MIGRATION

- THE EXPEDITION

PGConf 2020 – 27th Feb

14:15 to 15:00

Raman
Head – Open source Migration Factory

Venkat
Technical Architect

THE FUTURE IS YOU

SOCIETE GENERALE

AGENDA

1
INTRODUCTION

2
MIGRATION EXPERIENCE

3
TOOLS

4
Q&A

SOCIETE GENERALE

INTRODUCTION

SOCIETE GENERALE

ABOUT SOCIETE GENERALE

Societe Generale is one of the leading European financial services groups.

Based on a diversified and integrated banking model, the Group combines **financial strength** and **proven expertise in innovation** with a strategy of **sustainable growth**, aiming to be the **trusted partner for its clients**, committed to the positive transformations of the world.

3 complementary core businesses:

- French Retail Banking
- International Retail Banking & Financial Services
- Global Banking & Investor Solutions

31 million*
individual clients, businesses and institutional investors

+ 149,000**
members of staff

67 countries
worldwide

€3,864 million
Group Net Income

€25,205 million
in Net Banking Income

10.9%
Common Equity Tier 1 ratio

OUR VALUES

Team Spirit	Innovation	Responsibility	Commitment

** Excluding insurance policyholders.*
*** Rounded figure. Headcount at end-2018 excluding temporary staff*

A TECHNOLOGY INTENSIVE COMPANY

IT BUDGET
~EUR 4 BILLION

23 500
IT STAFF IN THE GROUP

40% **OF IT TEAMS**
WORKING IN AGILE

EUR 650 MILLION
OF INVESTMENT DEDICATED TO SECURITY OVER 3 YEARS

1500+ **API IN PRODUCTION**

60% **OF ELIGIBLE SERVERS**
IN CLOUD

1 BILLION
CONNECTIONS/YEAR

70 MILLIONS
NOTIFICATIONS
IN REAL TIME

4+ MILLIONS
APP DOWNLOADS

500 000
CONTRACTS SIGNED ELECTRONICALLY

40 PETABYTES
OF DATA

SOCIETE GENERALE

Financial figures as of 31.12.2018 - Some figures have been ro *own*

OUR OPENSOURCE STRATEGY : 3 PILLARS

"Think Opensource first"

USE

- OPEN SOURCE FIRST
- ALTERNATIVE - EVERY IT ASSET
- SUPPORT STRATEGY

CONTRIBUTE

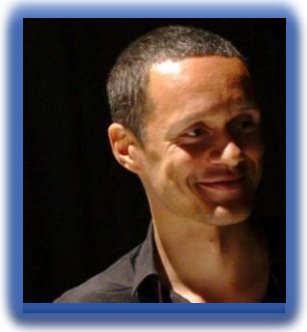
- SHARING & COLLABORATING
- CONTRIBUTION FRAMEWORK
- PRODUCT ROADMAP

ATTRACT

- PARTNERSHIP - PRODUCT/EVENT
- EXPERT CAREER PATH
- COMMUNICATE SG AMBITION

SOCIETE GENERALE

OUR INHOUSE PG EXPERTS & DEVELOPERS



Anthony NOWOCIEN

PostgreSQL Expert and a Developer (**maintainer code2pg**)
Expertise in DB, Development,
An avid traveler and Presenter,
Secretary of PostgreSQL FR.

Venkat SUSARLA

Technical and Cloud Solution Architect
Expertise in DB, Scripting and Cloud
An AI Enthusiast and an avid reader

Karthik KUMAR

ora2pg & code2pg developer

Kiran PERICHARLA

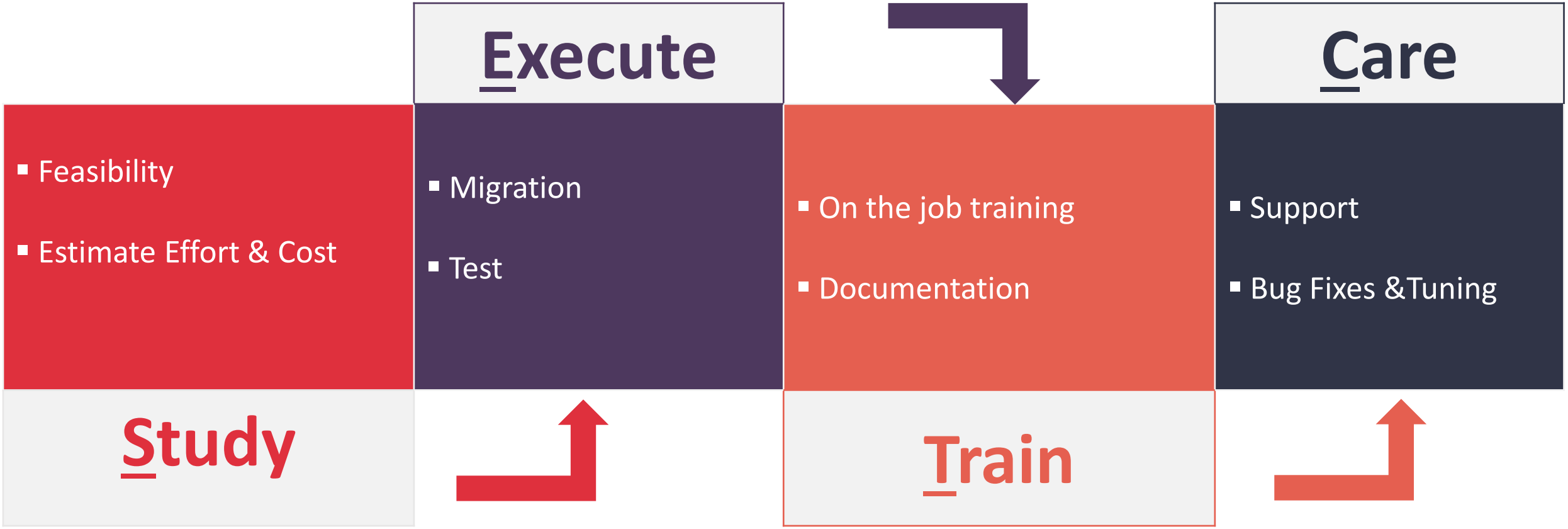
ora2pg & code2pg developer

SOCIETE GENERALE

MIGRATION PROCESS & STATS

SOCIETE GENERALE

MIGRATION PROCESS – SETC

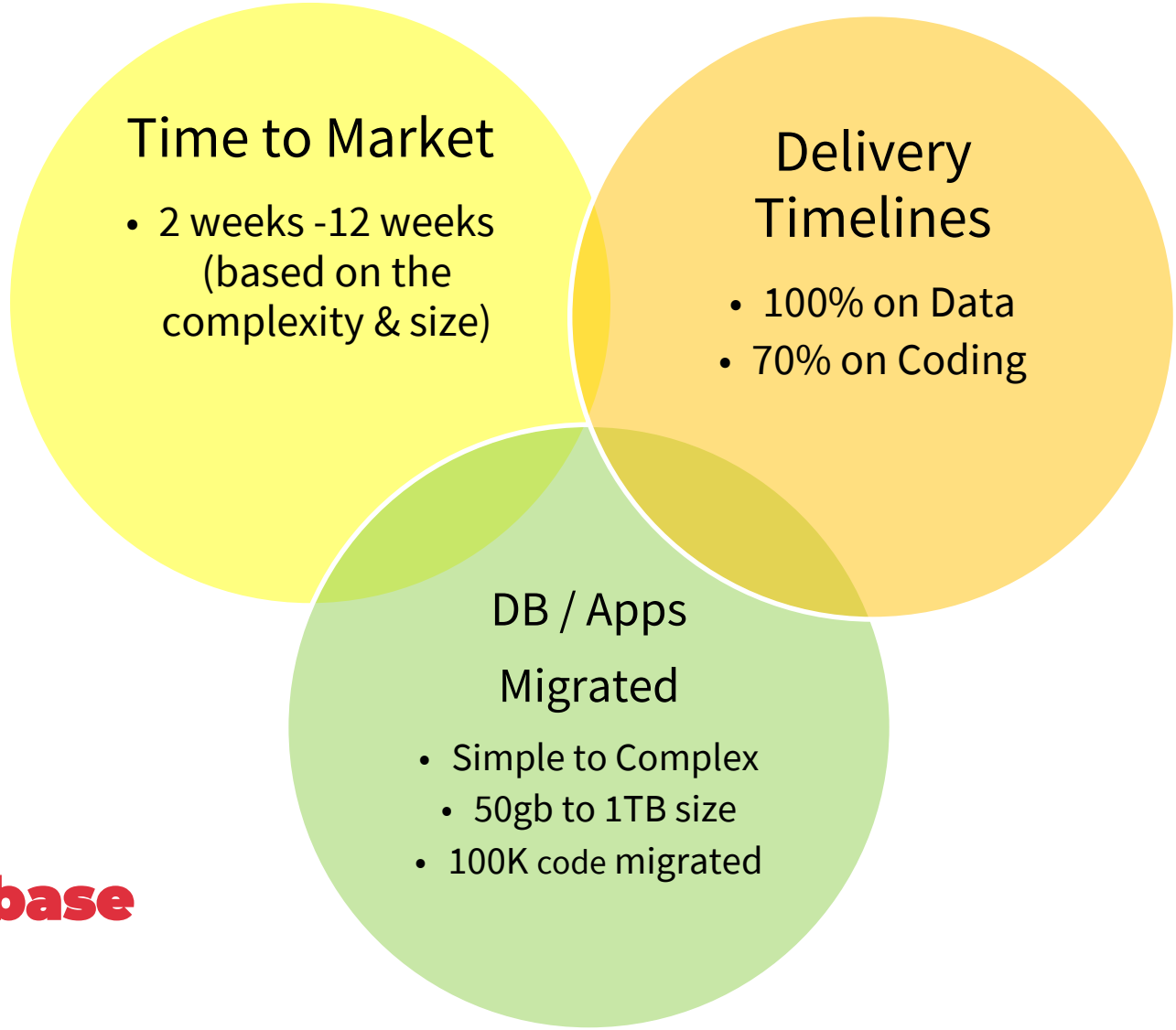


SOCIETE GENERALE

STATS

125+
MIGRATED DB

150+
DB IN THE PIPE



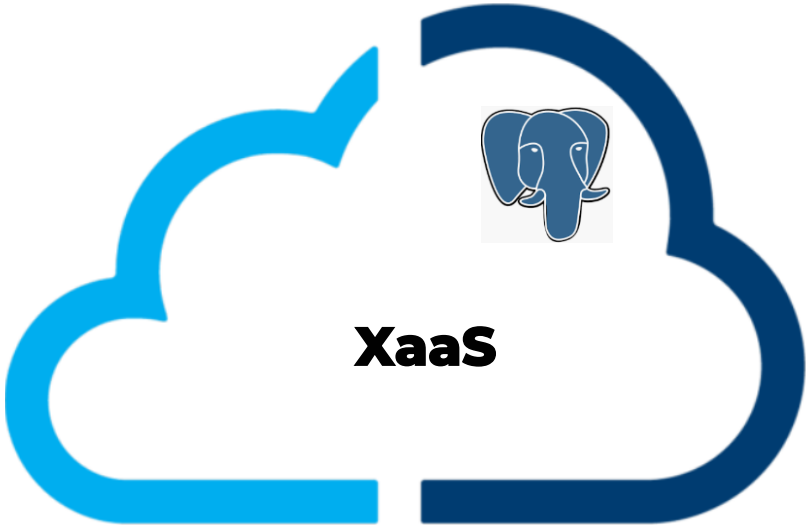
Technologies
Oracle, MSSQL, MySQL, Sybase

MIGRATION EXPERIENCE

SOCIETE GENERALE

PG XAAS & MULTI CLOUD

PostgreSQL – SG's Default Database



Anything as a Service

- Operating System
- Middleware
- Certificate
- DNS
- Storage (Object and File)
- Databases
- Bigdata
- Load Balancer

SOCIETE GENERALE

SHARING OUR EXPERIENCES ON :

- 3RD PARTY ENCRYPTION
- FDW AND MVIEW MAGIC
- LOB EXTRACTION
- BYTEA TO OID
- DYNAMIC PARTITIONS
- ETL TASKS
- OTHER NUGGETS
- PERFORMANCE

SOCIETE GENERALE

3RD PARTY ENCRYPTION

Challenge : Migrate the encrypted data.

- Decrypt data using source decryption algorithm

- Enable pgcrypto extension
- Define encryption key
- Use **pgp_sym_encrypt(data text, psw text [, options text]) returns bytea** to encrypt and load the data
- Use **pgp_sym_decrypt(msg bytea, psw text [, options text]) returns text** to decrypt and read the data

SOCIETE GENERALE

FDW AND MVIEW MAGIC

Challenge : Migrate data from multiple source, as daily activity.

Strategy:
- Extract the table structure using ora2pg.
- Create foreign servers and tables.
- Create Mview.
- Refresh Mview.
- Create batch jobs.

Outcome:
- Seamless merger of schemas from different applications.
- Faster migration approach for large tables, compared to ora2pg.

```
CREATE EXTENSION oracle_fdw;

CREATE SERVER oradb FOREIGN DATA WRAPPER oracle_fdw OPTIONS
        ( dbserver '//dbserver.mydomain.com:1521/ORADB' );

GRANT USAGE ON FOREIGN SERVER oradb TO pguser;

CREATE USER MAPPING FOR pguser SERVER oradb OPTIONS (
        USER 'orauser',
        PASSWORD '<pwd>'
);

CREATE FOREIGN TABLE ft_tab1 (col1 <datatype> ,col1 <datatype>)
        SERVER <foreign_server_name>
        OPTIONS ( SCHEMA 'oracle_schema_name', TABLE 'tab1' );

CREATE MATERIALIZED VIEW  tab1 AS SELECT
        col1, col2, col3
        FROM ft_tab1 WITH DATA;

REFRESH MATERIALIZED VIEW tab1 WITH data;
```

SOCIETE GENERALE

LOB EXTRACTION

Challenge : Data extraction of 300gb with LOB (character) was taking more than 72hrs.

Strategy:

- Table segregation LOW,MEDIUM,LARGE (LOB)
- Take the parallelism advantage of ora2pg.

```
- ora2pg - t COPY - o COPY_X_LOB_LOT
 < number >.sql - j 2 - T < LOCATION >
/ LOT < number > - b < LOCATION >
/ LOT < number > - c < LOCATION >
/ LOT < number > /
< specific_config > _LOT < number >.conf
```

- 8 core CPU and 64gb RAM

Outcome:

- Total extraction time came down to 12hrs from 72hrs.

ORA2PG Config Changes:

.

.

```
DATA_LIMIT 500              #DEFAULT 10000
BLOB_LIMIT 100              #DEFAULT    500 divide by 10
LONGREADLEN 102400000   #DEFAULT 1047552
ALLOW
EXCLUDE
```

.

.

SOCIETE GENERALE

BYTEA TO OID

Challenge : Specific application requirement to have OID column data type,while ora2pg does to BYTEA.

Issue while inserting data into OID column:

```
#  INSERT INTO tab1 SELECT * FROM tab1_bkp;
   ERROR: 42804: COLUMN col4 IS OF TYPE oid but expression IS OF TYPE bytea
   line 1: INSERT INTO tab1 SELECT * FROM tab1_bkp
                               ^
   HINT: You will need TO rewrite OR CAST the expression.
   LOCATION: transformAssignedExpr, parse_target.c :540
   Time: 0.427 ms
```

SOCIETE GENERALE

BYTEA TO OID

Strategy:

```
CREATE OR REPLACE FUNCTION blob_write (lbytea bytea)
                RETURNS oid VOLATILE
                LANGUAGE plpgsql
                AS $f$
DECLARE
                loid oid;
                lfd integer;
                lsize integer;
BEGIN
                IF
                (lbytea IS NULL)
                THEN RETURN NULL;
                END IF;
                loid := lo_create(0);
                lfd := lo_open(loid, 131072);
                lsize := lowrite(lfd, lbytea);
                PERFORM
                                lo_close(lfd);
                RETURN loid;
END;
$f$;
```

```
CREATE CAST( bytea AS oid ) WITH FUNCTION blob_write (bytea )
AS ASSIGNMENT ;

ALTER TABLE tab1 RENAME TO tab1_bkp;

CREATE TABLE tab1 (
  col1 bigint,
  col2 bigint,
  col3 oid );

INSERT INTO tab1 SELECT * FROM tab1_bkp;
```

Solution:

- Load the table, with BYTEA, from ora2pg COPY statement.
- Create blob_write function.
- Create CAST.
- Insert Data into table.

SOCIETE GENERALE

DYNAMIC PARTITIONS

Challenge : Dynamic partitions for future load after the migration.

Strategy:

- Application team wanted the source partitioned data to be moved to PG partitioned data.
- PG version 9.6
- Load the data into PG in a non-partitioned table.
- Partition it using INHERITANCE.
- Insert the data.
- Create trigger to Dynamically create partitions, for future loads.

Outcome:

- Much easier and fast to move the data to Postgres.
- No yearly creation of partitions.

Sample Steps:

```
CREATE TABLE tab1_2012_02
            ( audit_id bigint, datetime timestamp without time zone,
trigger_event character varying(255) )
 INHERITS ( tab1 );
CREATE TABLE tab1_2012_03
            ( audit_id bigint, datetime timestamp without time zone,
trigger_event character varying(255) )
INHERITS ( tab1 );

CREATE TABLE tab1_14aug2019 AS TABLE tab1;

TRUNCATE TABLE tab1;

INSERT INTO tab1_2012_02
            SELECT * FROM tab1_14aug2019
            WHERE datetime >= '2012-02-01 00:00:00' AND datetime <=
'2012-02-29 23:59:00';
INSERT INTO tab1_2012_03
            SELECT * FROM tab1_14aug2019
            WHERE datetime >= '2012-03-01 00:00:00' AND datetime <=
'2012-03-31 23:59:00';
```

SOCIETE GENERALE

DYNAMIC PARTITIONS

```
CREATE OR REPLACE FUNCTION dyn_create_partition_and_insert ()
            RETURNS TRIGGER
      AS $BODY$
DECLARE
            partition_date text; PARTITION TEXT;month_start text;month_end text;
BEGIN
  month_start := date_trunc('month', NEW.datetime);  month_end := (date_trunc('month', NEW.datetime) + interval '1 MONTH - 1 day');
  partition_date := to_char(NEW.datetime, 'YYYY_MM'); PARTITION := TG_TABLE_NAME || '_' || partition_date;
IF NOT EXISTS (
  SELECT relname FROM pg_class WHERE relname = PARTITION) THEN RAISE NOTICE 'A partition has been created %', PARTITION;
    RAISE NOTICE 'value of month start %', month_start; RAISE NOTICE 'value of month end %', month_end;
    RAISE NOTICE 'value of datetime %', NEW.datetime;
EXECUTE 'CREATE TABLE ' || PARTITION || ' (check (datetime >= ''' || month_start || ''' AND datetime <= ''' || month_end || ''’))
INHERITS (' || TG_TABLE_NAME || ');’;
END IF;
    EXECUTE 'INSERT INTO ' || PARTITION || ' SELECT(' || TG_TABLE_NAME || ' ' || quote_literal(NEW) || ').*;’;
    RETURN NULL; END; $BODY$
LANGUAGE plpgsql
VOLATILE
COST 100;
```

SOCIETE GENERALE

DYNAMIC PARTITIONS

Sample Steps:

```
CREATE TRIGGER tab1_part_insert_trigger
    BEFORE INSERT ON tab1
FOR  EACH ROW
EXECUTE PROCEDURE dyn_create_partition_and_insert ();
```

```
CREATE VIEW show_partitions_tab1
AS SELECT nmsp_parent.nspname
AS parent_schema, parent.relname
AS parent, nmsp_child.nspname
AS child_schema, child.relname
AS child
FROM pg_inherits
JOIN pg_class parent ON pg_inherits.inhparent = parent.oid
JOIN pg_class child ON pg_inherits.inhrelid = child.oid
JOIN pg_namespace nmsp_parent ON nmsp_parent.oid =
parent.relnamespace
JOIN pg_namespace nmsp_child ON nmsp_child.oid =
child.relnamespace
WHERE parent.relname = 'tab1';
```

```
INSERT INTO tab1 VALUES (7926, '2019-09-17
14:40:00', 'INSERT');

CHECK the Partitions:
--------------------
SELECT * FROM show_partitions_tab1 WHERE child LIKE '%2019%';

parent_schema | parent | child_schema | child
 ------------------+--------+----------------+-----------------
XXXXXXXXX    | tab1   | XXXXXXXXX | tab1_2019_02
XXXXXXXXX    | tab1   | XXXXXXXXX | tab1_2019_03
XXXXXXXXX    | tab1   | XXXXXXXXX | tab1_2019_04
```

SOCIETE GENERALE

ETL TASKS – BE VIGILANT

Challenges :

- Post migration of database and batch job code to postgresql, ETL was taking 6 hrs. instead of **required 20min**

- Handling discarded rows

Trials:

- 1. We tried approaches like pgloader, insert statements, tuning of the odbc drivers.

- 2. Used the insert and parallel mechanism to improve the time and it came down to **18mins.**

- Wrote specific function "**pg_discard**"

OTHER NUGGETS

Facts	Oracle	PostgreSQL
Hierarchical queries	CONNECT BY	WITH recursive
Explicit commit/rollback within a SP	COMMIT/ROLLBACK	Version <11 use dblink inside the function
Global variables in a package	DECLARE in package spec	to set : set Globalvar.var1 = 'test'; to get : current_setting('Globalvar.var1');
Autonomous transaction	pragma AUTONOMOUS_TRANSACTION	dblink
Joins	Supports Non ANSI and ANSI joins	Supports only ANSI
Read/ write from/to OS file	UTL_FILE package	UTL_FILE provided by orafce extension

PERFORMANCE - SUCCESS FACTOR

90% - Improved performance using **WITH CTE** (Common Table Expressions) on Corelated subquery

30% - Creating indexes and avoiding UPPER and LOWER functions in WHERE eg. Think better alternatives
upper(column1) = upper('test')

120% - Rewriting the logic (predicate clause)

176% - Improved by modifying obsolete library for ODBC client (13hrs to 1 hr)

SOCIETE GENERALE

OUR KEY TOOLS

SOCIETE GENERALE

TOOLS WE USE

ORA2PG - USED FOR DATABASE ESTIMATION & MIGRATION

CODE2PG - USED FOR APPLICATION (EMBEDED SQL) ESTIMATION & MIGRATION HINTS

SOCIETE GENERALE

ORA2PG

- Migration tool from Oracle to PostgreSQL developed by Gilles Darold.
- First release in 2001, v20 today.
- Added MySQL to PostgreSQL recently.
- We @Migration Factory uses it extensively for estimations & migrations
- SG’s developers had enhanced to add MSSQL estimation & Migration to PostgreSQL
    - Link : https://github.com/societe-generale/ora2pg-mssql
    - Discussion in progress with Gilles Darold for code integration .

ORA2PG

FOR ORACLE

```
ora2pg --project_base /home/user/ora2pg --init_project aw_project
[…]

$ cat ora2pg.conf
[...]

ORACLE_DSN     dbi:Oracle:host=myhost;sid=mydb;port=myport
ORACLE_USER   myoracleuser
ORACLE_PWD    myoraclepwd

[...]

$ ora2pg -t SHOW_VERSION -c config/ora2pg.conf

Oracle Database 11g Enterprise Edition Release 11.2.0.3.0
```

Available on https://github.com/darold/ora2pg/releases

SOCIETE GENERALE

ORA2PG

FOR MS SQL

```
ora2pg --project_base /home/user/ora2pg --init_project aw_project
[…]

$ cat ora2pg.conf
[...]

ORACLE_DSN      dbi:ODBC:Driver = ODBC Driver 17 FOR SQL SERVER; SERVER = myserver.myhost, myport; DATABASE = mydatabase
ORACLE_USER     mssql_user
ORACLE_PWD      mssql_pwd
[...]

$ ora2pg -t SHOW_VERSION -c config/ora2pg.conf

Microsoft SQL SERVER 2014 (SP1 - GDR) (KB4019091) - 12.0.4237.0 (X64)
```

Available on https://github.com/societe-generale/ora2pg-mssql

SOCIETE GENERALE

ORA2PG : ORACLE

Ora2Pg - Database Migration Report

Version Oracle Database 12c Enterprise Edition Release 12.1.0.2.0
Schema HR
Size 9.62 MB

Object	Number	Invalid	Estimated cost	Comments	Details
DATABASE LINK	4	0	12	Database links will be exported as SQL/MED PostgreSQL's Foreign Data Wrapper (FDW) extentions using oracle_fdw.	
FUNCTION	2	0	9	Total size of function code: 421 bytes.	get_tab_ptf: 4 get_tab_tf: 3
INDEX	29	0	4.8	17 index(es) are concerned by the export, others are automatically generated and will do so on PostgreSQL. Bitmap index(es) will be exported as b-tree index(es) if any. Cluster, domain, bitmap join and IOT indexes will not be exported at all. Reverse indexes are not exported too, you may use a trigram-based index (see pg_trgm) or a reverse() function based index and search. Use 'varchar_pattern_ops', 'text_pattern_ops' or 'bpchar_pattern_ops' operators in your indexes to improve search with the LIKE operator respectively into varchar, text or char columns.	5 domain index(es) 1 function based b-tree index(es) 11 b-tree index(es)
JOB	0	0	0	Job are not exported. You may set external cron job with them.	
MATERIALIZED VIEW	2	0	6	All materialized view will be exported as snapshot materialized views, they are only updated when fully refreshed.	
PACKAGE BODY	2	0	44	Total size of package code: 2992 bytes. Number of procedures and functions found inside those packages: 6.	emp_mgmt.create_dept: 3 emp_mgmt.hire: 11 emp_mgmt.increase_comm: 3 emp_mgmt.increase_sal: 3 emp_mgmt.remove_dept: 3 emp_mgmt.remove_emp: 3
PROCEDURE	2	0	8	Total size of procedure code: 772 bytes.	secure_dml: 3 add_job_history: 3
SEQUENCE	4	0	0.4	Sequences are fully supported, but all call to sequence_name.NEXTVAL or sequence_name.CURRVAL will be transformed into NEXTVAL('sequence_name') or CURRVAL('sequence_name').	
SYNONYM	0	0	0	SYNONYMs will be exported as views. SYNONYMs do not exists with PostgreSQL but a common workaround is to use views or set the PostgreSQL search_path in your session to access object outside the current schema.	emp_details_view_v is an alias to HR.EMP_DETAILS_VIEW public.emp_table is a link to hr.employees@curr_user offices is an alias to HR.LOCATIONS
TABLE	36	0	18.2	1 external table(s) will be exported as file_fdw foreign table. See EXTERNAL_TO_FDW configuration directive to export as standard table or use COPY in your code if you just want to load data from external files. 2 check constraint(s).	1 binary columns 5 unknow types Total number of rows: 1552 Top 5 of tables sorted by number of rows: customer_summary has 1154 rows

SOCIETE GENERALE

ORA2PG : MS SQL

Ora2Pg - Database Migration Report

Version Microsoft SQL Server 2017 (RTM-CU9-GDR) (KB4293805) - 14.0.3035.2 (X64)
Schema
Size 336.00 MB

Object	Number	Invalid	Estimated cost	Comments	Details
DATABASE LINK	1	0	3	Database links will be exported as SQL/MED PostgreSQL's Foreign Data Wrapper (FDW) extensions using tds_fdw.	
FUNCTION	11	0	65	Total size of function code: 7460 bytes.	ufngetstock: 5 ufngetcontactinformation: 4 ufngetpurchaseorderstatustext: 3 ufngetaccountingenddate: 7 ufngetproductlistprice: 5 ufngetproductdealerprice: 5 ufngetdocumentstatustext: 3 ufngetaccountingstartdate: 5 ufnleadingzeros: 9 ufngetsalesorderstatustext: 3 ufngetproductstandardcost: 5
GLOBAL TEMPORARY TABLE	0	0	0	Global temporary table are not supported by PostgreSQL and will not be exported. You will have to rewrite some application code to match the PostgreSQL temporary table behavior.	
INDEX	36	0	3.6	0 index(es) are concerned by the export, others are automatically generated and will do so on PostgreSQL. Bitmap will be exported as btree_gin index(es) and hash index(es) will be exported as b-tree index(es) if any. Domain index are exported as b-tree but commented to be edited to mainly use FTS. Cluster, bitmap join and IOT indexes will not be exported at all. Reverse indexes are not exported too, you may use a trigram-based index (see pg_trgm) or a reverse() function based index and search. Use 'varchar_pattern_ops', 'text_pattern_ops' or 'bpchar_pattern_ops' operators in your indexes to improve search with the LIKE operator respectively into varchar, text or char columns.	
JOB	0	0	0	Jobs are not exported. You may set external cron job with them.	
PROCEDURE	10	0	96	Total size of procedure code: 15593 bytes.	uspprinterror: 15 uspupdateemployeehireinfo: 5 uspgetwhereusedproductid: 8 usplogerror: 22 uspgetbillofmaterials: 8 uspupdateemployeepersonalinfo: 7 uspupdateemployeelogin: 5 uspgetemployeemanagers: 3 uspsearchcandidateresumes: 10 uspgetmanageremployees: 3
SYNONYM	2	0	1	SYNONYMs will be exported as views. SYNONYMs do not exists with PostgreSQL but a common workaround is to use views or set the PostgreSQL search_path in your session to access object outside the current schema.	syn_schematest is an alias to .dbo.test synonym_test is an alias to .dbo.test
TABLE	75	0	40.3	88 check constraint(s).	7 binary columns 2 reserved words in column name Total number of rows: 760837

SOCIETE GENERALE

CODE2PG

LINK :- https://github.com/societe-generale/code2pg

ESTIMATIONS

In man/days

Report formats available in html, txt

CONFIGURABLE

Optional config file

Estimations can be adjusted depending on team expertise

Can simulate orafce usage

TECH

Perl script

Several Perl modules needed

DEVELOPER

Help him identify code to migrate and propose an equivalent instruction if possible

Can tag source code

EXTRA

SVN direct access

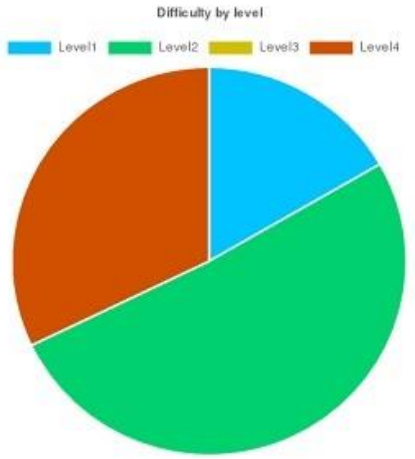
Oracle, SQL Server, DB2

CODE2PG

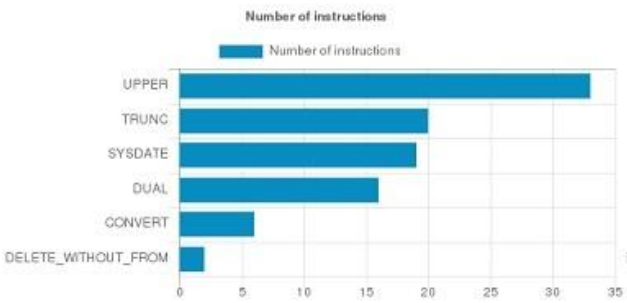
```
./code2pg -e java -e jsp -l comma-strings -d /tmp/project -o report.html
```

code2pg

Version	0.9.1
Analysis date	12/12/2017
Source code directory	https://
Number of files with extension .java	130
Language	java
Number of analyzed LOC	0
Log file	
orafce usage	No



Level	Number of instructions	Time/instruction	Estimated time (minutes)	Man-days
1	51	1	51	0.14
2	39	4	156	0.43
3	0	8	0	0.00
4	6	16	96	0.27
TOTAL	96	3.16	303	**0.84**



SOCIETE GENERALE

CODE2PG

```
user@localhost # ./code2pg --help ADD_MONTHS

ADD_MONTHS
==========

ORACLE
------

- Instruction: ADD_MONTHS
- Documentation: http://docs.oracle.com/database/121/SQLRF/functions011.htm

POSTGRESQL
----------

- PostgreSQL instruction: No equivalent
- Documentation:
- Level 3 - estimated time : 8 minutes.

COMMENTS
---------

A function can be created. For example:
    ```sql
    CREATE OR REPLACE FUNCTION public.add_months(date date, months integer)
     RETURNS date
     LANGUAGE plpgsql
    AS $function$
    BEGIN
    RETURN (date + (months * '1 month' :: interval)) :: date;
    END;
    $function$
    ```
```

SOCIETE GENERALE

WAYS TO KNOW MORE

Contact :

Janakiraman VENUGOPALAN
Janakiraman.venugopalan@socgen.com

Jyotiprasad RATH
jyotiprasad.rath@socgen.com

https://github.com/societe-generale

https://github.com/societe-generale/ora2pg-mssql

https://github.com/societe-generale/code2pg

SOCIETE GENERALE

Q & A

SOCIETE GENERALE

THE FUTURE IS YOU

SOCIETE GENERALE