Custom background workers

PostgreSQL 9.3

Michael Paquier

2013/02/16
Tokyo, #pgunconf

About the lecturer

- Michael Paquier
- Engineer at VMware
- Working on PostgreSQL stuff
- PostgreSQL hacker
- Contact
  - Twitter: @michaelpq
  - michael@otacoo.com
  - Blog: michael.otacoo.com

Bgworker?

- Introduction in PostgreSQL 9.3
- Child process of postmaster
- Possible to run user-defined code in shared library loaded by PG core
- Set of APIs for process-related plug-ins
  - Customizable
  - Extensible
  - Adaptable

Features

- Several options
  - Access to databases
  - Access to shared memory
  - Serial transactions
  - Start/stop/restart control
- User-defined parameters

Ideas

- Statistics-related
  - Save pg_stat_statement module information to tier-table with certain timing (Ex: 1 TX/s)
  - Export pg_stat_* stuff automatically
- Maintenance
  - Reindex automatically invalid indexes
  - Kill inactive connections after certain duration (pg_stat_activity + pg_terminate_backend)
- Example: “Hello World”
  - Registered as bgworker: hello world
  - Have a look at pg_log!

Hello World (1)

- Headers

```
/* Minimum set of headers */
#include "postgres.h"
#include "postmaster/bgworker.h"
#include "storage/ipc.h"
#include "storage/latch.h"
#include "storage/proc.h"
#include "fmgr.h"

/* Essential for shared libs! */
PG_MODULE_MAGIC;

/* Entry point of library loading */
void _PG_init(void);

/* Signal handling */
static bool got_sigterm = false;
```

Hello World (2)

- Initialization with _PG_init()

```
void
_PG_init(void)
{
    BackgroundWorker    worker;

    /* Register the worker processes */
    worker.bgw_flags = BGWORKER_SHMEM_ACCESS;
    worker.bgw_start_time = BgWorkerStart_RecoveryFinished;
    worker.bgw_main = hello_main;
    worker.bgw_sighup = NULL;
    worker.bgw_sigterm = hello_sigterm;
    worker.bgw_name = "hello world";
    worker.bgw_restart_time = BGW_NEVER_RESTART;
    worker.bgw_main_arg = NULL;
    RegisterBackgroundWorker(&worker);
}
```

Hello World (3)

- Main loop

```
static void
hello_main(void *main_arg)
{
    /* We're now ready to receive signals */
    BackgroundWorkerUnblockSignals();
    while (!got_sigterm)
    {
        int     rc;
       /* Wait 10s */
        rc = WaitLatch(&MyProc->procLatch,
                       WL_LATCH_SET | WL_TIMEOUT | WL_POSTMASTER_DEATH,
                       10000L);
        ResetLatch(&MyProc->procLatch);
        elog(LOG, "Hello World!");       /* Say Hello to the world */
    }
    proc_exit(0);
}
```

Hello World (4)

- SIGTERM handler

```
static void
hello_sigterm(SIGNAL_ARGS)
{
    int         save_errno = errno;
    got_sigterm = true;
    if (MyProc)
        SetLatch(&MyProc->procLatch);
    errno = save_errno;
}
```

- Makefile

```
MODULES = hello_world
PG_CONFIG = pg_config
PGXS := $(shell $(PG_CONFIG) --pgxs)
include $(PGXS)
```

Things to remember

- Set shared_preload_libraries in postgresql.conf
- Load with _PG_init
- Unblock signals
- WaitLatch for interruptible sleep
- BGWORKER_SHMEM_ACCESS | BGWORKER_BACKEND_DATABASE_CONNECTION

Additional tips

- Security !
- Have a look at contrib/worker_spi
- Documentation
  - www.postgresql.org/docs/devel/static/bgworker.html

Thanks!
Questions?

2013/02/16
Tokyo, #pgunconf