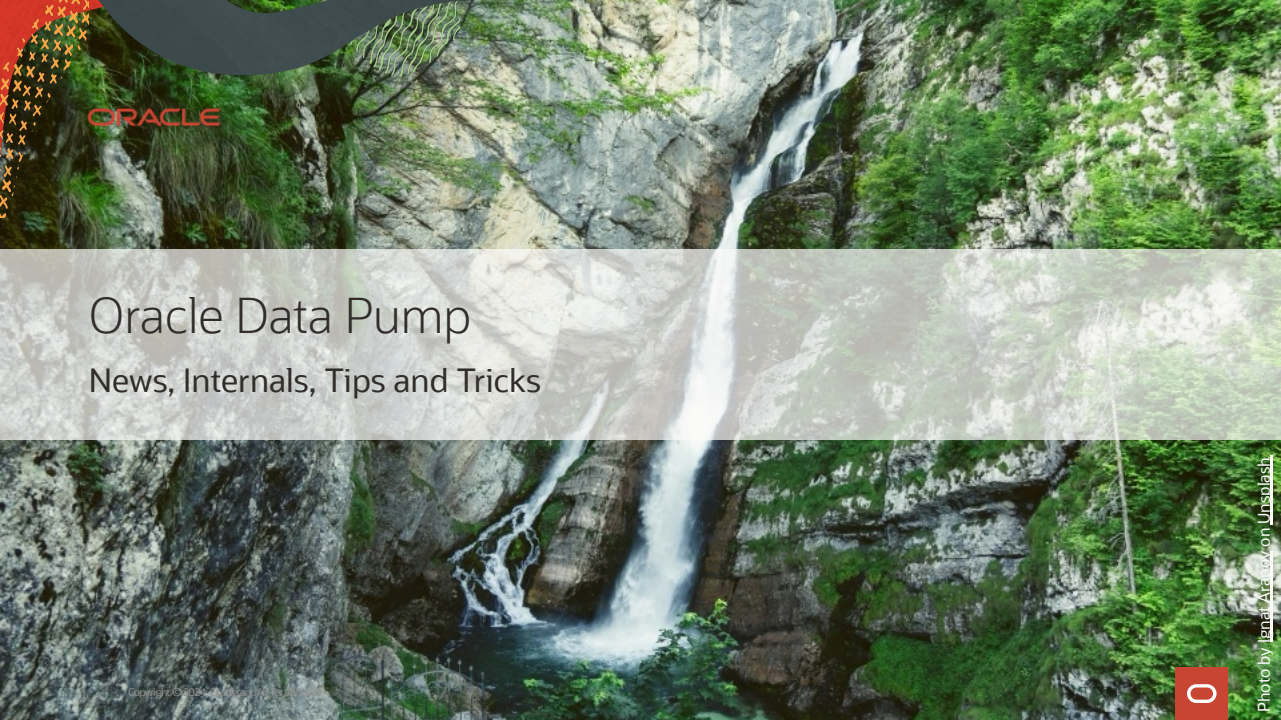


Oracle Data Pump

News, Internals, Tips and Tricks

Photo by Jonas Arnaldo on Unsplash

Daniel Overby Hansen

Senior Principal Product Manager

dohdatabase

@dohdatabase

https://dohdatabase.com

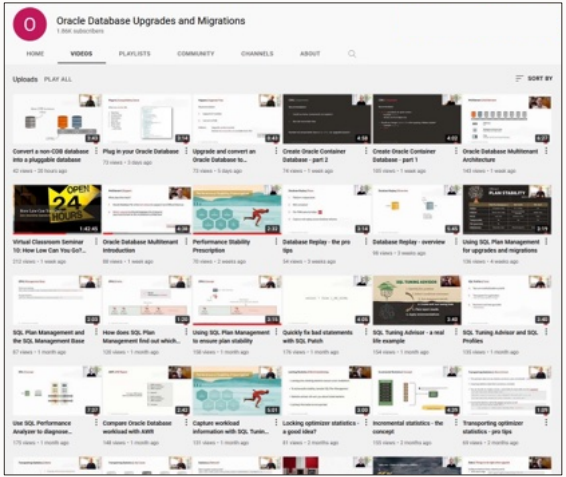
Web Seminar	Slides
Episode 16 (replaces Episode 1 from Feb 2021) Oracle Database Release and Patching Strategy for 19c and 23c *115 minutes – May 10, 2023*	
Episode 17 From SR to Patch – Insights into the Oracle Database Development process *55 minutes – June 22, 2023*	
NEW Episode 18 Cross Platform Migration – Transportable Tablespaces to the Extreme *145 min – February 22, 2024*	
Episode 2 AutoUpgrade to Oracle Database 19c *115 minutes – Feb 20, 2021*	
Episode 3 Performance Stability, Tips and Tricks and Underscores *120 minutes – Mar 4, 2021*	
Episode 4 Migration to Oracle Multitenant	

Recorded Web Seminars

https://MikeDietrichDE.com/videos

More than 35 hours of technical content, on-demand, anytime, anywhere

YouTube | Oracle Database Upgrades and Migrations



- 300+ videos
- New videos every week
- No marketing
- No buzzword
- All tech

SCAN ME

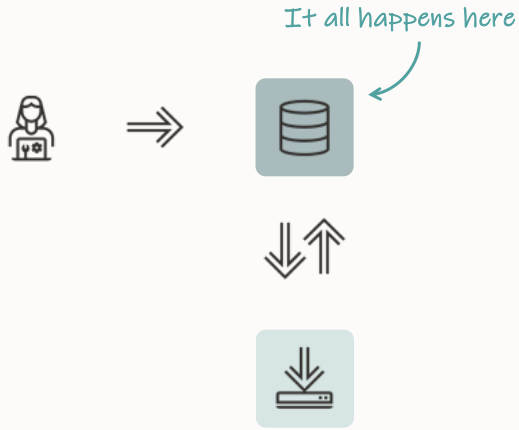
Internals

Architecture

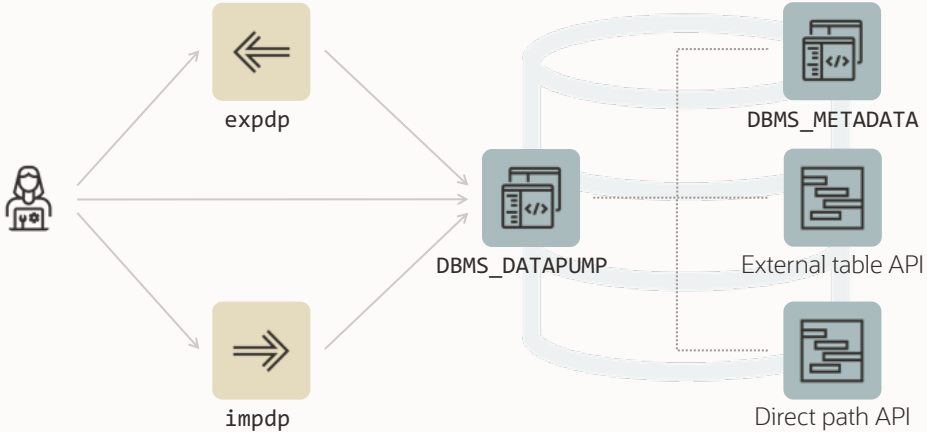
Architecture

Data Pump is server-based,

not client-based



Architecture



You can use **DBMS_DATAPUMP**

Data Pump API

1. Enable SQL trace on a test database

```
SQL> alter system set event='10046 trace name context forever, level 4';
```

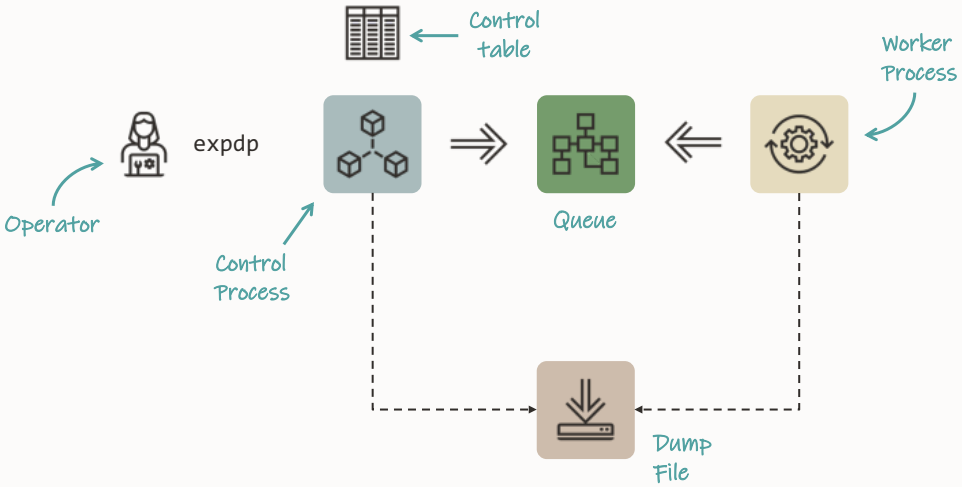
2. Execute your Data Pump command

```
$ impdp system ... parfile=import.par
```

3. Examine the trace file

```
$ vi ORCL_ora_12345.trc
```

Architecture



Architecture

```
$ expdp dpuser/oracle schemas=app keep_master=y
```


Architecture

```
SQL> select name, value_t from dpuser.sys_export_schema_01;

NAME                 VALUE_T
-------------------- --------------------------------------------
SYS_EXPORT_SCHEMA_01 DB19.LOCALDOMAIN
LOG_FILE_DIRECTORY   DATA_PUMP_DIR
LOG_FILE_NAME        export.log
CLIENT_COMMAND       dpuser/******** schemas=app keep_master=y
SCHEMA_LIST          'APP'
SCHEMA_EXPR          IN ('APP')
COMPRESSION          METADATA_ONLY
COMPRESSION_ALGORITHM BASIC
DATA_ACCESS_METHOD   AUTOMATIC
.
.
.
```

Unloading and Loading

Data Files

Used for transportable tablespace

Only metadata is unloaded into/loaded from dumpfile

Data remains in data files

Direct Path

External Tables

Insert as Select

Conventional Path

Pro tip: Cross-endian data migration requires data files are converted

Unloading and Loading

Data Files

Direct Path

External Tables

Insert as Select

Conventional Path

Unloads from / load into data files directly

Circumvents SQL layer

Fast

Not usable in all situations

Pro tip: Data Pump automatically selects the best unload/load method

Unloading and Loading

Data Files

Direct Path

External Tables

- Use SQL layer to unload to / load from external table
- Can use `APPEND` hint for faster load
- Very good parallel capabilities
- Dump file format similar to direct path

Insert as Select

Conventional Path

Pro tip: Data unloaded with Data Pump is not compatible with a regular external table (`CREATE TABLE ... ORGANIZATION EXTERNAL ...`)

Unloading and Loading

Data Files

Direct Path

External Tables

Insert as Select

- Used by network link imports only
- Will disable use of direct path

Conventional Path

- Not very common

Unloading and Loading

Data Files

Direct Path

External Tables

Insert as Select

Conventional Path

- Used as last resort
- Slower
- Import only

Unloading and Loading

```
$ expdp dpuser/oracle schemas=app metrics=y
```


Metadata

A category of metadata is described by an object path

Examples:

```
TABLE
TABLE/INDEX
TABLE/STATISTICS/TABLE_STATISTICS
TABLE/TRIGGER
```

You can get a full list of object paths from these views:

```
DATABASE_EXPORT_OBJECTS
SCHEMA_EXPORT_OBJECTS
TABLE_EXPORT_OBJECTS
```


Metadata

```
SQL> select object_path, comments from schema_export_objects;

OBJECT_PATH                    COMMENTS
------------------------------ --------------------------------------------------
ALTER_FUNCTION                 Recompile functions
ALTER_PACKAGE_SPEC             Recompile package specifications
ALTER_PROCEDURE                Recompile procedures
ANALYTIC_VIEW                  Analytic Views
AQ                             Advanced Queuing
ASSOCIATION                    Statistics type associations
ATTRIBUTE_DIMENSION            Attribute Dimensions
AUDIT_OBJ                      Object audits on the selected tables
CLUSTER                        Clusters in the selected schemas and their indexes
CLUSTERING                     Table clustering
.
.
.
```

Use `INCLUDE` or `EXCLUDE` to add or remove a specific category of metadata

- Mutually exclusive

Combine `INCLUDE` or `EXCLUDE`

- Mutually exclusive

Metadata | Dependencies

Example: Excluding a table will also exclude

- Indexes
- Constraints
- Grants
- Triggers
- And the like upon that table

Example: Excluding an index will also exclude

- Statistics on that index

To migrate your data, you typically use Data Pump in schema or full mode

SCHEMA

Individual schemas and what they own

FULL

All schemas plus
more or less everything in the database

Move | Full Export

Objects exported only in full export:

- Audit trail and policies
- Database Vault
- Directories
- Profiles and password verify function
- Public database links
- Public synonyms
- Roles
- SQL Management Objects (plan histories, SQL plan baselines, SQL profiles, etc.)
- Tablespaces
- Users (other than those specified in `SCHEMAS` parameter)
- Workspace manager (for schema export you need to use `DBMS_WM.Export_Schemas`)

...

Data Pump never exports grants on SYS objects

- Not even in a full export
- Add them manually following the import

Tips & Tricks

Data Pump Best Practices

Apply Data Pump Bundle Patch

- Oracle Database 19c

Release Updates rarely contain Data Pump fixes

Data Pump Bundle Patch contains almost 200 functional and performance fixes

- Data Pump Recommended Proactive Patches For 19.10 and Above (Doc ID 2819284.1)

Importing a complete application with data drops from almost 2.5 hours to 48 minutes – by just applying the Data Pump bundle patch

A global provider of financial services

Always convert to SecureFile LOBs

A short history of **binary data types**

v4	LONG and LONG RAW
8.0	CLOB and BLOB
11g	SecureFile LOBs

v4	LONG and LONG RAW
8.0	BasicFile LOBs
11g	SecureFile LOBs

```
--How to find columns using the legacy BasicFile LOBs

select * from dba_lobs where securefile='NO'
```


Legacy binary data types should have been migrated to SecureFile LOBs

```
--Converting a BasicFile LOB to SecureFile during import,
--is faster than not converting it.
--Overview of Oracle LOBs (Doc ID: 1490228.1)

impdp ... transform=lob_storage:securefile
```


SecureFile LOBs

Importing as BasicFiles

```
... imported "SCHEMA"."TABLE"      31.83 GB  681025 rows in 804 seconds using direct_path
```

Importing as SecureFiles

```
... imported "SCHEMA"."TABLE"     31.83 GB  681025 rows in 261 seconds using external_table
```


If exporting SecureFile LOBs is slow, apply 19.23.0 Data Pump Bundle Patch

- Alternatively, trick Data Pump with fake stats

Boost performance even more with partitioning

Do you still have BasicFile LOBs?

- Use DIY parallelism during export
- Be sure to convert to SecureFile LOB on import

Use parallel and multiple dump files

```
--Apply parallelism by simply specifying a degree
expdp ... parallel=8


--Use different parallel degree on import
impdp ... parallel=32
```


```
--Use %L to allow multiple dump files
expdp ... parallel=8 dumpfile=exp%L.dmp
```


```
--Split dump files into minor files for easier transport
expdp ... parallel=8 dumpfile=exp%L.dmp filesize=10G
```


```
-- After export, store a checksum in the dump file.
-- Detects in-flight corruption or alteration.
-- Specify other algorithms using checksum_algorithm parameter.

expdp ... checksum=yes

impdp ... verify_checksum=yes
          verify_only=yes
```


Transportable jobs can use parallel in Oracle Database 21c

```
-- Any transportable jobs can now run in parallel
-- Parallel unload/load of metadata provide a significant performance boost

expdp ... full=y transportable=always parallel=16
expdp ... tablespace=<list> parallel=16

impdp ... parallel=16
```

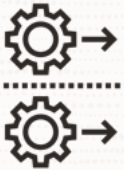

Parallel Transportable | Benchmark

Oracle E-Business Suite database

600.000+ objects

Export parallel 1	2h 2m
Import parallel 1	6h 44m
Total	**8h 46m**

Export parallel 16	1h 8m
Import parallel 16	1h 23m
Total	**2h 31m**



Set parallel to 2 x physical cores or number of OCPUs

Always add diagnostics to your log file

```
--Add more diagnostics information to your log file

expdp ... metrics=yes logtime=all
impdp ... metrics=yes logtime=all
```


Specify a job name and attach to the job

```
$ expdp system ... job_name=APPS_STUFF

Export: Release 23.0.0.0.0 - Production on Fri May 3 14:56:06 2024
Version 23.4.0.24.05

Copyright (c) 1982, 2024, Oracle and/or its affiliates.  All rights reserved.
Connected to: Oracle Database 23ai EE High Perf Release 23.0.0.0.0 - Production
03-MAY-24 14:56:13.420: Starting "SYSTEM"."APPS_STUFF"
03-MAY-24 14:56:13.799: W-1 Startup on instance 1 took 0 seconds
03-MAY-24 14:56:30.550: W-2 Startup on instance 1 took 0 seconds
03-MAY-24 14:56:38.519: W-3 Startup on instance 1 took 0 seconds
03-MAY-24 14:56:38.529: W-4 Startup on instance 1 took 0 seconds
```

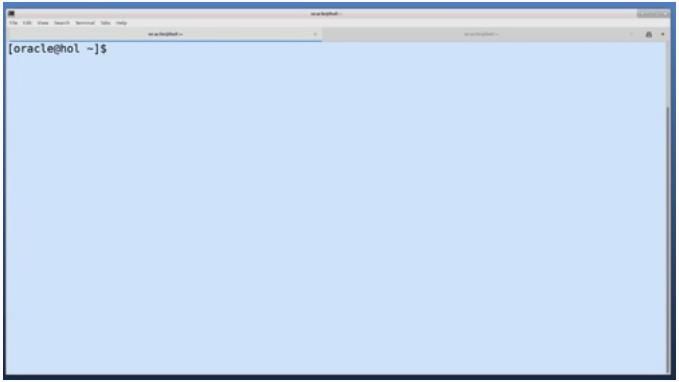
```
$ expdp ... attach=SYSTEM.APPS_STUFF

Export> status

...

Worker 1 Status:
  Instance ID: 1
  Instance name: CDB23
  Host name: dbs23
  Object start time: Friday, 3 May, 2024 15:22:30
  Object status at: Friday, 3 May, 2024 15:30:35
  Process Name: DW00
  State: EXECUTING
  Object Schema: APPS
  Object Name: AP_INVOICE_DISTRIBUTIONS_PKG
  Object Type: DATABASE_EXPORT/SCHEMA/PACKAGE_BODIES/PACKAGE/PACKAGE_BODY
  Completed Objects: 1,938
  Worker Parallelism: 1
```

Interactive Command Mode | Demo



Watch on YouTube

Tracing Data Pump

Ensure dictionary and fixed objects statistics are accurate

- Before export
- Before import
- Immediately after import

```
begin
    --dbms_stats.gather_dictionary_stats;
    dbms_stats.gather_schema_stats('SYS');
    dbms_stats.gather_schema_stats('SYSTEM');
    dbms_stats.gather_fixed_objects_stats;
end;
/
```


```
begin
   --dbms_stats.gather_dictionary_stats;
   dbms_stats.gather_schema_stats('SYS');
   dbms_stats.gather_schema_stats('SYSTEM');
   dbms_stats.gather_fixed_objects_stats;
end;
/
```

"After gathering dictionary stats, our Data Pump export went from 46 to 8 minutes"

```
-- Change AWR snap interval to 15 minutes and create snapshot
exec dbms_workload_repository.modify_snapshot_settings(null, 15);
exec dbms_workload_repository.create_snapshot;

-- Optionally, enable SQL trace for Data Pump processes or specific SQL ID
alter system set events 'sql_trace {process: pname = dw | process: pname = dm} level=8';
alter system set events 'sql_trace[SQL: 03g1bnw08m4ds ]';

-- Run Data Pump job with trace (Doc ID 286496.1)
expdp ... metrics=yes logtime=all trace=1FF0300
impdp ... metrics=yes logtime =all trace=1FF0300

-- Create AWR snapshot and produce AWR report
exec dbms_workload_repository.modify_snapshot_settings(null, <original-value>);
exec dbms_workload_repository.create_snapshot;
@?/rdbms/admin/awrrpt
```

```
-- Change AWR snap interval to 15 minutes and create snapshot
exec dbms_workload_repository.modify_snapshot_settings(null, 15);
exec dbms_workload_repository.create_snapshot;

-- Optionally, enable SQL trace for Data Pump processes or specific SQL ID
alter system set events 'sql_trace {process: pname = dw | process: pname = dm} level=8';
alter system set events 'sql_trace[SQL: 03g1bnw08m4ds ]';

-- Run Data Pump job with trace (Doc ID 286496.1)
expdp ... metrics=yes logtime =all trace=1FF0300
impdp ... metrics=yes logtime =all trace=1FF0300

-- Create AWR snapshot and produce AWR report
exec dbms_workload_repository.modify_snapshot_settings(null, <original-value>);
exec dbms_workload_repository.create_snapshot;
@?/rdbms/admin/awrrpt
```

```
-- Change AWR snap interval to 15 minutes and create snapshot
exec dbms_workload_repository.modify_snapshot_settings(null, 15);
exec dbms_workload_repository.create_snapshot;

-- Optionally, enable SQL trace for Data Pump processes or specific SQL ID
alter system set events 'sql_trace {process: pname = dw | process: pname = dm} level=8';
alter system set events 'sql_trace[SQL: 03g1bnw08m4ds ]';

-- Run Data Pump job with trace (Doc ID 286496.1)
expdp ... metrics=yes logtime=all trace=1FF0300
impdp ... metrics=yes logtime=all trace=1FF0300

-- Create AWR snapshot and produce AWR report
exec dbms_workload_repository.modify_snapshot_settings(null, <original-value>);
exec dbms_workload_repository.create_snapshot;
@?/rdbms/admin/awrrpt
```

```
-- Change AWR snap interval to 15 minutes and create snapshot
exec dbms_workload_repository.modify_snapshot_settings(null, 15);
exec dbms_workload_repository.create_snapshot;

-- Optionally, enable SQL trace for Data Pump processes or specific SQL ID
alter system set events 'sql_trace {process: pname = dw | process: pname = dm} level=8';
alter system set events 'sql_trace[SQL: 03g1bnw08m4ds ]';

-- Run Data Pump job with trace (Doc ID 286496.1)
expdp ... metrics=yes logtime =all trace=1FF0300
impdp ... metrics=yes logtime =all trace=1FF0300

-- Create AWR snapshot and produce AWR report
exec dbms_workload_repository.modify_snapshot_settings(null, <original-value>);
exec dbms_workload_repository.create_snapshot;
@?/rdbms/admin/awrrpt
```

Data Pump Trace

Collect:

- Data Pump log file
- AWR report
- Data Pump trace files
  - Stored in the database trace directory
  - Control process file name: `*dm*`
  - Worker process file names: `*dw*`

How to trace Data Pump jobs

- MOS Doc ID 286496.1

News

Coming or already there

Speed up imports by adding constraints in NOVALIDATE mode

```
--Example of which commands Data Pump import might execute as part of an import

create table sales ( .... );

insert into sales as select ... ;

alter table sales add constraint c_sales_1 check (c1 in (0,1)) enable validate;
alter table sales add constraint c_sales_2 check (c2 in ('A','B')) enable validate;
alter table sales add constraint c_sales_3 check (c3 > 0) enable validate;
```

Recursive full table scan

Recursive full table scan

Recursive full table scan

```
--Add constraints using NOVALIDATE clause

impdp ... transform=constr_novalidate
```


```
--Transforming constraints to NOVALIDATE to speed up import

alter table sales add constraint c_sales_1 check (c1 in (0,1)) enable novalidate;
alter table sales add constraint c_sales_2 check (c2 in ('A','B')) enable novalidate;
alter table sales add constraint c_sales_3 check (c3 > 0) enable novalidate;
```

Instant,
No full table scan

```
--Transforming constraints to NOVALIDATE to speed up import

alter table sales add constraint c_sales_1 check (c1 in (0,1)) enable novalidate;
alter table sales add constraint c_sales_2 check (c2 in ('A','B')) enable novalidate;
alter table sales add constraint c_sales_3 check (c3 > 0) enable novalidate;
```

NOVALIDATE Constraints

- Potentially a huge time saver

- Validate constraints after import, or even after go-live

```
alter table sales modify constraint ... validate;
```

  - Still requires a full scan of the table
  - But no table lock!

NOVALIDATE Constraints

- Use with care if you are transforming data on import
- Requires 19.23.0 Data Pump Bundle Patch
- Data Pump always validates certain constraints
- NOVALIDATE constraints prevent the optimizer from certain query rewrites
  - Check `QUERY_REWRITE_INTEGRITY`

Use index size to determine parallel degree on index creation

Index Creation

Before 12.1

1 worker, `CREATE INDEX .... PARALLEL 16`

Really good for few big indexes

From 12.1

16 workers, `CREATE INDEX .... PARALLEL 1`

Really good for many small indexes

From 23

1 worker, `CREATE INDEX .... PARALLEL 8`
1 worker, `CREATE INDEX .... PARALLEL 4`
1 worker, `CREATE INDEX .... PARALLEL 2`
2 workers, `CREATE INDEX .... PARALLEL 1`

The best of both worlds

Thank You