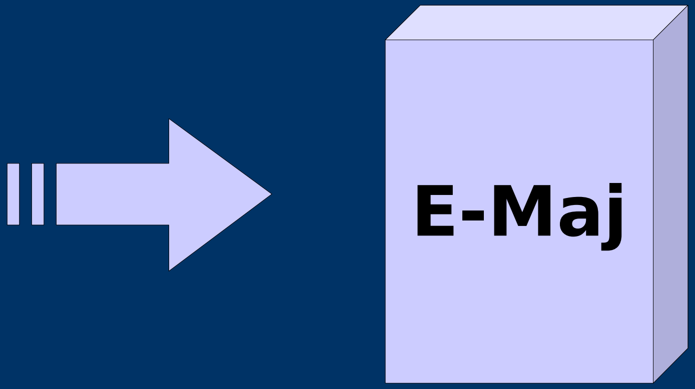
E-Maj 0.7:
a PostgreSQL contrib

Ph.Beaudoin – June 2010

From the idea of logical restore to … E-Maj

- Original idea = table_log contrib from Andreas Scherbaum
  - 1 trigger per table to log all updates into a log table
  - 1 function to cancel the updates
- Development of plpgsql functions extending the concept to build a solution usable on production



French acronym for
« Enregistrement des Mises A Jour »,
i.e. Updates recording

Requirements

- Reliability:
  - Absolute integrity of databases after « rollbacks »
  - Manage all objects (tables, séquences, contraintes,...)
- Ease of use for DBAs and production people:
  - Easy to understand and use
  - Easy to automatize (« scriptable »)
- Performance:
  - Limited overhead of the log
  - Acceptable « rollback » duration
- Maintenability
- Security

Concepts

- Table_group = a set of tables and/or sequences belonging to a unique schema or several schemas and having the same life cycle ; it's the object on which « marks » and « rollbacks » are applied ; it's the only object manipulated by users
- Mark = stable point in the life of a table_group, whose state can be set back ; is identified by a name
- Rollback = positionning of a table_group at its state when a mark was previously set

Installation

- Preliminary operations:
  - plpgsql language has to be created in the database
  - a tablespace, named tspemaj, must have been created in the cluster
- Installation done with a unique script, named emaj.sql ; to be launched using a super-user ROLE
- The installation in a database adds :
  - 1 schema (emaj) containing
  - 30 plpgsql functions and
  - 8 technical tables and 2 types

Initialisation

- 1) Populate emaj_group_def table to define groups and the tables/sequences they contain
- 2) For each group :
  - SELECT emaj_create_group (groupe);
  - => creates for each application table:
    - 1 trigger associated to table updates
    - 1 log table into tablespace tspemaj
    - 1 function to « rollback » the updates on the application table
  - A emaj_drop_group (groupe) function … drops a previously created group

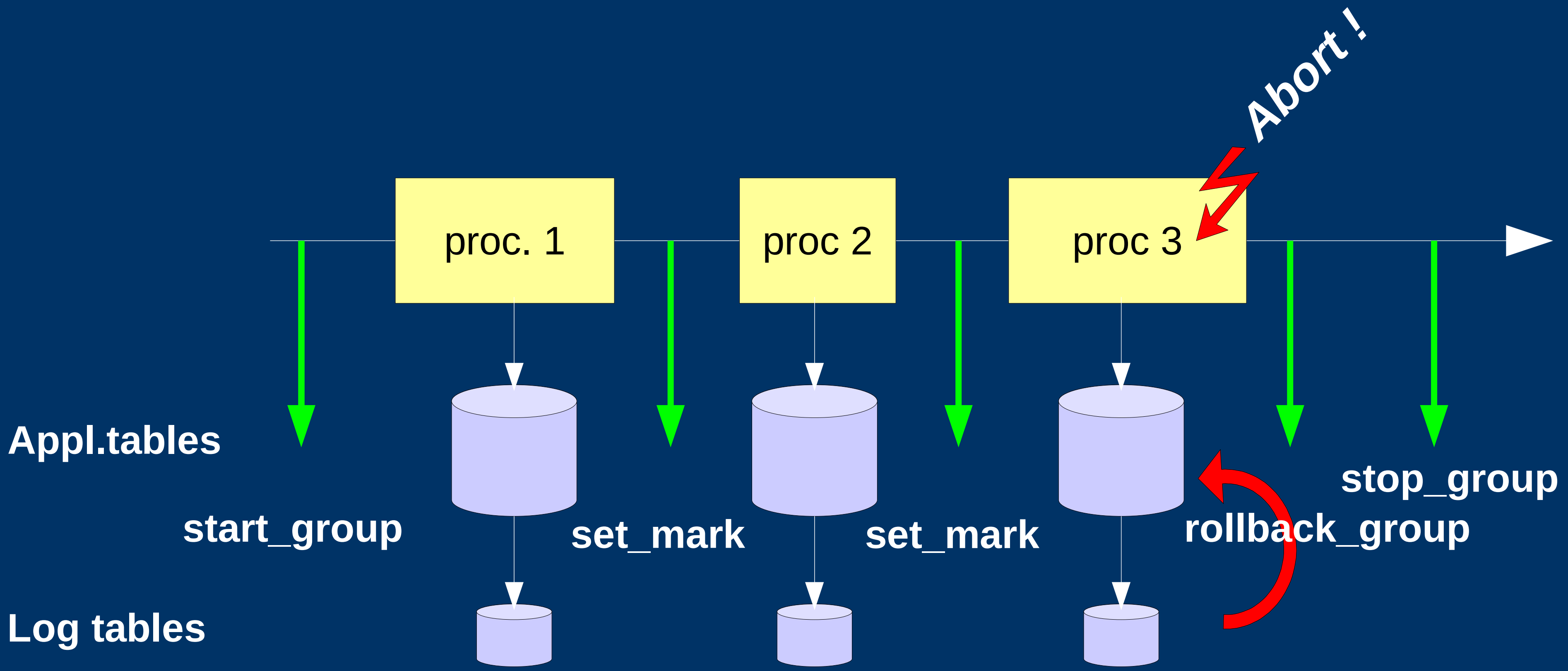
Main functions

- emaj_start_group (groupe, mark)
  - Activates log triggers and set an initial mark
- emaj_set_mark_group (groupe, mark)
  - Sets an intermediate mark
- emaj_rollback_group (groupe, mark)
  - Rollbacks tables et sequences of the group to their state at mark set
- emaj_stop_group (groupe)
  - Deactivates log triggers => rollback no longer possible

Marks usage strategies

- « mono-mark » usage to minimise disk space use
  - repeat
    - start_group (group, mark)
    - processing i
    - stop_group (group)
- « multi-marks » usage for more flexibility in rollbacks
  - start_group (group, mark1)
  - repeat
    - processing i
    - emaj_set_mark (group, mark i+1)
  - stop_group (group)

A typical E-Maj sequence ...



Possible usages

- Provides a rollback capability on batch processing without being obliged to either pgdump/restore tables or physicaly save and restore the entire cluster disk space
- All the more interesting as:
  - tables are large with relatively limited updates
  - several tables groups / databases share the same cluster
- Can also help application tests in providing a way to quickly rollback updates issued by a test and repeat those tests

Statistic functions

- emaj_rlbk_stat_group (group, mark)
  - Quickly provides per table statistics about the number of rows in log tables between a mark and the current situation
  - This gives data to estimate the duration of a rollback to a given mark
- emaj_log_stat_group (group, begin_mark, end_mark)
  - Delivers statistics on updates between 2 marks, per table et per ROLE that initiated the updates
  - Allows to check of ROLEs who initiated updates and estimate the duration of a potential rollback

Other secondary functions

- emaj_rollback_and_stop_group (group, mark)
  - Chains the calls to rollback_group et stop_group functions - this allow to differ the rows deletion from log tables in order to get quicker rollback
- emaj_delete_mark_group (group, mark)
  - Suppess an intermediate mark
- emaj_reset_group (group)
  - Purges log tables before the next emaj_start_group call. This is a way to reclaim disk space if needed
- emaj_verify_all ()
  - Verifies the consistency of the emaj schema, detecting orphan log tables or functions

Test functions

- emaj_snap_group (group, directory)
  - Snaps all tables and sequences of a group on individual files located on a directory
  - Rows are ordered by primay keys
  - Snap files can be diff with a reference to be sure the log and rollback operations worked properly

Parallel rollback extension

- A php module performs parallel restore
- Acts as a client of the database
- Automaticaly spreads the group to rollback into a given number of subgroups
- Performs the parallel rollback in a unique transaction (2PC)
- emajParallelRollback.php -d <database> -h <host> -p <port> -U <user> -W <password> -g <group_name> -m <mark> -s <#subgroups>
- Other options: --help, -v, --version
- Needs php with the PostgreSQL extension

Reliability

- Many checks in particular at emaj_start_group and emaj_rollback_group time
  - Do all listed tables and sequences exists ?
  - Do the triggers and log tables exist with the right columns and types ?
  - Are we sure the table stuctures have not changed between start_group and rollback_group functions
- Exclusive lock on tables at start_group and rollback_group time to be sure no transaction are currently accessing the tables
- Rollback all tables et sequences in a single transaction

Security

- E-Maj objects are created by a super-user
- No right is granted on the emaj schema and all related tables and functions
- All tables from emaj schema are only visible by super-users
- Log triggers are created as « SECURITY DEFINER »
- Protection against SQL injections

Current limits

- PostgreSQL : at least version 8.2
- Every application table belonging to a group needs a PRIMARY KEY
- Schema name length + application table name length <= 52 characters
- DDL or TRUNCATE operations cannot be managed by E-Maj.

E-Maj: to do...

- More testing
- More performance measurements
- Track TRUNCATE ? (8.4 minimum)

To conclude...

- More information in the readme.txt and releaseNotes.txt files
- Many thanks for their help to :
  - Andreas Scherbaum
  - Jean-Paul Argudo and Dalibo team