In-Flight PLSQL Deployment: Issues & Solutions

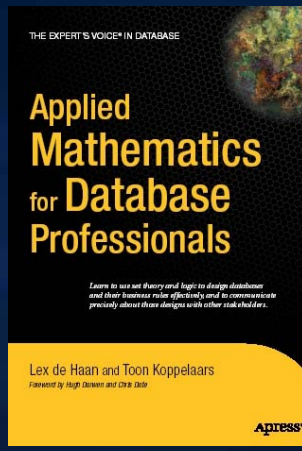
Toon Koppelaars
www.RuleGen.com

OGH: DBA-Dag High Availability
3 November 2009

Who am I?

- Oracle technology since 1987
- Developer, DBA team lead, software architect
- Focus
  - Using DBMS for what it was designed for
  - …
  - …
- OakTable member

- Authored this book with Lex de Haan ➔

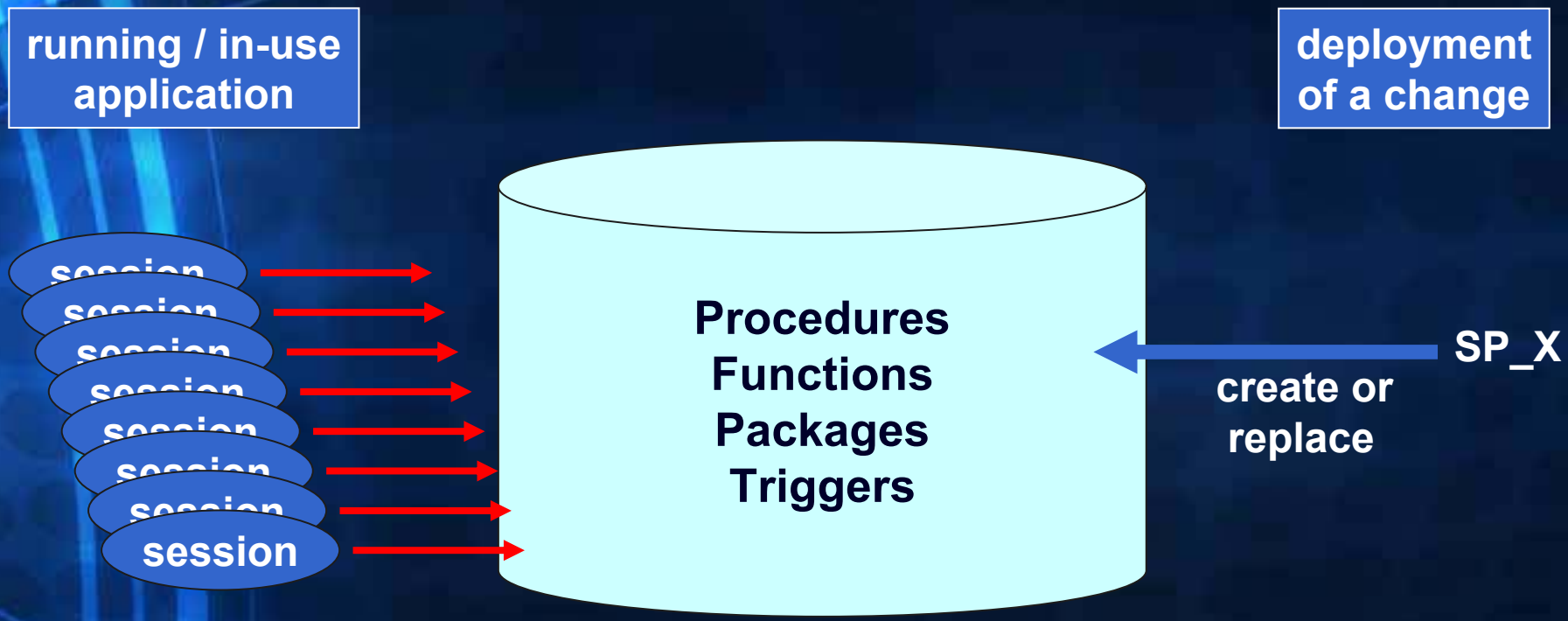


- Frequent presenter
- TheHelsinkiDeclaration.blogspot.com
-

Overview

- What is IFPLSQLD
- Concepts
  - Invalidation tree
  - Run-lock tree
- The Main Issue
- Solutions
  - Break invalidation tree
  - Break run-lock tree
- Package state loss
- Oracle Editions
- Wrap up

In-Flight PLSQL Deployment

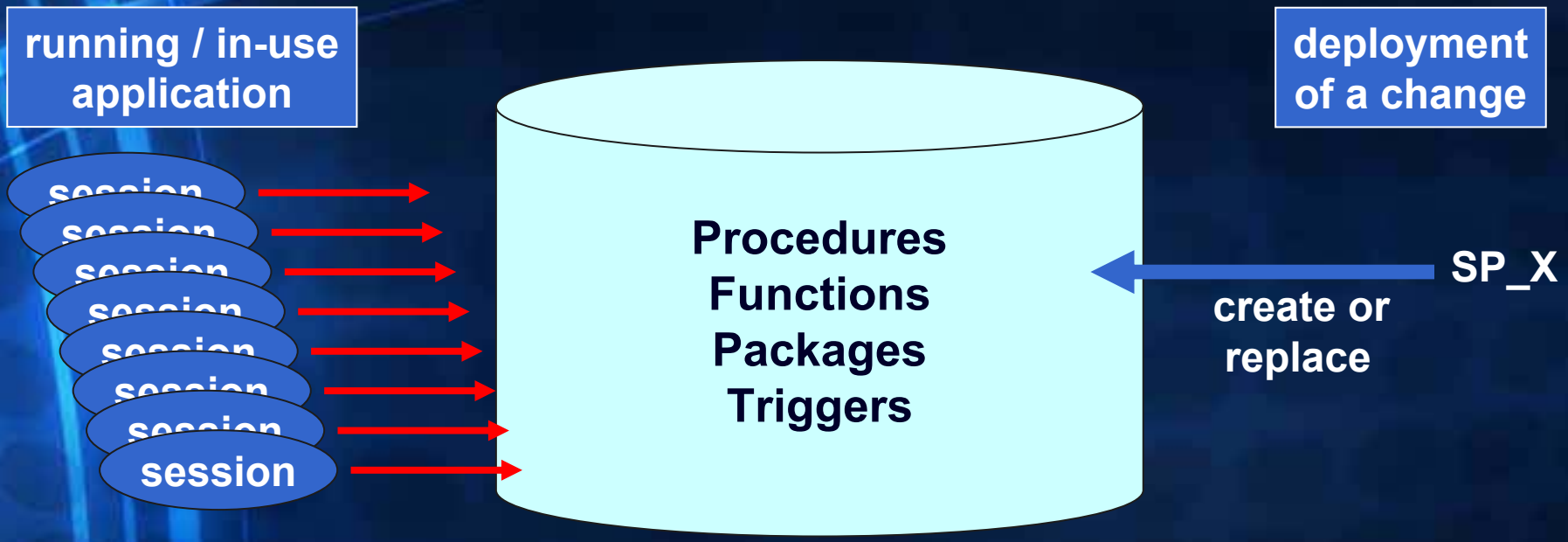


Why?

- Todays subject: high availability
  - 52 * 7 * 24

- "(Business) Change is the only constant" ➔ code changes
- "Bugfree software doesn't exist" ➔ code changes (hot fixes)

In-Flight PLSQL Deployment



- What *code architecture measures* can we take to enable IFPLSQLD as much as possible?
  - Scope: prc, fun, pck, trg

Assumption

- Custom development environment
- The code change has been tested
- And is therefor safe to be deployed
  - In-flight

Concepts: Dependencies

Concepts: Dependencies

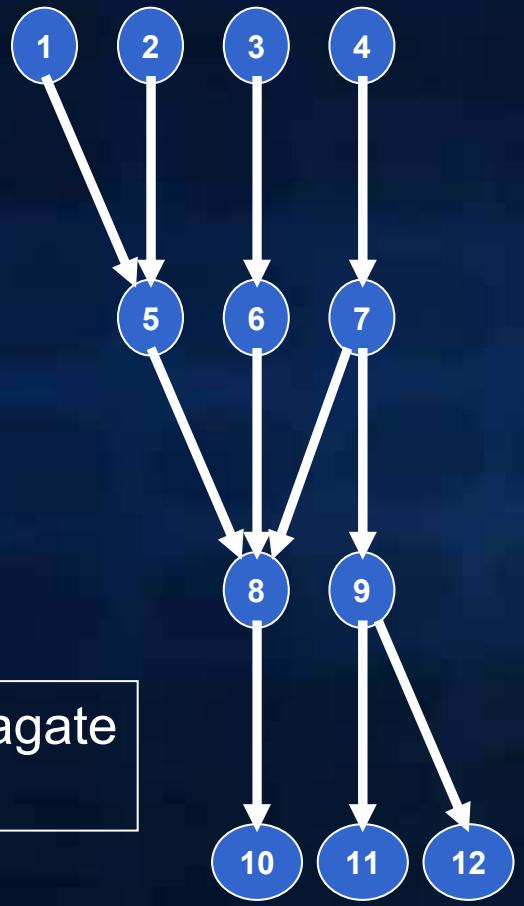
- DBMS maintains a “dependency tree” of stored plsql objects
  - It knows what object depends on what other objects: dba_dependencies
- Deploying an object will cause invalidation of all objects upwards in this tree
- Automatic recompilation

demo01a,b,c.sql

Concepts: Dependencies

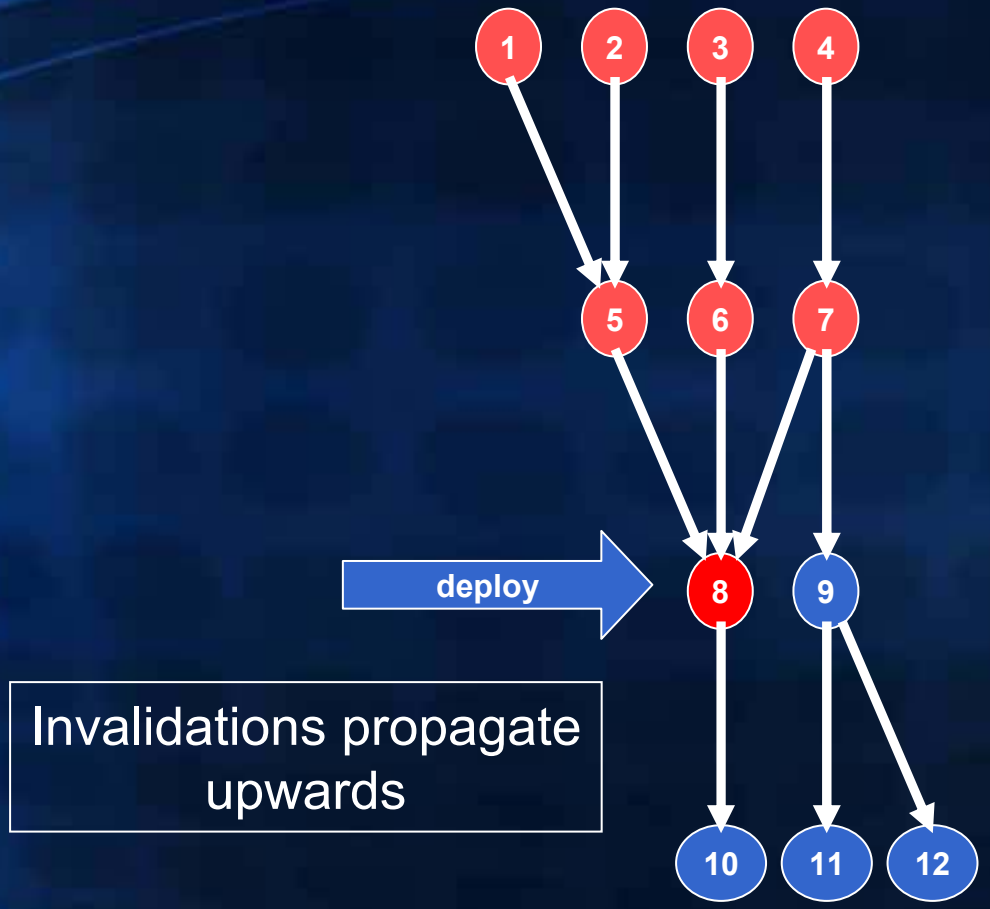
- 11G
  - Invalidate upwards → only if signature changes (or object itself is invalid)
- 10G and earlier
  - Invalidate upwards → always

Concepts: Dependencies



Invalidations propagate upwards

Concepts: Dependencies



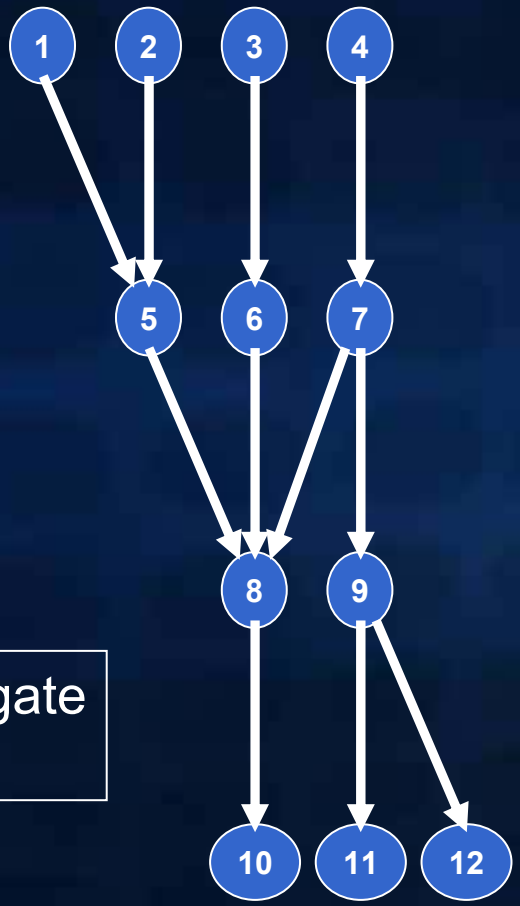
Invalidations propagate upwards

Concepts: Run Locks

Concepts: Run Locks

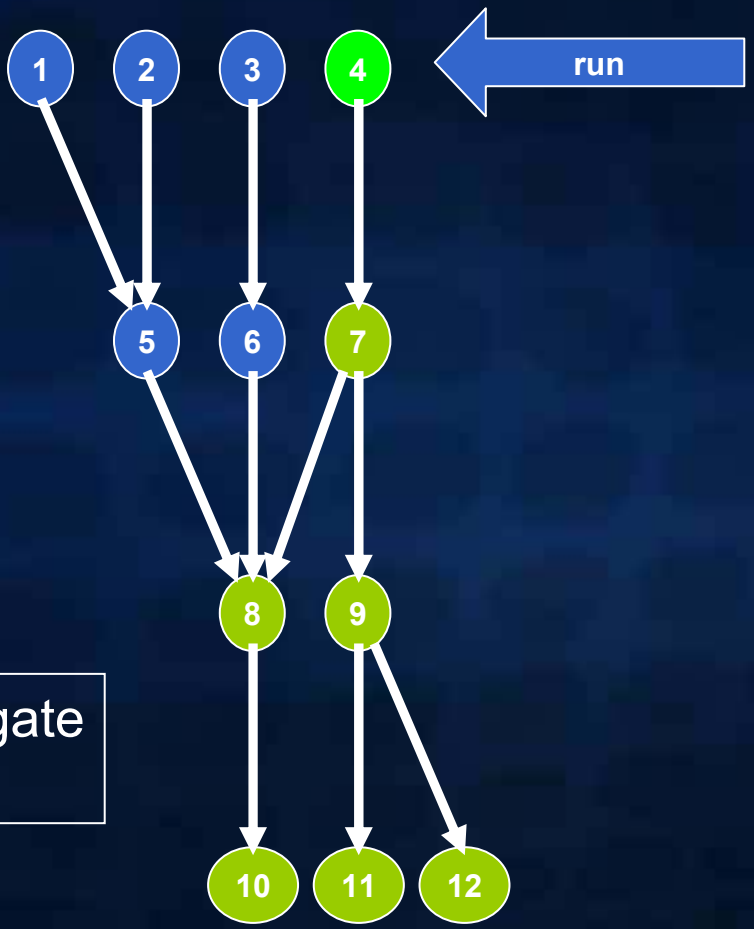
- Object definitions are available in shared-pool
  - These definitions are “locked” by sessions that *currently* depend on them
    - i.e. that run their code
    - Multiple sessions can share such lock
  - Sessions that want to change (or invalidate) these definitions require an exclusive lock

Concepts: Run Locks



Run-locks propagate downwards

Concepts: Run Locks



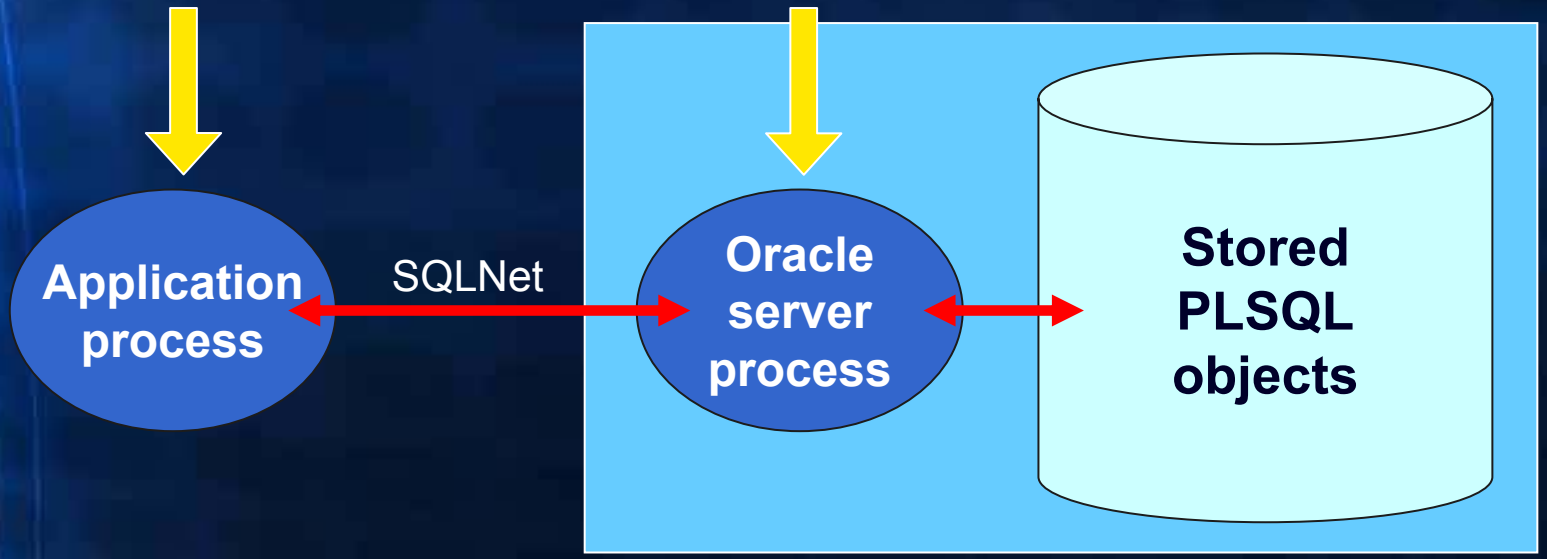
Run-locks propagate downwards

demo03a.sql

Concepts: CPU-counter Location and Run Locks

- At any time, "CPU-counter" resides either inside DBMS, or outside DBMS, i.e. inside client program

Forms
ADF-BC
Java/jdbc
...
sqlplus



Run-locks only exist when the CPU-counter is inside the DBMS

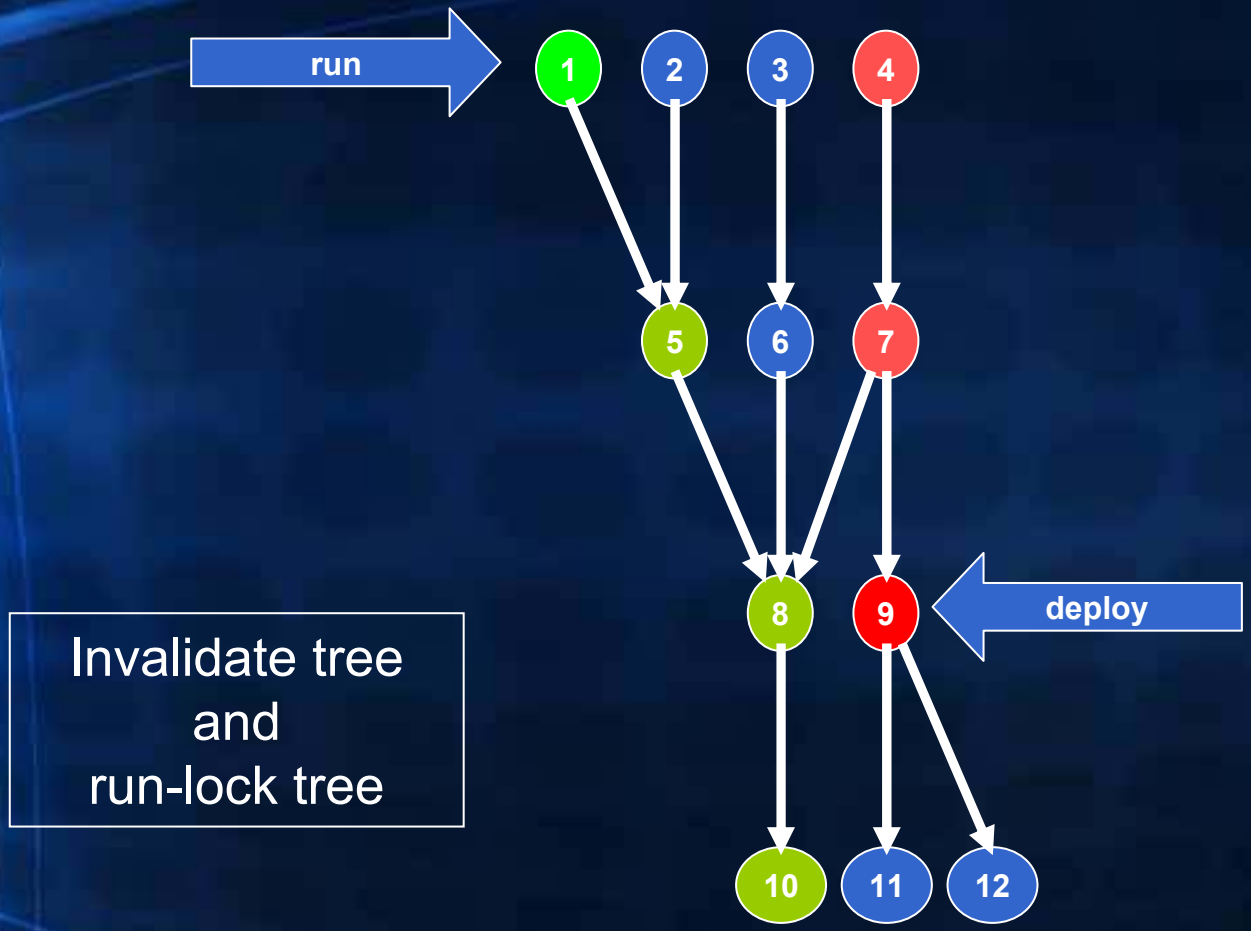
Concepts: CPU-counter Location and Run Locks

- Run-lock trees come and go
  - Demo00a.sql

Main Issue

- If
  ({ object | object in some
  run-lock tree }
  INTERSECT
  { object | object in current
  invalidation tree })
  =
  not empty
  Then
  IFPLSQLD is blocked

In-Flight Deployment



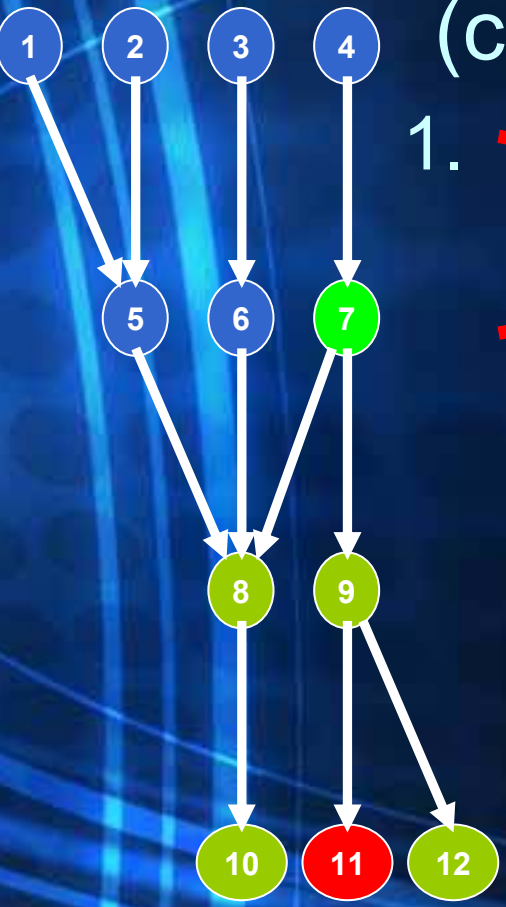
Invalidate tree
and
run-lock tree

Solution to Enable IFPLSQLD

- Prevent a non-empty intersection (collisions) by:
  1. Minimizing upward invalidation tree that is required for the deployment
  2. Minimizing (duration of) all current downward run-lock trees
     - "duration of" ➔ frequently releasing them (for a very short time)

Solution

- Prevent a non-empty intersection (collision) by:
  1. ~~Minimizing the upward invalidation tree that is required for the deployment~~



Deployment of Object 11 does not invalidate object 9, however since object 9 is in the run-lock tree and 9 depends on 11, 11 will also be in the run-lock tree.

➔ So only helpful to minimize recompile time

Minimizing Upward Invalidation Tree

- Packages break the dependency tree

Body of PCK3 can be changed, without Invalidating SP2 and SP1

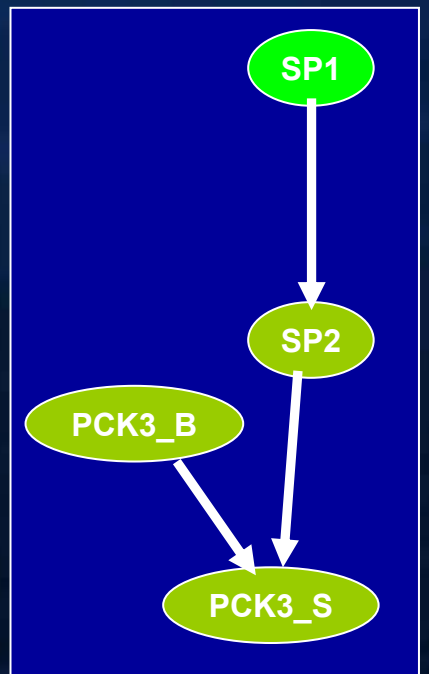
demo02a.sql

Minimizing Recompilation of Invalidated Objects

- Packages break dependency tree
  - ➔ which stops upwards invalidation

Note: packages do not stop downwards run-lock tree

- Demo02b.sql



Minimizing Upward Invalidation Tree

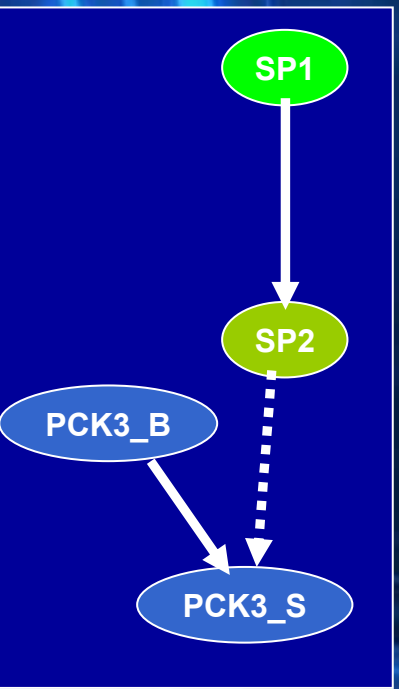
- 11G helps us a lot
  - Fine grained dependency tracking
    - Focusses on table modifications
  - "Signature-change-only" invalidations
  - Less upward invalidations

➔ Less stuff to (auto) recompile after deployment

➔Less "downtime"

Minimizing downward run-lock tree

- Native Dynamic SQL (NDS) to the rescue
  - Demo04a.sql

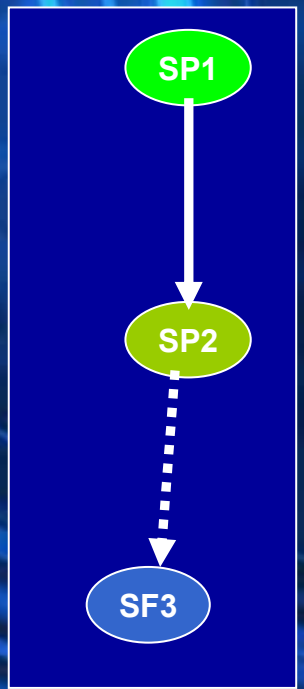


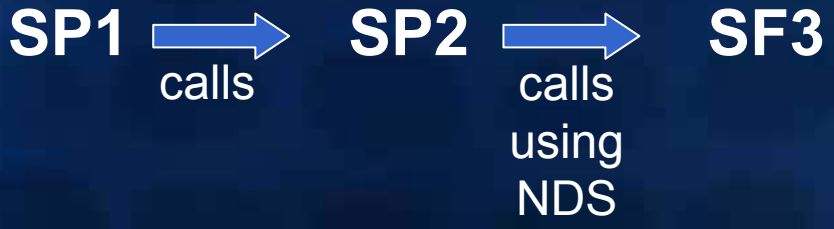
SP1 → calls → **SP2** → calls using NDS → **PCK3.SF3**

PCK3_S and PCK3_B included in run-lock tree only when CPU-counter is in PCK3_B

Minimizing downward run-lock tree

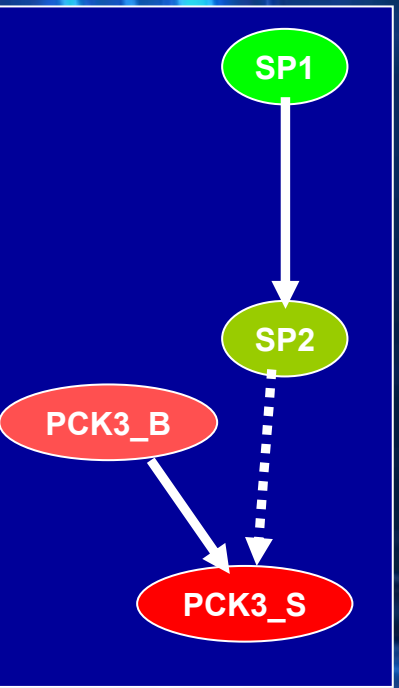
Note: function SF3 might as well have been standalone instead of packaged for this demo





Minimizing downward run-lock tree

- By the way...
  - NDS also minimizes upward invalidation tree!
  - Irrespective of SF3 being packaged or not



Minimizing downward run-lock tree

- NDS creates a client-side program inside the DBMS
  - Block with NDS-call = client
  - When “CPU-counter” is (back) in client ➔ run-lock tree of NDS-call is non-existent
- Educated use of NDS within processes that are always inside DBMS
  - Eg. batchjobs often have some top-level loop
  - Inside loop use NDS to call code

Other Common Issue: Package State Loss

ORA-04068: existing state of packages has been discarded

- Demo05a.sql
- Let's just "handle the exception"
  - Demo05b.sql
  - Demo05c.sql

Package State Loss

- Solutions
  1. Introduce dedicated state-only packages
     - These hardly ever require deployment
  2. Use PRAGMA Serially_Reusable

State Only Package

- Demo06a.sql
- Initialized globals do not get re-initialized

PRAGMA Serially_Reusable

- Demo06b.sql
- Use of this PRAGMA:
  - Depends on functionality of init-code for global(s)
  - Initialized globals get re-initialized

11Gr2: Editions

- Promise to enable IFPLSQLD
  - Create a new edition
  - Deploy objects in new edition
  - Switch users to new edition
- Question:
  - Can we deploy objects in new edition without being blocked by run-lock trees?

11Gr2: Editions

- Demo07a.sql
  - Recreates starting point
- Demo07b.sql
  - Deploys SF3 in new edition

11Gr2: Editions

- We can deploy objects without being blocked by run-lock tree
  - Auto recompile happens on first execution
    - All objects in invalidation tree are copied to new edition too
    - Allthough these were not changed
    - Reason?
  - Still need to minimize invalidation tree?
    - No ➔ forced (manual) recompile can be done as part of edition build

11Gr2: Editions

- My guess:
  - This feature will eventually enable true in-flight Oracle patch-installs and upgrades
- But first:
  - For us to do the necessary bug hunting
  - ☺

Wrapping Up

- IFPLSQLD is blocked if run-lock tree and invalidation tree have objects in common
  - NDS can be a lifesaver to enable IFPLSQLD
- 11G + packages minimize object invalidations ➔ less downtime
- Package state loss can be fixed:
  - with state-only packages or,
  - serially_reusable
- 11Gr2 Editions feature likely to be the ultimate IFPLSQLD enabler

Questions?

Toon@RuleGen.com