CERN

Application Continuity, seen from CERN

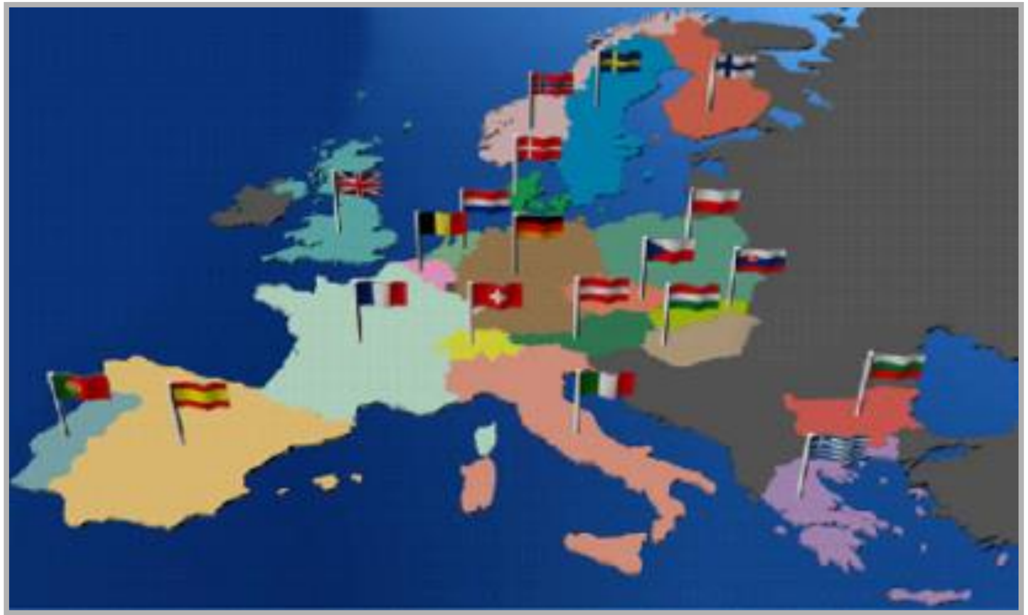
Andrei Dumitru – CERN IT

Agenda

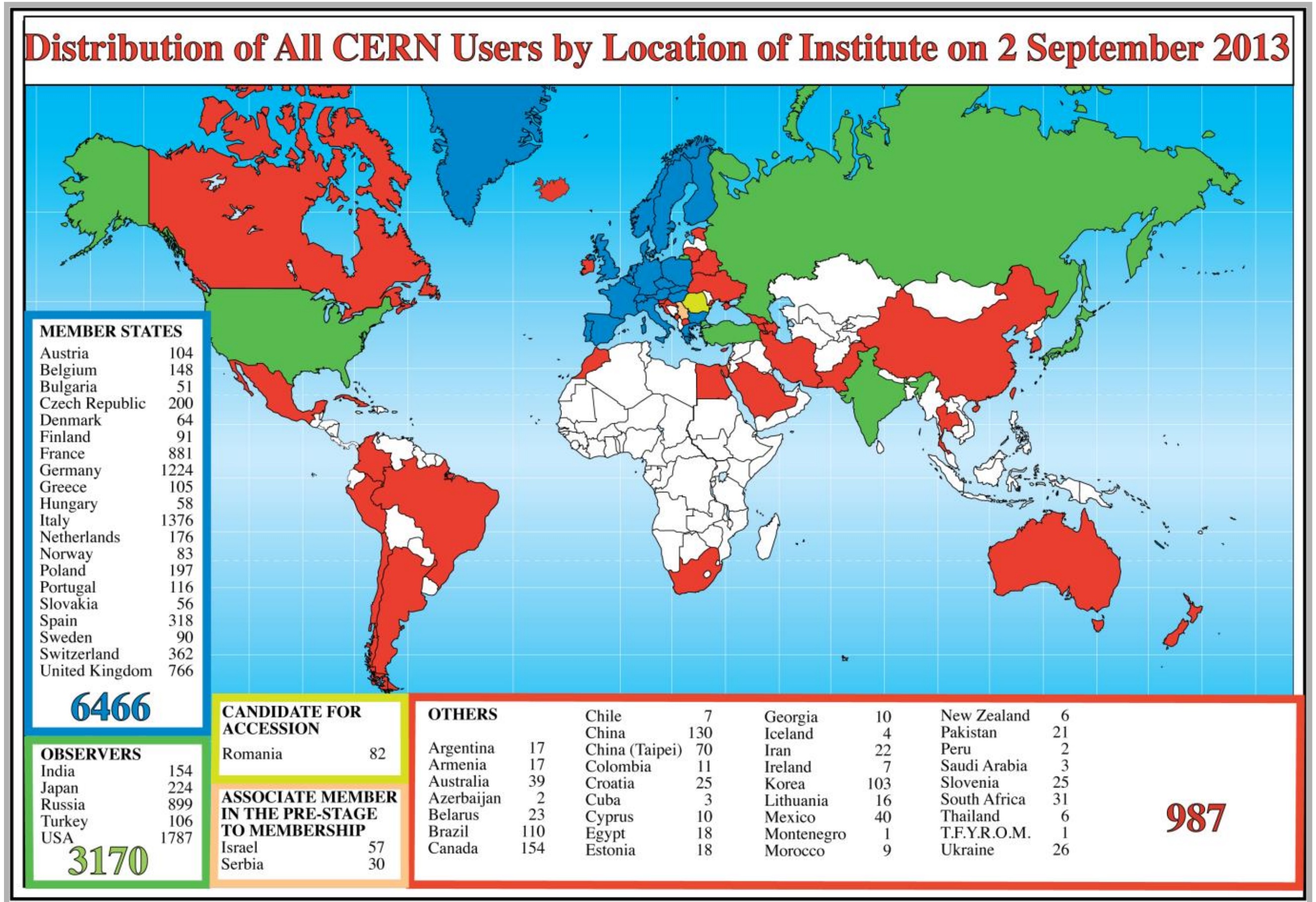
- About CERN
- Transaction Guard
- Application Continuity
- CERN’s experience

CERN

- European Organization for Nuclear Research founded in 1954
- 20 Member States, 5 Observer States + UNESCO and UE
- 60 Non-member States collaborate with CERN
- 2400 staff members work at CERN as personnel, 10 000 more researchers from institutes world-wide



Collaboration on an international scale



Distribution of All CERN Users by Location of Institute on 2 September 2013

MEMBER STATES	
Austria	104
Belgium	148
Bulgaria	51
Czech Republic	200
Denmark	64
Finland	91
France	881
Germany	1224
Greece	105
Hungary	58
Italy	1376
Netherlands	176
Norway	83
Poland	197
Portugal	116
Slovakia	56
Spain	318
Sweden	90
Switzerland	362
United Kingdom	766
	6466

OBSERVERS	
India	154
Japan	224
Russia	899
Turkey	106
USA	1787
	3170

CANDIDATE FOR ACCESSION	
Romania	82

ASSOCIATE MEMBER IN THE PRE-STAGE TO MEMBERSHIP	
Israel	57
Serbia	30

OTHERS							
		Chile	7	Georgia	10	New Zealand	6
		China	130	Iceland	4	Pakistan	21
Argentina	17	China (Taipei)	70	Iran	22	Peru	2
Armenia	17	Colombia	11	Ireland	7	Saudi Arabia	3
Australia	39	Croatia	25	Korea	103	Slovenia	25
Azerbaijan	2	Cuba	3	Lithuania	16	South Africa	31
Belarus	23	Cyprus	10	Mexico	40	Thailand	6
Brazil	110	Egypt	18	Montenegro	1	T.F.Y.R.O.M.	1
Canada	154	Estonia	18	Morocco	9	Ukraine	26

987

LHC, Experiments, Physics

Large Hadron Collider (LHC)

- World’s largest and most powerful particle accelerator
- 27-kilometre ring of superconducting magnets
- Currently undergoing upgrades, restart in 2015

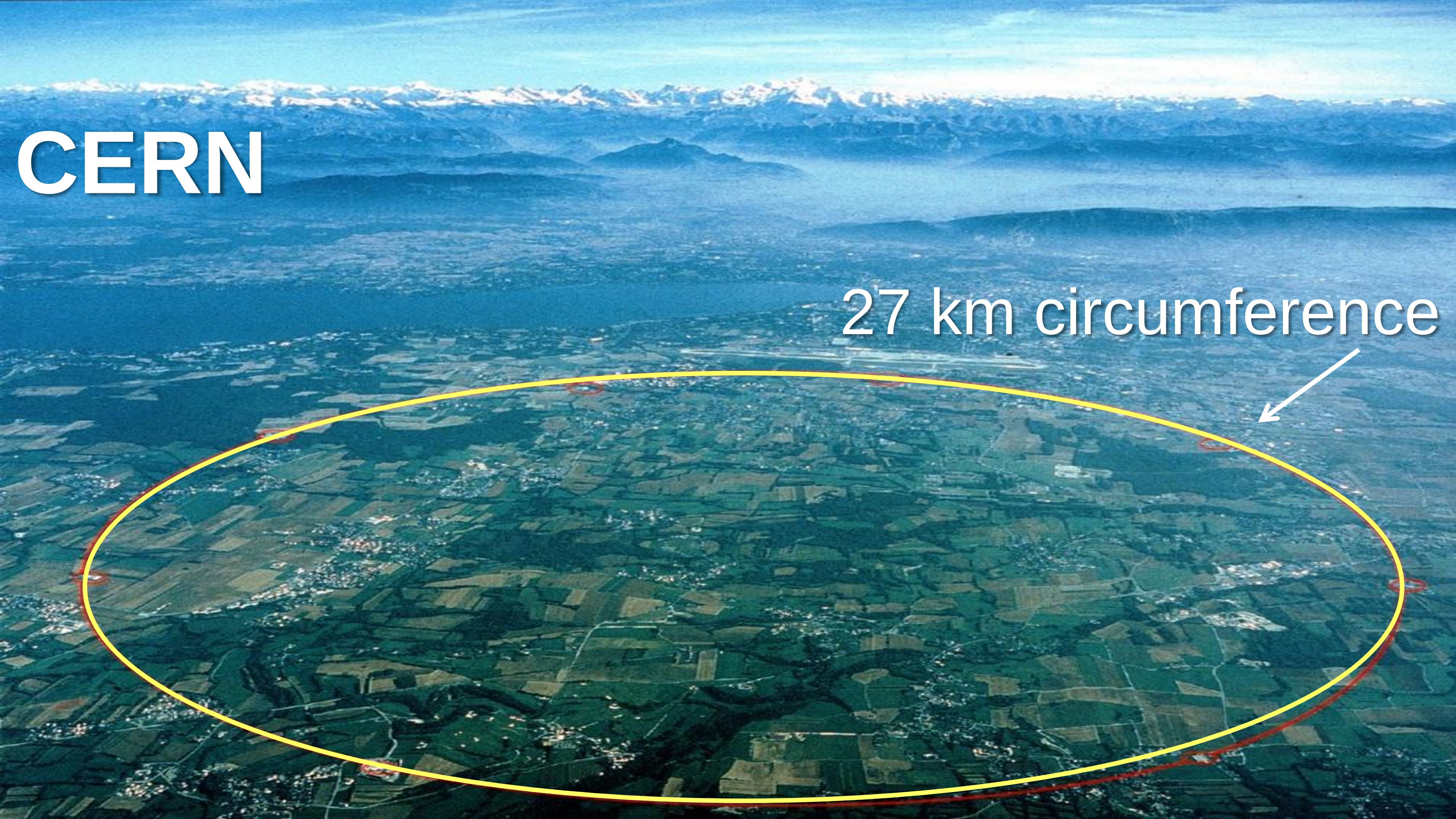
The products of particle collisions

- captured by complex detectors AND analyzed by software in the experiments dedicated to LHC

Higgs particle discovered!

The Nobel Prize in Physics 2013 to François Englert and Peter W. Higgs *"for the theoretical discovery of a mechanism that contributes to our understanding of the origin of mass of subatomic particles, and which recently was confirmed through the discovery of the predicted fundamental particle, by the ATLAS and CMS experiments at CERN's Large Hadron Collider"*

CERN



27 km circumference

CERN openlab

- Public-private partnership
- Mission:
  - accelerate the development of cutting-edge solutions to be used by the worldwide LHC community
- Neutral ground for advanced R&D

Partners

HUAWEI

intel

ORACLE

SIEMENS

Contributors

rackspace
the open cloud company

Associates

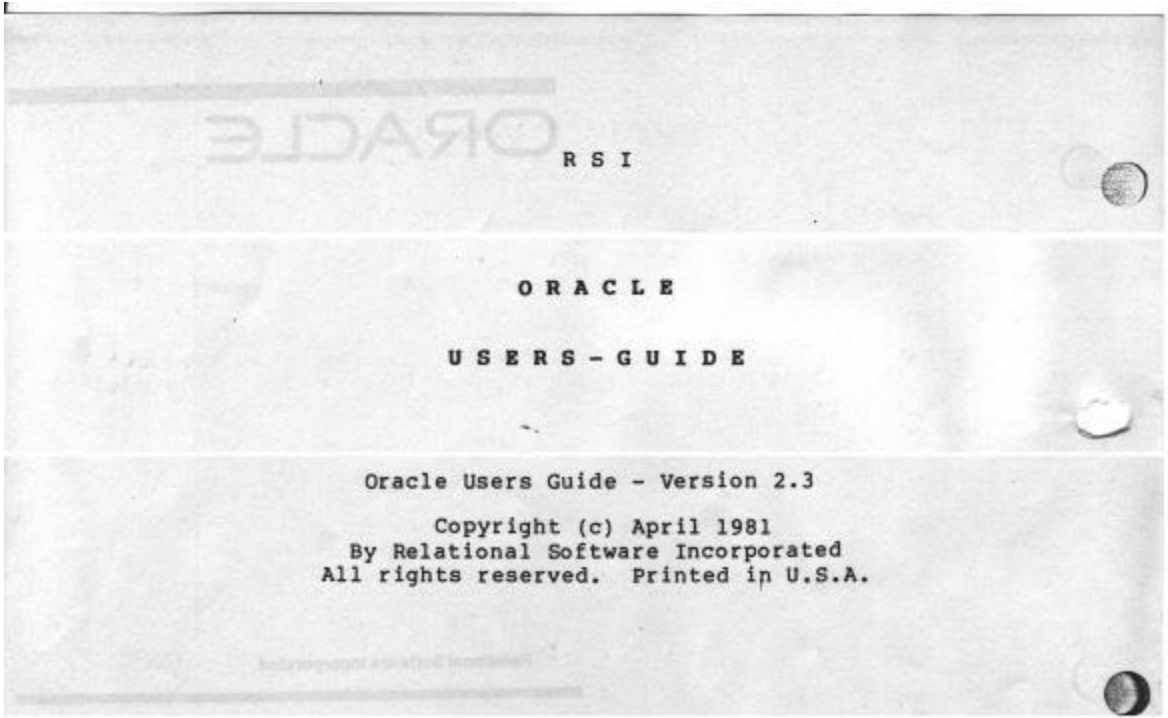
Yandex

CERN

CERN openlab

Oracle at CERN

- 1982: start with Oracle at CERN (*accelerator control*)



R S I

ORACLE

USERS-GUIDE

Oracle Users Guide - Version 2.3

Copyright (c) April 1981
By Relational Software Incorporated
All rights reserved. Printed in U.S.A.

Credit: N. Segura Chinchilla

Oracle at CERN

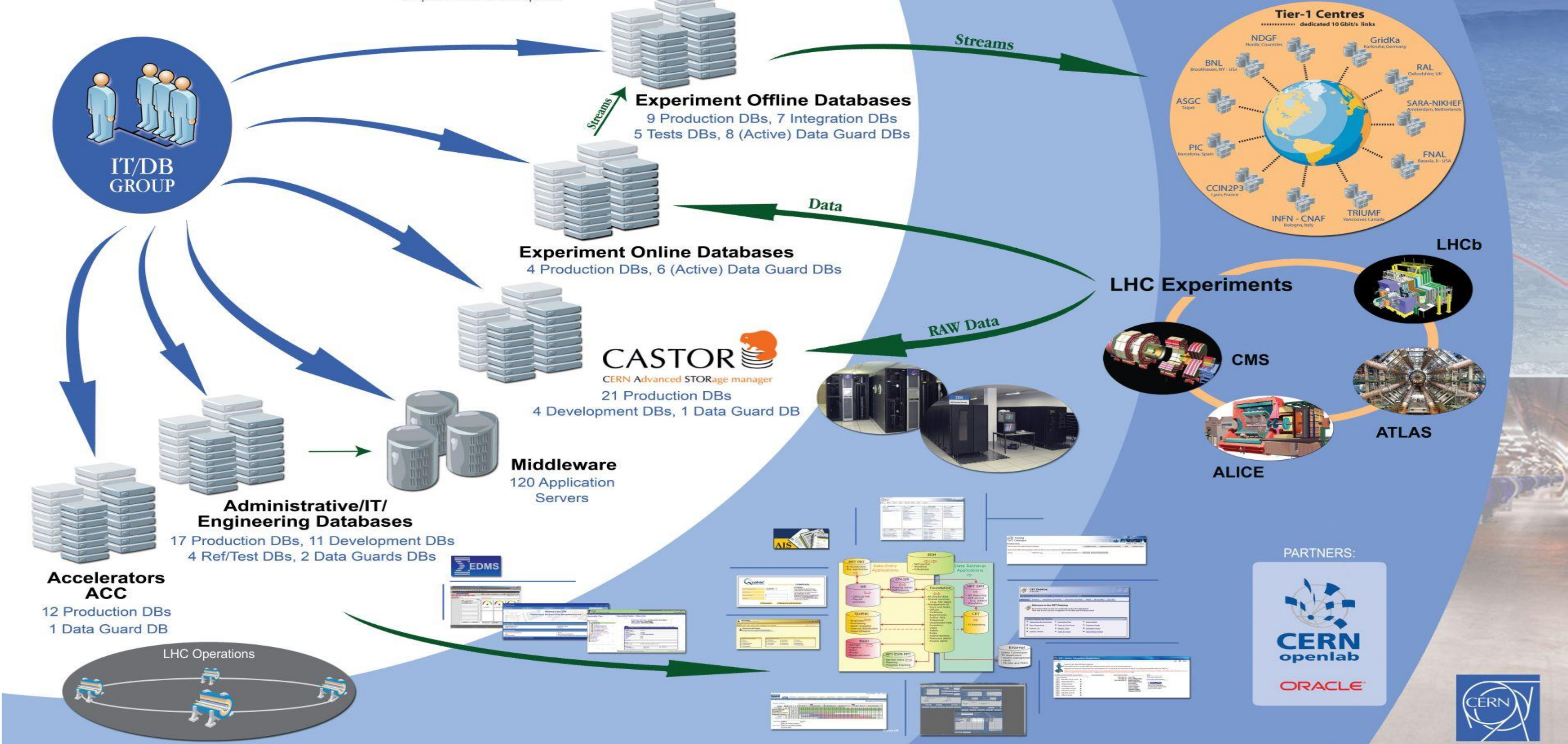
- Relational DBs play a key role in the LHC production chains
  - Accelerator logging and monitoring systems
  - Online acquisition, offline: data (re)processing, data distribution, analysis
  - Grid infrastructure and operation services
    - Monitoring, dashboards, etc.
  - Data management services
    - File catalogues, file transfers, etc.
  - Metadata and transaction processing for tape storage system

Database SERVICES

At the heart of CERN, LHC and Experiment Operations

CERN IT Department

http://cern.ch/it-dep/db/



IT/DB GROUP

Experiment Offline Databases
9 Production DBs, 7 Integration DBs
5 Tests DBs, 8 (Active) Data Guard DBs

Experiment Online Databases
4 Production DBs, 6 (Active) Data Guard DBs

CASTOR
CERN Advanced STORage manager
21 Production DBs
4 Development DBs, 1 Data Guard DB

Middleware
120 Application Servers

Administrative/IT/ Engineering Databases
17 Production DBs, 11 Development DBs
4 Ref/Test DBs, 2 Data Guards DBs

Accelerators ACC
12 Production DBs
1 Data Guard DB

Tier-1 Centres
dedicated 10 Gbit/s links

LHC Experiments

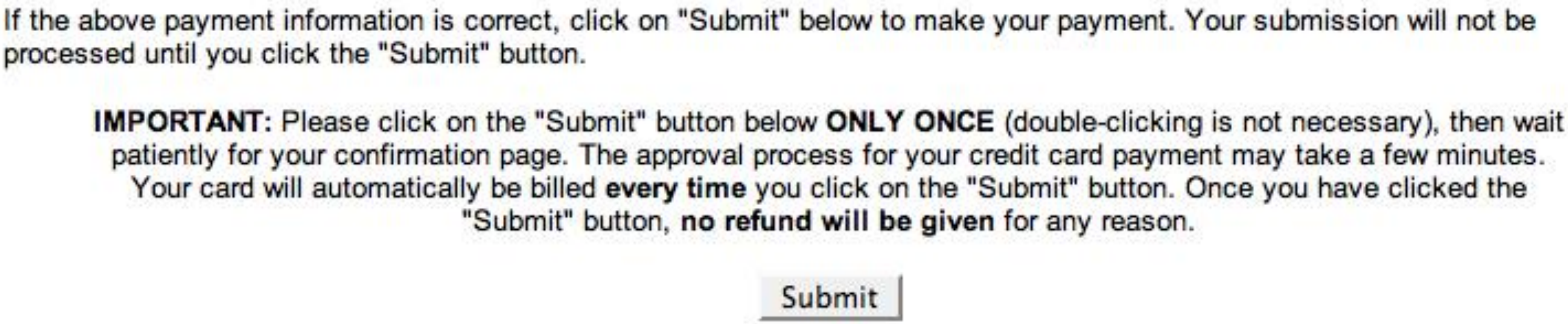
PARTNERS:
CERN openlab
ORACLE

Current status on June 2013

Agenda

- About CERN
- Transaction Guard
- Application Continuity
- CERN’s experience

CERN



If the above payment information is correct, click on "Submit" below to make your payment. Your submission will not be processed until you click the "Submit" button.

IMPORTANT: Please click on the "Submit" button below **ONLY ONCE** (double-clicking is not necessary), then wait patiently for your confirmation page. The approval process for your credit card payment may take a few minutes. Your card will automatically be billed **every time** you click on the "Submit" button. Once you have clicked the "Submit" button, **no refund will be given** for any reason.

Source: http://www.grokdotcom.com

Transaction Guard

- Protocol and developer API
  - JDBC, OCI, OCCI, and ODP.Net drivers
- Applications know exact transaction status
  - Take informed decision on how to continue
- Used transparently by Application Continuity

Transaction Guard Coverage

Supported:

- Local transactions
- DDL and DCL transactions
- Distributed and Remote transactions
- Parallel transaction
- Commit on Success (auto-commit)
- PL/SQL with embedded COMMIT

Intentional Exclusions:

- Recursive transactions
- Autonomous transactions

Exclusions 12.1:

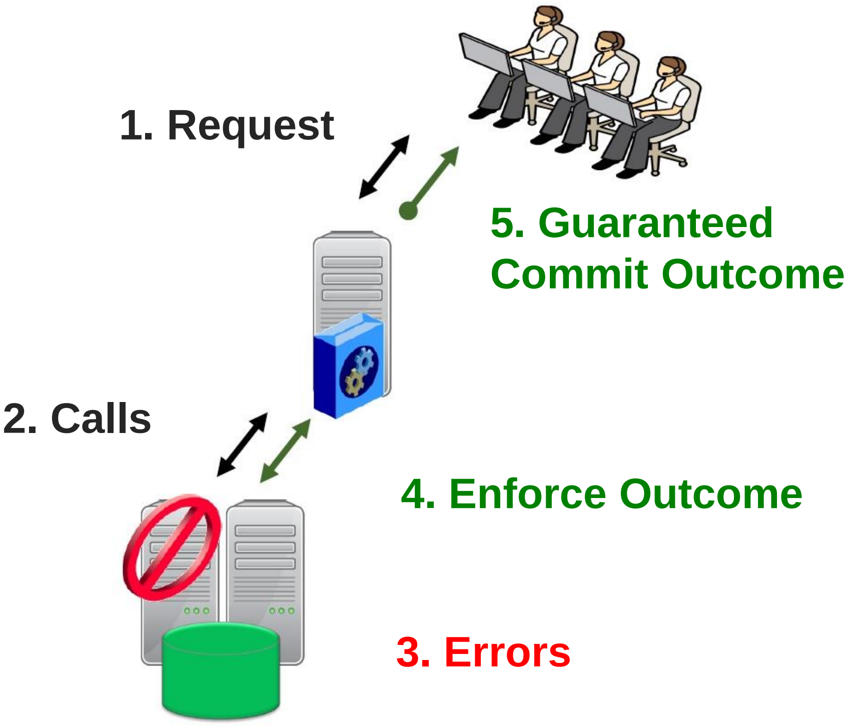
- XA transactions
- Active Data Guard with r/w DB Links for forwarding transactions
- Replication to Golden Gate
- Replication to Logical Standby

Database Concepts

- Logical Transaction Identifier (LTXID)
  - used to obtain the commit outcome
  - at-most-once semantics
- Reliable Commit Outcome
- Recoverable Error - external system failure
  - JDBC throws `SQLRecoverableException`

Transaction Guard

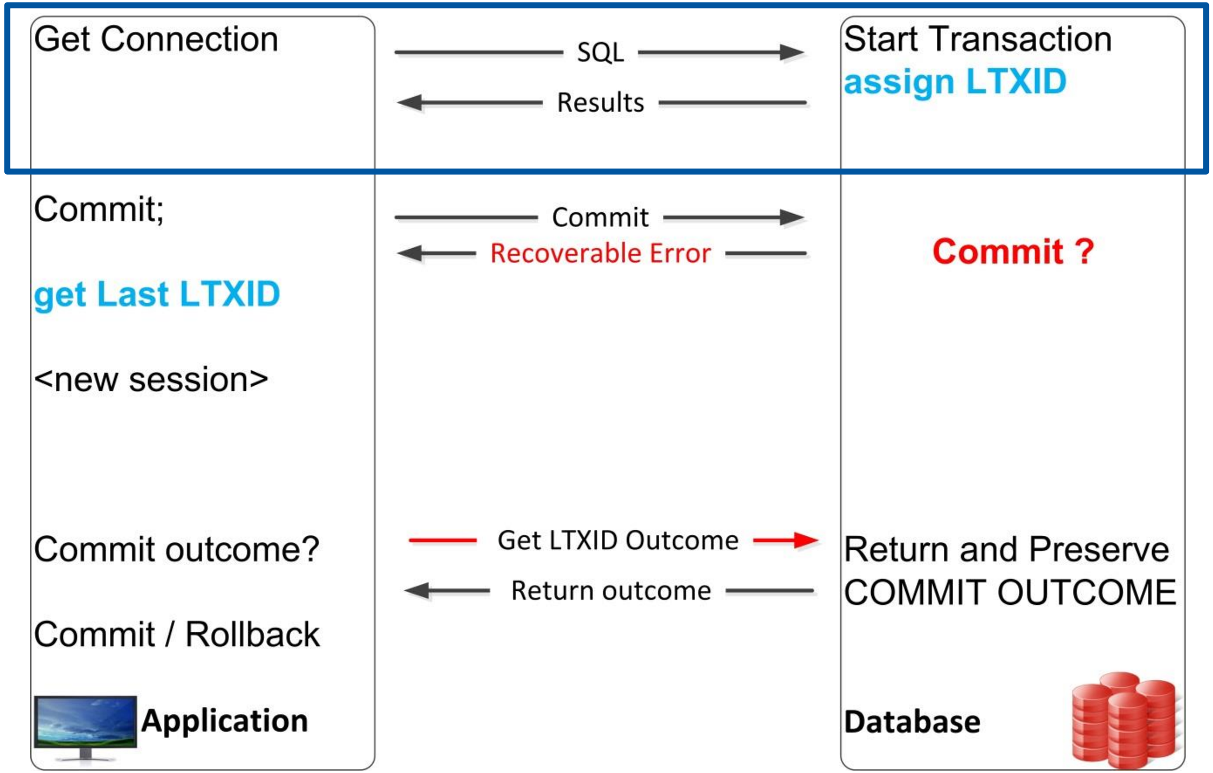
- Reliable commit outcome of transaction after an outage
- Durable commit outcome
- Provides transaction idempotence
- Prevents logical corruption



Oracle Database 12c

Transaction Guard

Database – Application Interaction



Get Connection

SQL

Results

Start Transaction

assign LTXID

Commit;

Commit

Recoverable Error

Commit ?

get Last LTXID

<new session>

Commit outcome?

Get LTXID Outcome

Return outcome

Return and Preserve COMMIT OUTCOME

Commit / Rollback

Application

Database

Transaction Guard Configuration

- Oracle Database ≥ 12.1
- Application Service (`srvctl or DBMS_SERVICE`)
- COMMIT_OUTCOME = TRUE
- RETENTION_TIMEOUT
- Grant execute on DBMS_APP_CONT package

```
srvctl add service -d tgtest -s tg_service
-r tgtest1 -a tgtest2
-commit_outcome TRUE
-retention 86400
```

Agenda

- About CERN
- Transaction Guard
- Application Continuity
- CERN's experience

Challenge

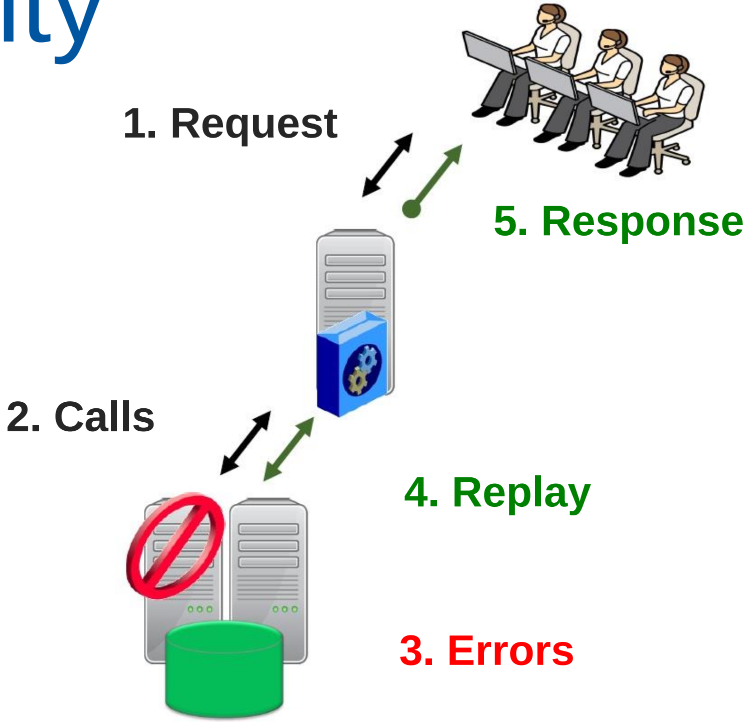
- Mask outages
- Ensure continuity of applications
- Difficulty to implement restart
- Interruptions required (patch)
- Issues happen (OS or DB), errors…

Application Continuity 12c

- Replays the operations performed before the interruption in case of recoverable errors
- Availability
  - Oracle RAC, Data Guard, WebLogic
  - JDBC Thin Driver, UCP or 3rd party Java Pools

Application Continuity

- Replays in-flight work on recoverable errors
- Masks many outages in a safe way
- Increase systems fault-tolerance

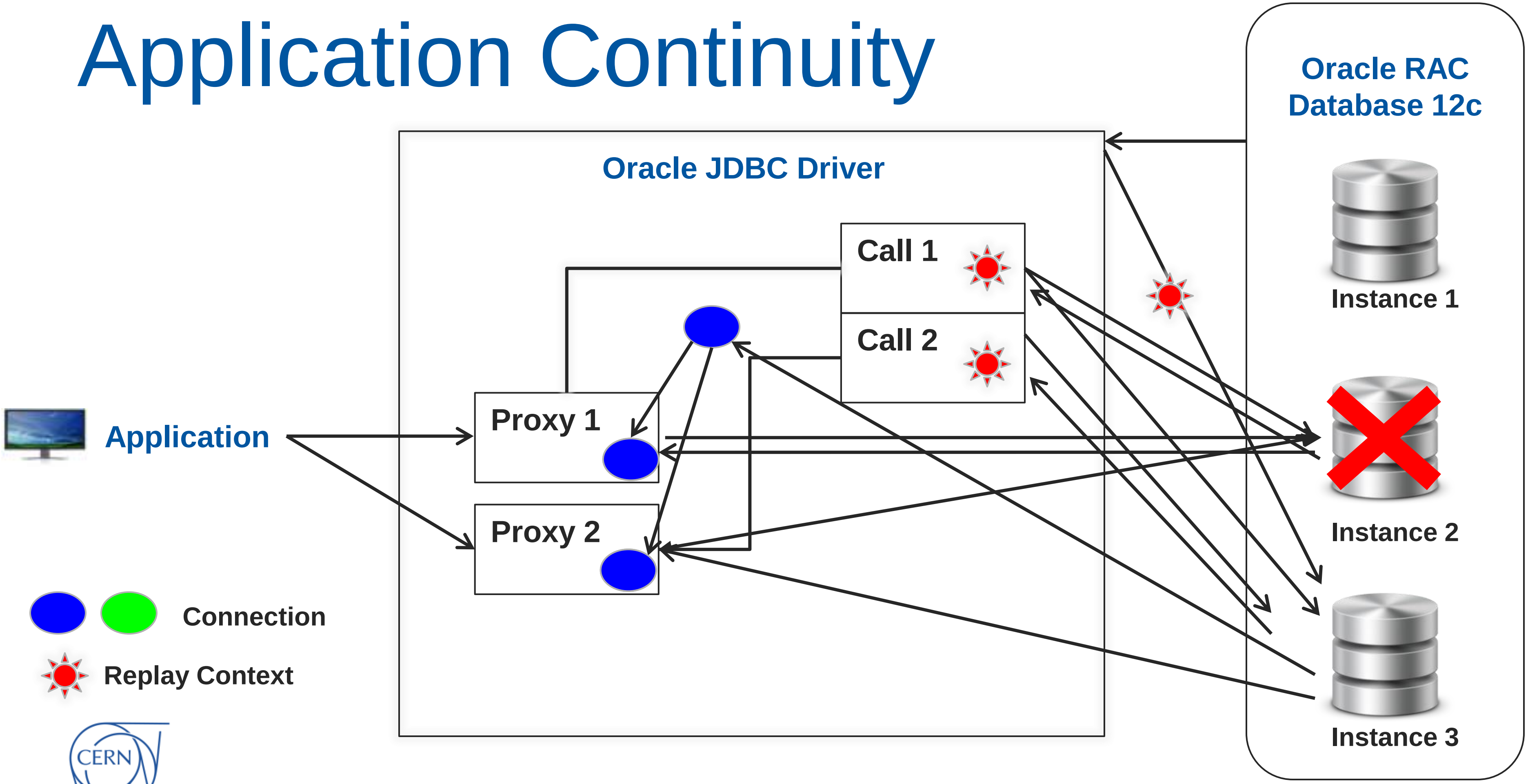


Oracle Database 12c

Application Continuity Phases

1. Capture (Normal Runtime)	2. Reconnect	3. Replay
• Identifies database requests • Decides what is replayable and what is not • Builds proxy objects • Holds original calls with validation	• Ensures request has replay enabled • Handles timeouts • Creates a new connection • Validates target database • Uses Transaction Guard to enforce last outcome	• Replays held calls • Continues replay, if user visible results match, based on validations • Continues request

Application Continuity



Application Continuity and Mutables

- Keep original mutable values for replay ?
- Support for keeping mutable object values
  - SYSDATE
  - SYSTIMESTAMP
  - SYS_GUID (only for serial execution plans)
  - sequence.NEXTVAL

```
grant KEEP SEQUENCE on actest.seq1 to appuser;
ALTER SEQUENCE my_seq KEEP;
```

Potential Side Effects

When replay is enabled, calls are repeated. `disableReplay`

- Autonomous transactions
- DBMS_ALERT calls - email or other notifications
- DBMS_FILE_TRANSFER calls - copying files
- DBMS_PIPE and RPC calls - to external sources
- UTL_FILE calls - writing text files
- UTL_HTTP calls - making HTTP callouts
- UTL_MAIL calls - sending email
- UTL_SMTP calls - sending SMTP messages
- UTL_TCP calls - sending TCP messages
- UTL_URL calls - accessing URLs

Application Continuity

No replay when:

- Time allowed for starting replay exceeded
- Application uses a restricted call
- Replay has been explicitly disabled
  - using the `disableReplay` API
- Session is killed or disconnected with `noreplay` keyword
- Request issues ALTER SYSTEM / DATABASE

Application Continuity Configuration

Database:

- FAN configured with ONS
- Application service properties:
  - FAILOVER_TYPE
  - REPLAY_INITIATION_TIMEOUT
  - FAILOVER_RETRIES
  - FAILOVER_DELAY

```
srvctl add service ...
-failovertype TRANSACTION
-replay_init_time 1800
-failoverretry 30
-failoverdelay 10
-commit_outcome TRUE
-retention 86400
-notification TRUE
```

MOS 1602233.1: How To Test Application Continuity Using A Standalone Java Program

Application Continuity Configuration

JDBC:

- Data source: `oracle.jdbc.replay.OracleDataSourceImpl`
- Connection string:

```
jdbc:oracle:thin:@(DESCRIPTION =
  (TRANSPORT_CONNECT_TIMEOUT=60)
  (CONNECT_TIMEOUT=60)(RETRY_COUNT=3)
  (FAILOVER=ON)(ADDRESS_LIST =(ADDRESS=
  (PROTOCOL=tcp)(HOST=actest-scan.cern.ch)
  (PORT=1521))(CONNECT_DATA=(SERVICE_NAME=ACTEST)))
```

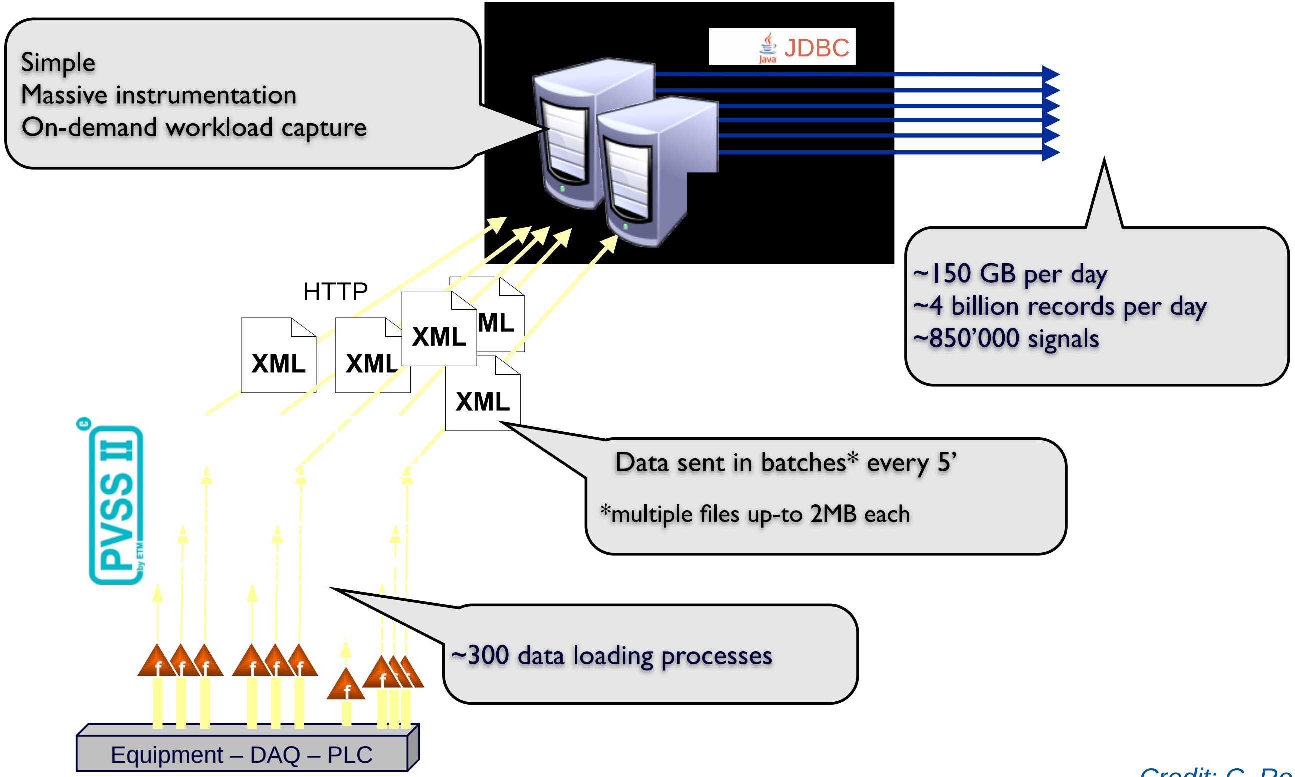
Agenda

- About CERN
- Transaction Guard
- Application Continuity
- CERN’s experience

Challenges

- Criticality of database
  - applications for the accelerator complex
  - no maintenance windows for the experiments
- Reduce visibility of
  - planned interventions (service move during patching)
  - un-planned interruptions (hardware failure etc.)

Application Continuity - application



Simple
Massive instrumentation
On-demand workload capture

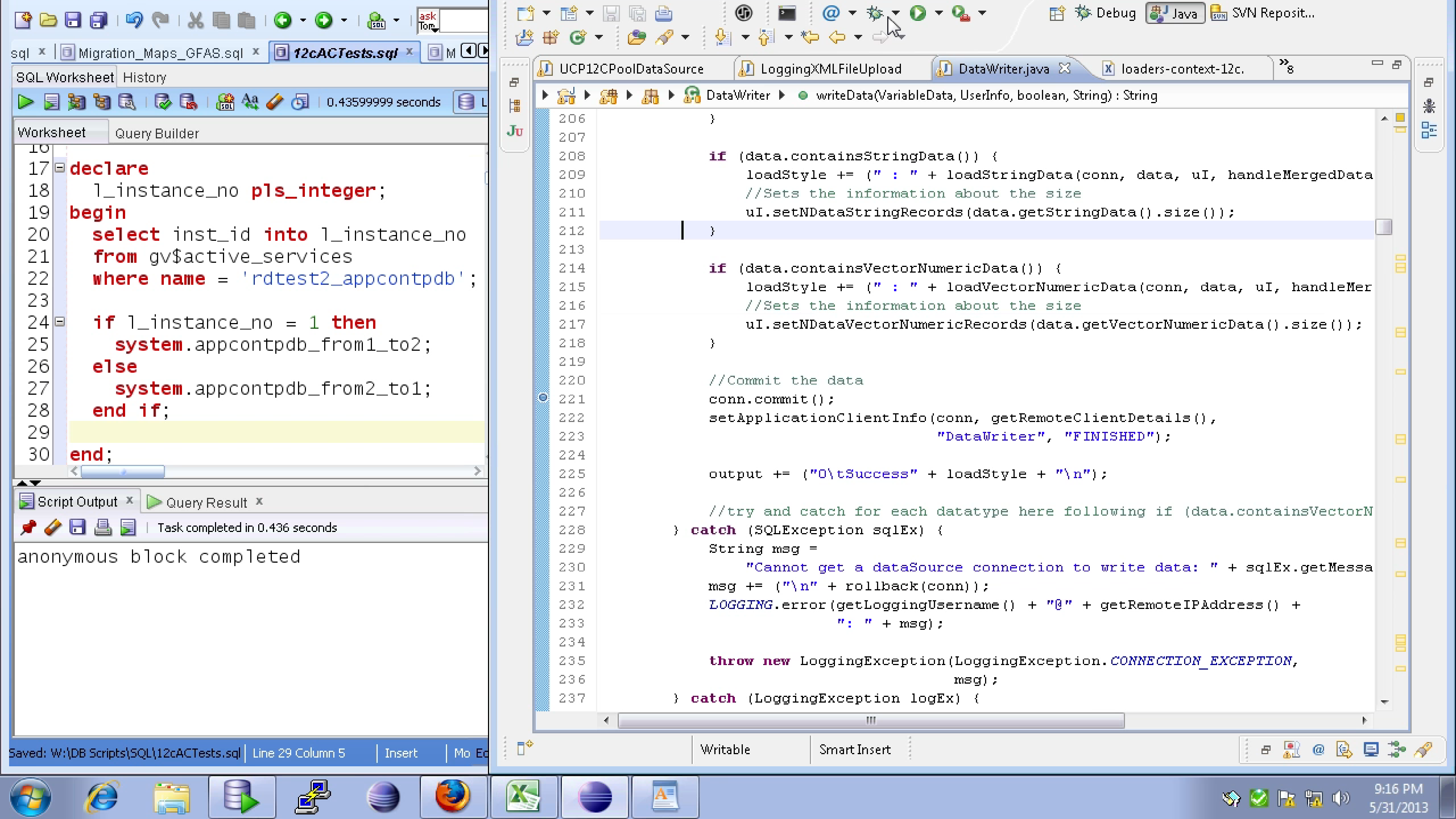
~150 GB per day
~4 billion records per day
~850'000 signals

Data sent in batches* every 5'

*multiple files up-to 2MB each

~300 data loading processes

Credit: C. Roderick



```
declare
  l_instance_no pls_integer;
begin
  select inst_id into l_instance_no
  from gv$active_services
  where name = 'rdtest2_appcontpdb';

  if l_instance_no = 1 then
    system.appcontpdb_from1_to2;
  else
    system.appcontpdb_from2_to1;
  end if;

end;
```

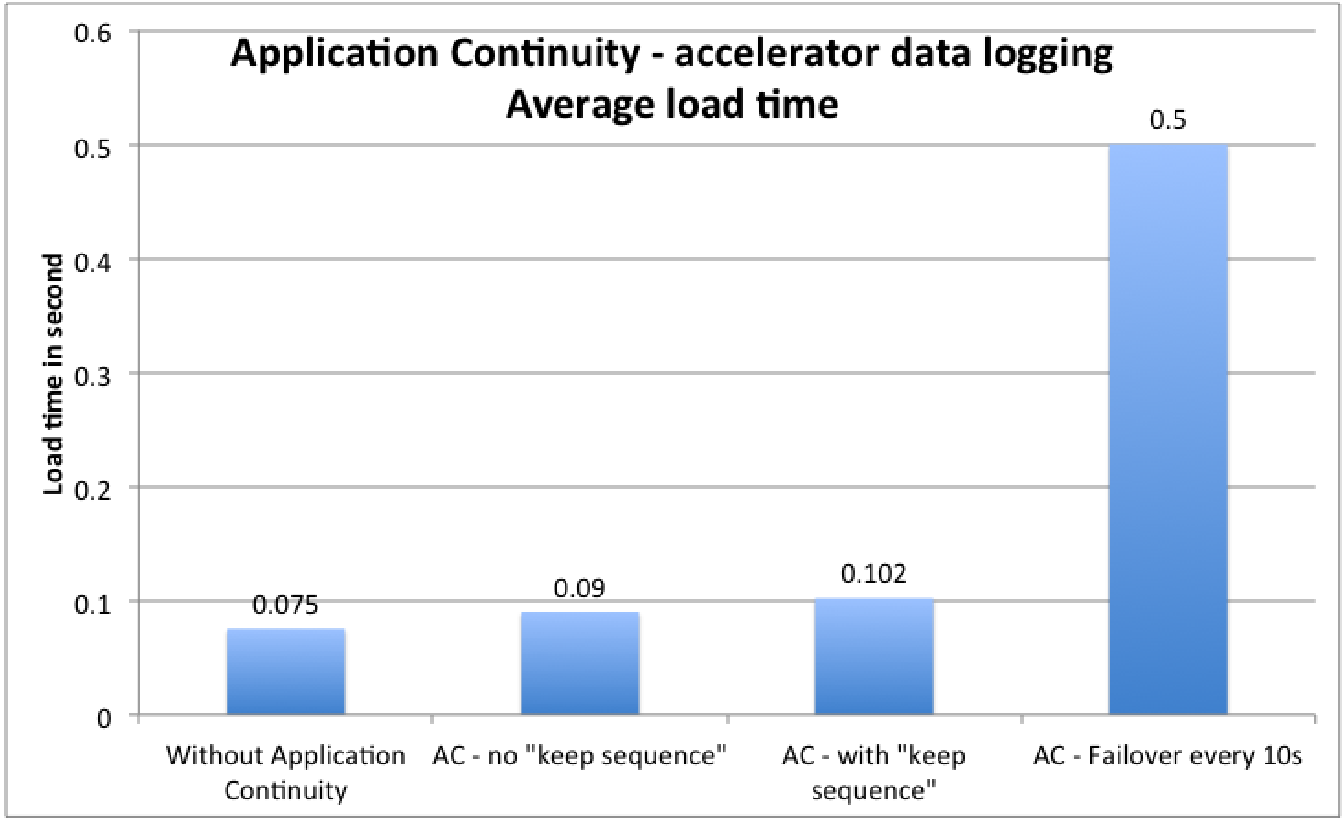
Task completed in 0.436 seconds

```
anonymous block completed
```

Saved: W:\DB Scripts\SQL\12cACTests.sql | Line 29 Column 5 | Insert | Mo Ed

```
          }

          if (data.containsStringData()) {
            loadStyle += (" : " + loadStringData(conn, data, uI, handleMergedData
            //Sets the information about the size
            uI.setNDataStringRecords(data.getStringData().size());
          }

          if (data.containsVectorNumericData()) {
            loadStyle += (" : " + loadVectorNumericData(conn, data, uI, handleMer
            //Sets the information about the size
            uI.setNDataVectorNumericRecords(data.getVectorNumericData().size());
          }

          //Commit the data
          conn.commit();
          setApplicationClientInfo(conn, getRemoteClientDetails(),
                                   "DataWriter", "FINISHED");

          output += ("0\tSuccess" + loadStyle + "\n");

          //try and catch for each datatype here following if (data.containsVectorN
        } catch (SQLException sqlEx) {
          String msg =
            "Cannot get a dataSource connection to write data: " + sqlEx.getMessa
          msg += ("\n" + rollback(conn));
          LOGGING.error(getLoggingUsername() + "@" + getRemoteIPAddress() +
                        ": " + msg);

          throw new LoggingException(LoggingException.CONNECTION_EXCEPTION,
                                     msg);
        } catch (LoggingException logEx) {
```

Application Continuity - results



Credit: C. Roderick, L. Canali, A. Dumitru

Benefits

- User experience vastly improved
- At-most-once-execution
- Reliable and durable commit outcome
- No ambiguous errors displayed by application
- Native solution to handle idempotence
- Mask many outages in a safe way

Conclusions

- Increase applications HA with Transaction Guard and Application Continuity
- Better handle planned/unplanned outages
- Known and durable outcome of transactions
- Transparent replay of submitted work

Acknowledgements

The work presented here on behalf of the:

- CERN Controls Group – *Chris Roderick*
- CERN Database Group – *Luca Canali*
- CERN openlab – *Andrei Dumitru*

In collaboration with the Application Continuity team

CERN

www.cern.ch

References

Docs:

- *Oracle Database Development Guide12c Release 1 (12.1)*
- *Transaction Guard with Oracle Database 12c, Oracle White Paper*, June 2013
- *Application Continuity with Oracle Database 12c*, Oracle White Paper,
- June 2013
- *Oracle database 12c Application Continuity and Transaction Guard*, Kuassi Mensah, Oracle Open World
- *Maximize Availability by Using Database Services with Oracle RAC*, Carol Colrain, Oracle Open World

Images:

- *Transaction Guard with Oracle Database 12c whitepaper*
- *Application Continuity with Oracle Database 12c, whitepaper*
- *http://www.grokdotcom.com*