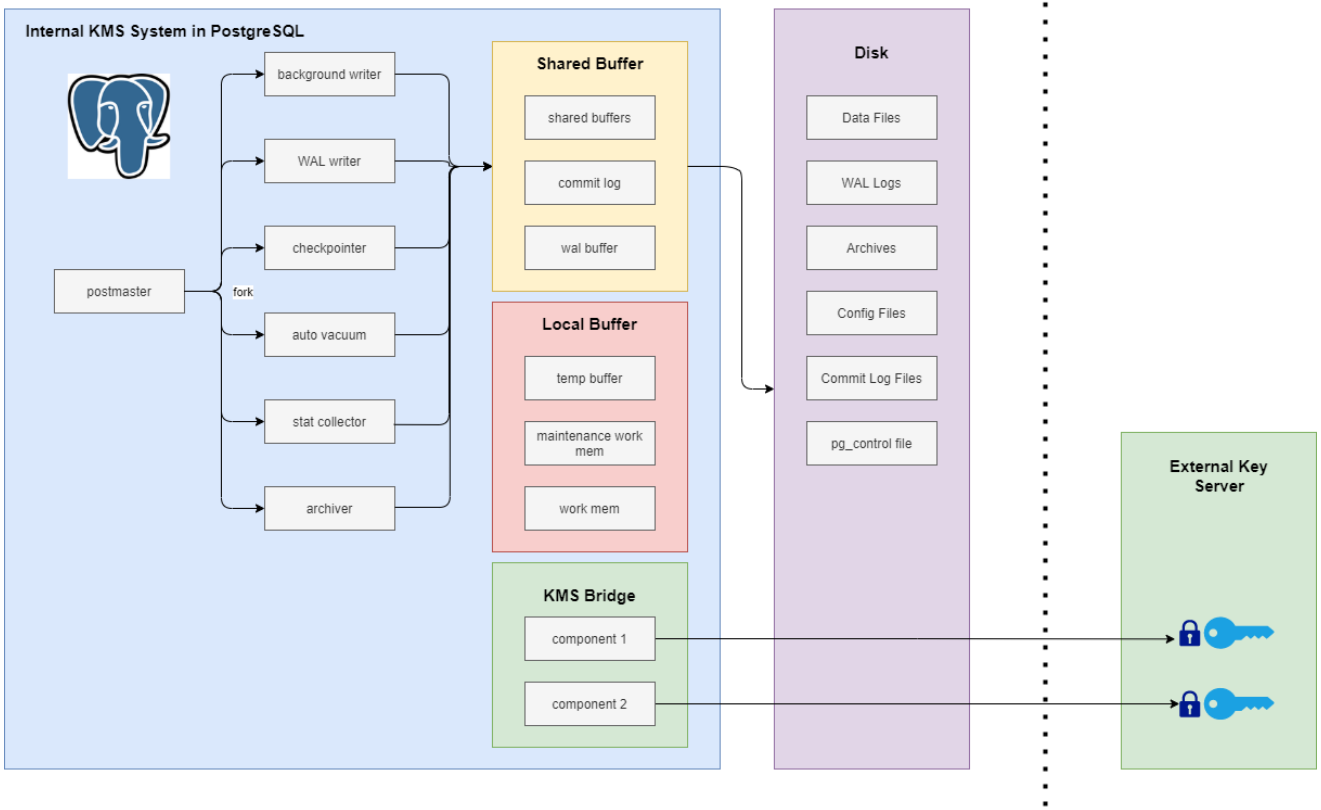
Saving KEK? Is it a Good Idea?



External KMS vs Internal KMS

External KMS

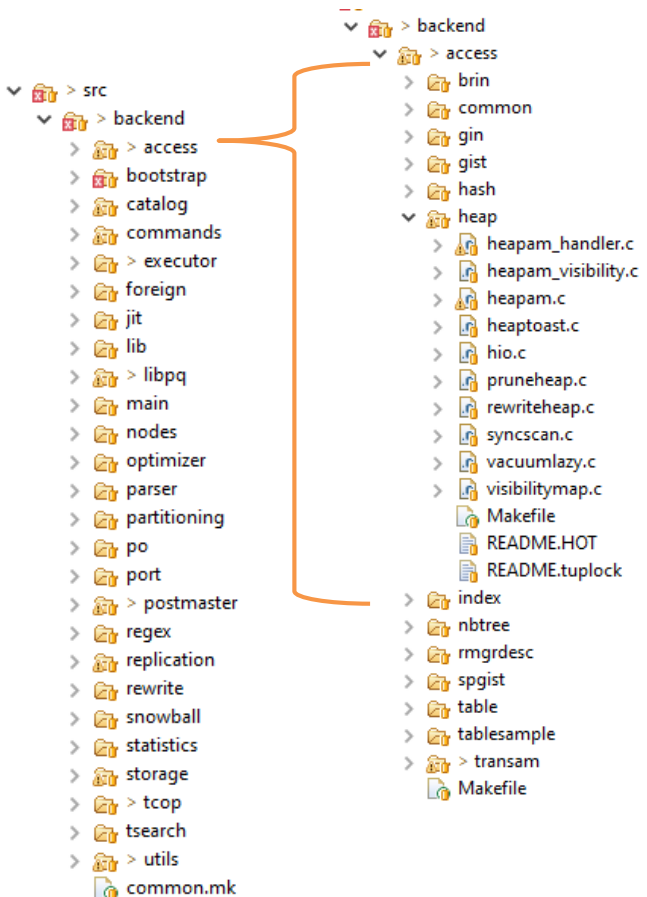
- External KMS means the KMS is built outside of the application
- The keys are stored in the separately from the application
- Normally there is a KMS Bridge component build in the application to communicate with external KMS



Adding A New Module

How To Add A New Module

- Most likely, the KMS module belongs to the backend of the PostgreSQL system, so it will reside in the “backend” folder.
- But there many sub-folders under the backend folder and each sub-folder contains more folders.
- Where should you place the new module?
- There is no right or wrong answers.
- Put your new module in a place where you feel makes the most sense
- Once you decided on the place, copy the Makefile from another component that is in the same level as yours.
- Change the names to your module.



Bruce Momjian Speech

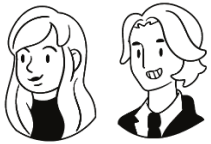
PostgreSQL Core team member, EDB VP and PostgreSQL Evangelist

- Co-founder and core team member of the PostgreSQL Global Development Group
- 我们很荣幸请到Bruce 来给北大讲讲当前国际 PostgreSQL 社区生态。
- 以及分享社区当前 TDE 和 KMS 的开发状态
- 让我们受益良多

Impacts Of TDE

TDE is Great! But…

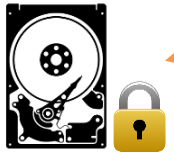
Request data via psql:
This is OK because PG will decrypt the data for you



You

PostgreSQL

TDE



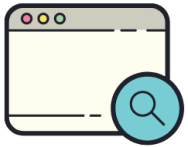
Encrypted Buffer Data

We want to **prevent** hacker from stealing our data

hacker

Encrypted WAL Data

We **want** this application to **access** our data, but it can't because it is encrypted!
… therefore TDE broke this app!



Application / Tools

Regression Framework

Using PG Regression Framework

- The regression tests are a comprehensive set of tests for the SQL implementation in PostgreSQL.
- They test standard SQL operations as well as the extended capabilities of PostgreSQL.
- So… whatever you are doing with PostgreSQL, you need to pass all of these regression tests to be “SQL” compliant.
- The framework is located in src/test/regress subfolder.

```
caryh@HGPC01:~/highgo/git/postgres.community2/postgres/src/test/regress$ ls -ltr
total 916
-rw-rw-r-- 1 caryh caryh    159 Jan  8 10:08 README
-rw-rw-r-- 1 caryh caryh    778 Jan  8 10:08 Makefile
drwxrwxr-x 2 caryh caryh   4096 Jan  8 10:08 data
-rw-rw-r-- 1 caryh caryh    165 Jan  8 10:08 resultmap
-rw-rw-r-- 1 caryh caryh    579 Jan  8 10:08 standby_schedule
-rw-rw-r-- 1 caryh caryh   5421 Apr 22 15:14 GNUmakefile
-rw-rw-r-- 1 caryh caryh   4569 Apr 22 15:14 parallel_schedule
drwxrwxr-x 2 caryh caryh   4096 Apr 22 15:14 output
drwxrwxr-x 2 caryh caryh   4096 Apr 22 15:14 input
drwxrwxr-x 2 caryh caryh  12288 Apr 22 15:14 expected
-rw-rw-r-- 1 caryh caryh   3226 Apr 22 15:14 serial_schedule
-rwxrwxr-x 1 caryh caryh   4438 Apr 22 15:14 regressplans.sh
-rw-rw-r-- 1 caryh caryh  27497 Apr 22 15:14 regress.c
-rw-rw-r-- 1 caryh caryh   3259 Apr 22 15:14 pg_regress_main.c
-rw-rw-r-- 1 caryh caryh   1458 Apr 22 15:14 pg_regress.h
-rw-rw-r-- 1 caryh caryh  66588 Apr 22 15:14 pg_regress.c
drwxrwxr-x 2 caryh caryh  12288 Apr 22 15:14 sql
-rw-rw-r-- 1 caryh caryh 208280 May 18 09:59 regress.o
-rwxrwxr-x 1 caryh caryh 127728 May 18 09:59 regress.so
-rw-rw-r-- 1 caryh caryh 100280 May 18 09:59 pg_regress.o
-rw-rw-r-- 1 caryh caryh  14536 May 18 09:59 pg_regress_main.o
-rwxrwxr-x 1 caryh caryh 177888 May 18 09:59 pg_regress
-rwxrwxr-x 1 caryh caryh  57872 May 18 09:59 refint.so
-rwxrwxr-x 1 caryh caryh  45984 May 18 09:59 autoinc.so
```

What is PG Extension

PG Extension - A modular design

- Postgres is a huge database system consisting of a wide range of data types, functions, features and operators that can be utilized to solve many common to complex problems.
- However, in the world full of complex problems, sometimes these are just not enough depending on the use case complexities.
- Worry not, since Postgres version 9, it is possible to extend Postgres' s existing functionalities with the use of "extensions"

PostgreSQL uses "modular" design, like modular pipe homes in Hong Kong

https://www.scmp.com/lifestyle/interiors-living/article/2121891/how-hong-kongs-low-cost-housing-pipe-dream-became

2 Ways To Make an Extension

Know what you have first

- An extension can be created using…
  - C Programming language
    - Basically to create one or more additional SQL functions that PG can use to perform certain tasks
  - PL/pgSQL procedural language
    - Basically a collection of SQL instructions to complete certain tasks

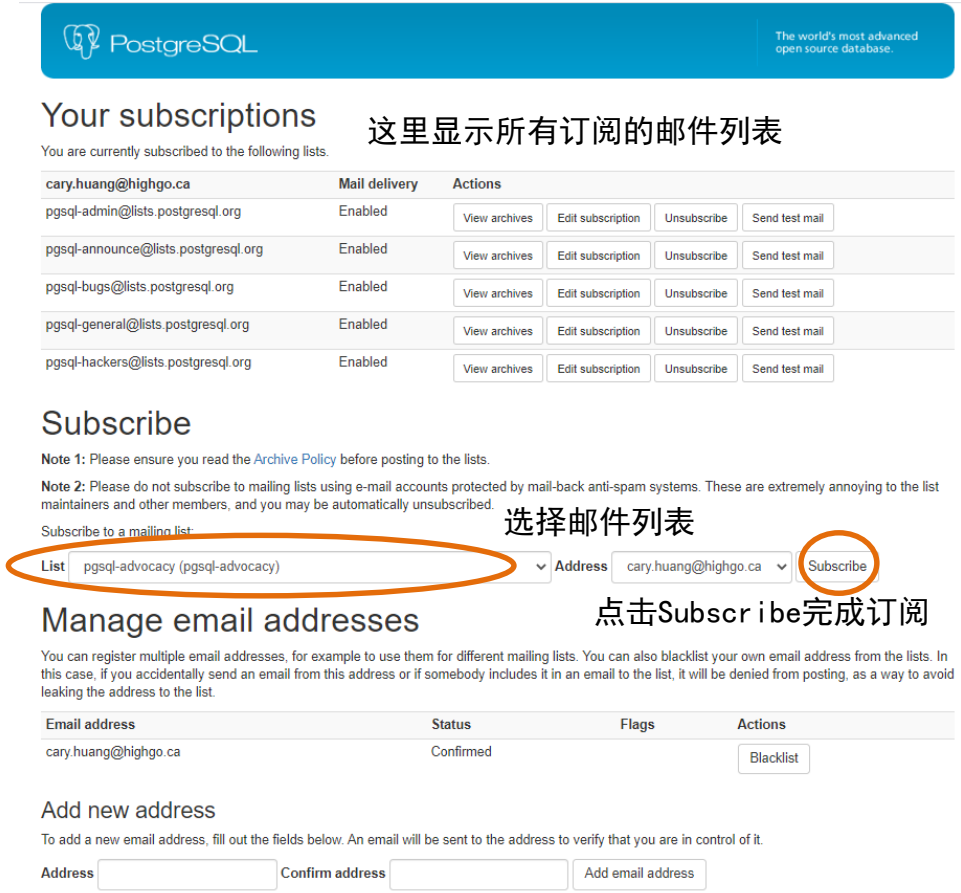
```c
/*
 * Function returning data from the shared buffer cache - buffer number,
 * relation node/tablespace/database/blocknum and dirty indicator.
 */
PG_FUNCTION_INFO_V1(pg_buffercache_pages);

Datum
pg_buffercache_pages(PG_FUNCTION_ARGS)
{
    FuncCallContext *funcctx;
    Datum       result;
    MemoryContext oldcontext;
    BufferCachePagesContext *fctx;  /* User function context. */
    TupleDesc   tupledesc;
    TupleDesc   expected_tupledesc;
    HeapTuple   tuple;

    if (SRF_IS_FIRSTCALL())
    {
        int         i;

        funcctx = SRF_FIRSTCALL_INIT();
```

```sql
CREATE FUNCTION char_count(TEXT, CHAR)
RETURNS INTEGER
LANGUAGE plpgsql IMMUTABLE STRICT
  AS $$
    DECLARE
      charCount INTEGER := 0;
      i INTEGER := 0;
      inputText TEXT := $1;
      targetChar CHAR := $2;
    BEGIN
    WHILE i <= length(inputText) LOOP
      IF substring( inputText from i for 1) = targetChar THEN
        charCount := charCount + 1;
      END IF;
        i := i + 1;
      END LOOP;

    RETURN(charCount);
    END;
  $$;
```

PG电子邮件列表订阅

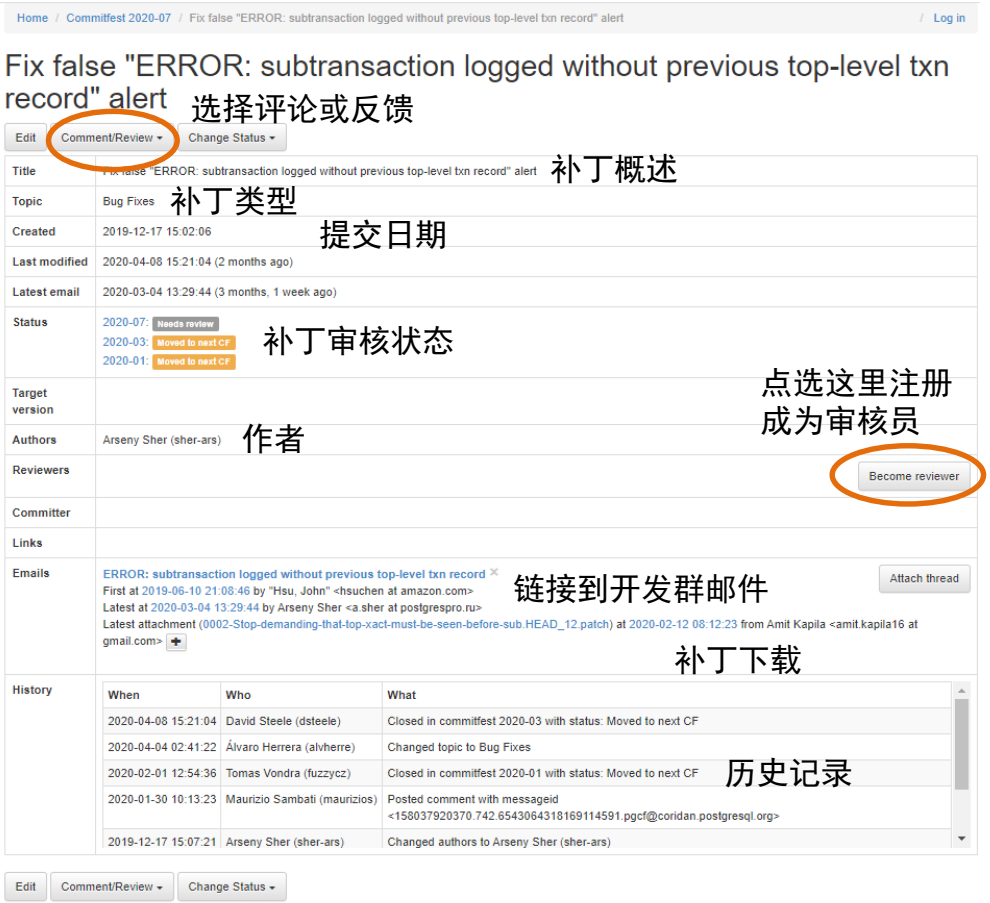
- 有了PG账户，第一件事情就是先透过这个链接 https://lists.postgresql.org/manage/ 订阅PG的邮件列表。
- PG社区提供了很多不同类别的邮件列表平台给PG爱好者参与讨论。
- 我建议至少订阅以下2个电子邮件列表，以接收最新的技术讨论，新闻和错误报告：
- PG开发群列表 pgsql-hackers 是最活跃的列表之一，也是这次培训最常用来与社区交流的沟通渠道。

错误提交列表 pgsql-bugs@lists.postgresql.org
PG开发群列表 pgsql-hackers@lists.postgresql.org



利用commitfest 平台做补丁审查

- PostgreSQL官方的补丁审查都是在commitfest网站（https://commitfest.postgresql.org/）维护的。这其实是一个PG开发群邮件列表(pgsql-hackers)邮件的集合。
- Commitfest 上的每一条记录其实就是一个链接到某一个含有补丁附件的开发群邮件线程
- Commitfest 上的每一条记录还带有状态信息，显示某补丁进入官方PostgreSQL源代码的状态。
- 在这里，我们可以看到其他贡献者提交的补丁以及过去的讨论并参与审核

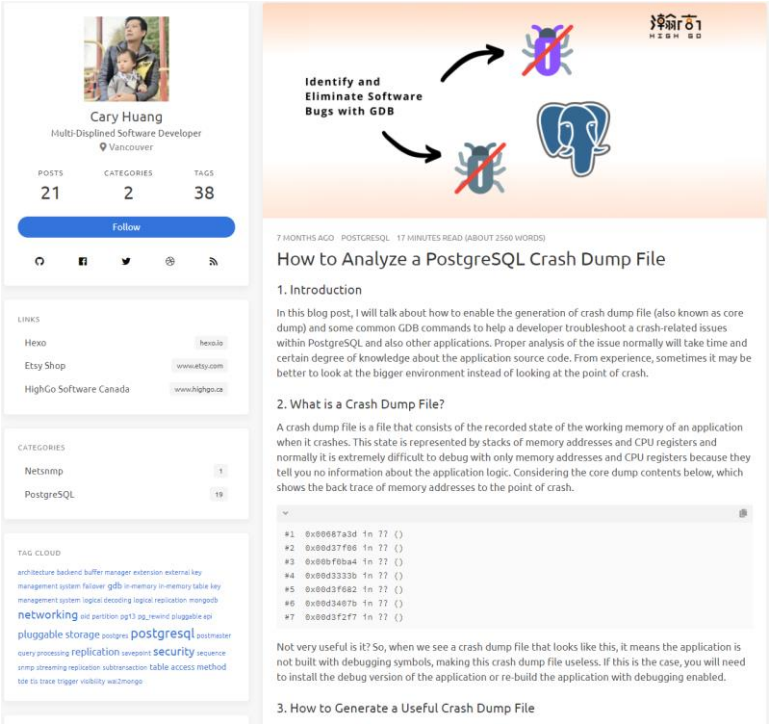


补丁生成及提交 - Cont.

- 准备好错误补丁了以后，建议找另一为软件背景的人员帮你审核该补丁，以获得一些反馈。David和Cary可以帮你们做审核。
- 改进以后，您就可以撰写一个新的电子邮件，其中包含您的补丁的说明。例如：
  - 它修复了什么？
  - 如何验证此修复？
  - 修复是基于什么概念或是测试？
  - 为什么需要这个补丁？
  - 这个补丁会不会影响PostgreSQL的正常运作？
- 邮件准备好了以后，就可以把它发送到PG开发群（pgsql-hackers@lists.postgresql.org）来开启一个新的社区讨论。
- 参与社区讨论通常是以英文来做交流

博客写什么？

- 你可以写关于PG的任何事情，包含PG操作教程，工作原理，你目前PG的工作心得。
- 或是你可以把博客当成你的笔记，学到了什么新知识就可以用博客的形式把它记录下来（像是gdb，加密，或密钥管理理论基础等等。。。）
- 可以不需要把一个博客写的很专业或死板，你可以用第一人称视角和你平时说话的语气去描述一个事情
- 利用博客网站慢慢建立你的个人声誉



关于PostgreSQL

PostgreSQL 无处不在

国内也越来越多知名企业在应用PostgreSQL:

- 工商银行
- 中国邮政储蓄
- 平安集团
- 苏宁
- 京东
- 去哪儿网
- 高德地图
- …等等

基于PostgreSQL相关的数据库产品也不断地在出现:

- 腾讯Tbase
- 阿里PolarDB
- 瀚高HighgoDB
- 亚信AntDB
- 华为OpenGauss
- 金仓KingBase
- 美创MCDB
- …等等

关于PostgreSQL

培训认证，政策导向

- 自从2019年“中国PostgreSQL培训认证”获得工信部下属中国软件行业协会及中国电子工业标准化技术协会权威认证。

- 多家培训机构也纷纷出现成为授权合作伙伴，例如：
  - 恩墨学院
  - 晟数学院
  - CUUG优技
  - 东方瑞通
  - 博森瑞
  - 柯普瑞等

- 在当前的国际竞争格局下，不能掌握核心技术就会受制于人，这对于所有民族基础软件企业、技术从业者都提出了更高要求。

- 在新时期信息安全新形势下，国内大量企业投身于利用以PG为代表的开源技术创新ICT产业的事业中。

- 并在国内造就大量的PG技术人才需求

实践课就到此结束了…
希望你们有所收获

祝你们成功！

David Zhang

联系方式：david.zhang@highgo.ca

Cary Huang

联系方式：cary.huang@highgo.ca

HIGH GO

融知与行 瀚且高远

THANKS