COLLABORATE14

TECHNOLOGY AND APPLICATIONS FORUM FOR THE ORACLE COMMUNITY

#C14LV

Wresting control of your Oracle data with Heat Map and ILM in Oracle DB 12c

John Kanagaraj
Member of Technical Staff,
PayPal Database Engineering,
An eBay Inc. company

REMINDER

Check in on the COLLABORATE mobile app

Agenda

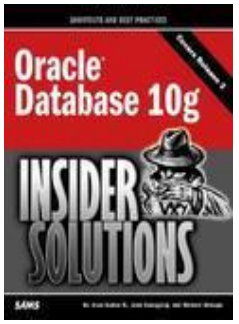
- Data challenges in the real world
- Introducing ILM
- Various methods to contain growth
- Introduction to Oracle Database 12c ILM
  - Oracle Database 12c Heat Map
  - Automatic Data Optimization
  - Partitioning techniques
- Leveraging storage vendor optimizations
- Rolling your own ILM
- Next steps
- Q & A

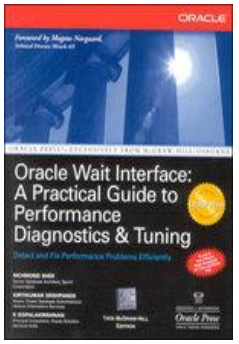
Related IOUG Sessions

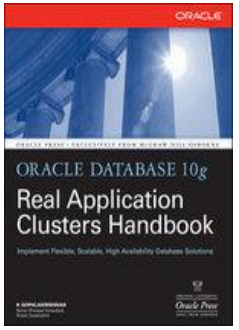
- Download these papers/PPT from IOUG Website
- Helps understand functionality, syntax and usage
  - 185: How Hot Is My Data? Leveraging Automatic Database Optimization (ADO) Features in Oracle 12c Database For Dramatic Performance Improvements
  - 187: Something Old, Something New: Leveraging Oracle 12c's Information Lifecycle Management (ILM) Features for Improved Database Performance
  - 14761: Exploring 11g/12c Partitioning New Features and Best Practices

Speaker Qualifications

- Currently Database Engineer @ PayPal
- Has been working with Oracle Databases and UNIX for too many years ☺
- Author and Technical editor
- Frequent speaker at OOW, IOUG COLLABORATE and regional OUGs
- Oracle ACE
- Contributing Editor, IOUG SELECT Journal
- Loves to mentor new speakers and authors!
- http://www.linkedin.com/in/johnkanagaraj







SELECT JOURNAL

ORACLE ACE

Housekeeping

- Check the font sizes
  - Can you read this at the back of the room?
  - Can you read this at the back of the room?
  - Just kidding!
- Silence your Phones!
- Q & A : Ask as we go along (and I will repeat the question)
  - Keep it relevant to the slide at hand
  - I might defer the question to a later slide or to the end
- It is a long day, so if you nod off it is ok (hopefully no snoring!)
- Survey: Challenges with DB size, Partitioning, 12c, Global Indexes

Data Challenges in the real world

So what is ILM

- ILM – “Information Lifecycle Management”
- Fancy word for understanding, purging and archiving data
  - Strategy, guided by business needs and rules
  - Results in policies, processes and tools to manage data lifecycle
- Policies need to come first: defined by business
  - Usually defined by compliance; users want “retain forever”!
  - Needs cataloging and understanding of data assets
- Processes define how to handle ILM
  - Defines what should be purged/archived/stored forever
  - Classifies and sets retention for data
- Tools – Used by techies to implement ILM policies

Data Challenges in the real world

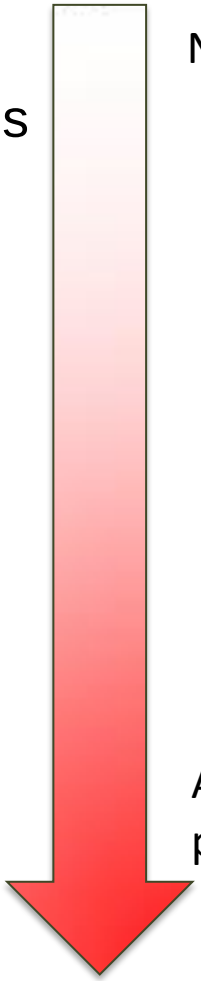
- Data – structured/unstructured - is exploding
- Compliance requires longer data retention
- "Keep forever" policies for legacy data and programs
- Unable to segregate data by access and by retention easily
- Storage tiering requires ability to *physically* segregate data
- Database manageability constrained by size
- No accepted standards to manage data lifecycle
  - External standards lacking
  - Internal standards usually missing
- "Do More with Less" mantra from Business

Typical approaches to data challenges

- "Do Nothing" / "Do no harm" / (Let sleeping dogs lie)
  - When storage cost is lesser than cost of throwing away data
  - Legacy data that no one understands
  - Data needs to be kept forever (research, health, "master" data)
- Compression
  - Reduce cost of storing data
  - Transparent access (almost)
  - Still not a good solution for all types of data
  - Can leverage "tiered storage" approach
- Archive to another store
  - Typically not accessible "online"
  - Still need to purge at some point in time

Types of data : “classify before you kill”

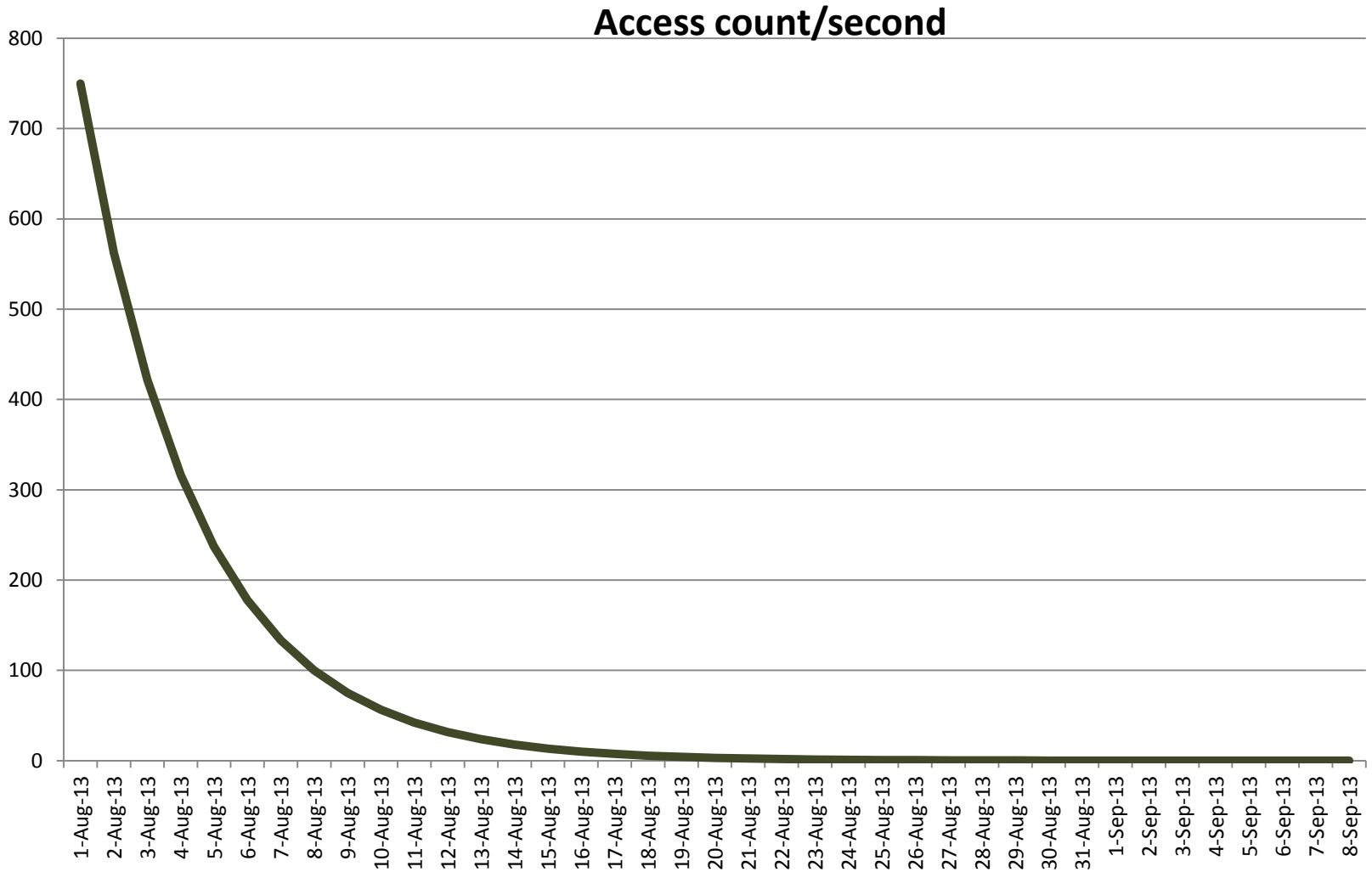
- Master
  - Typically long-lived data: User details/credentials
  - Evolves slowly – Active/Inactive patterns
- Transactions
  - Produced by interactions related to master data
  - Usually voluminous: Sales records, Cart details
  - Typically has a defined lifecycle
  - Changes master data’s state
- Saga
  - Typically records changes to master
  - Shorter life than Transactions
  - E.g. Error logs, external state change events



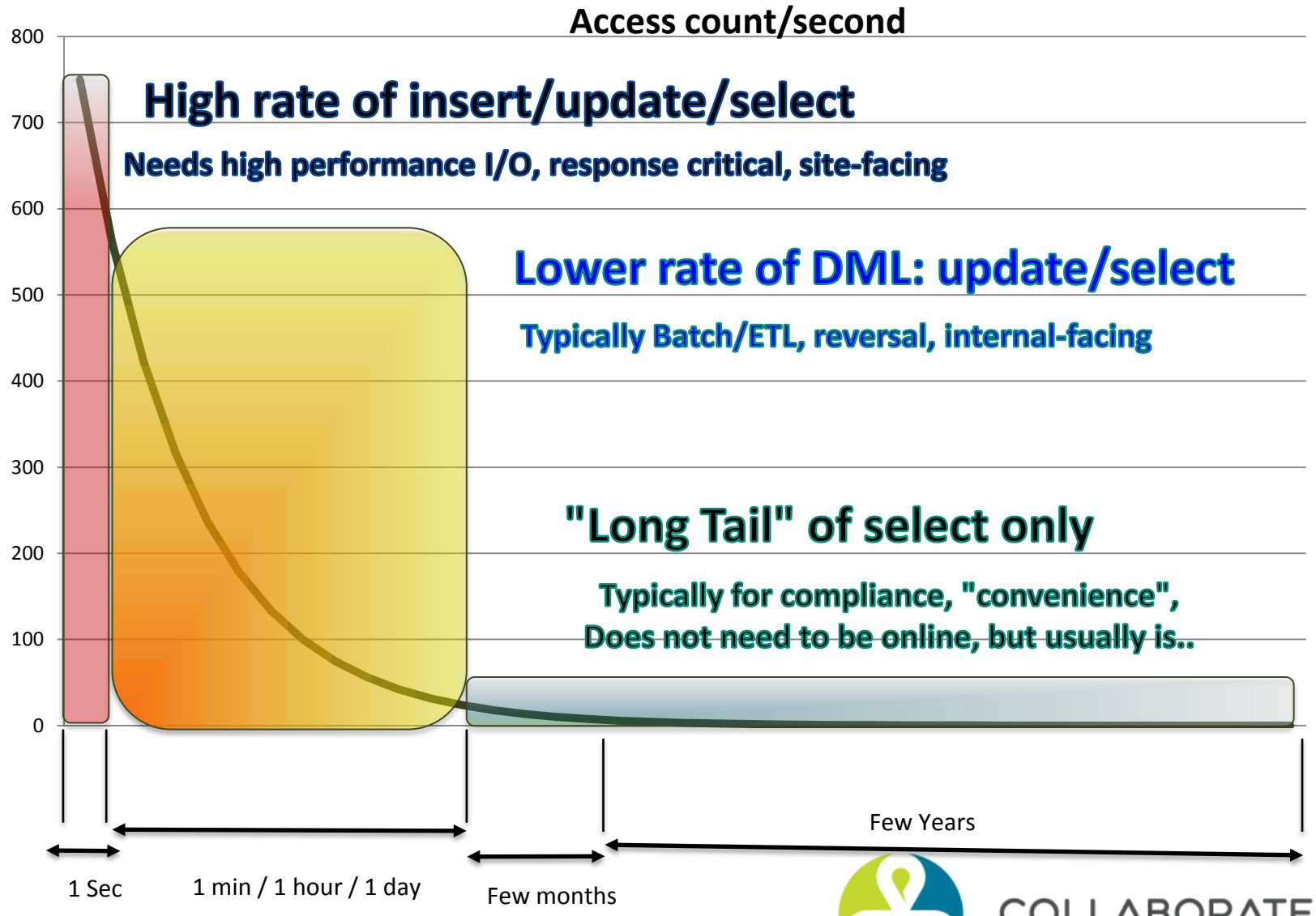
ILM Compliance Policies

- Data Retention – Consider the data being handled: Is it possible to purge/archive?
- Immutability – Does the data change in any way, and how can you prove it did not change since it was “frozen”
- Privacy – Who controls access to archives and how do we protect it?
- Auditing – How do we track who requested this data?
- Expiration – How do we ensure that data is purged as per agreed policies, both external and internal?
- Restoration – How do we store/restore this data in a manner that allows access even past the technology’s “sell-by” date?

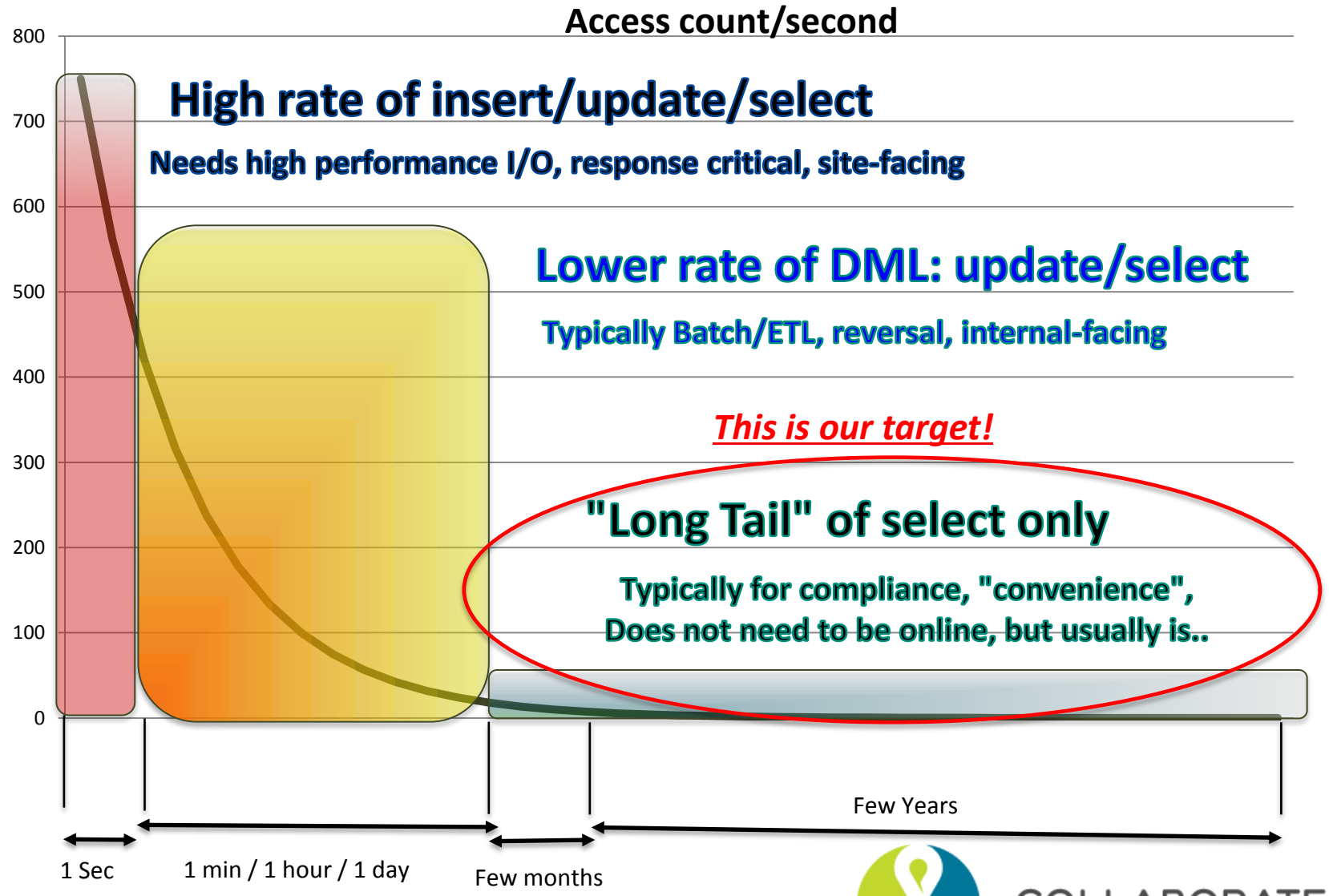
Typical Data Access Patterns



Typical Data Access Patterns



Typical Data Access Patterns



Tools you need to implement ILM

HeatMap / SegStats / Roll-your-own

- Ability to understand data access patterns
  - What parts are being Inserted, Updated, Deleted or Read?
  - What is the rate at which this is being done?
  - How are they (and Who is) performing these activities?

Partitioning and ADO

- Ability to segregate data by these access patterns
  - Needs physical separation at lowest level possible
  - A method to divide (or “partition”) this data by access
  - Typically driven by Time (or Date/Time)
- Ability to handle disposition of data
  - Automatic, enforceable means of segregating data
  - Application transparency
  - Provide ability to access offline or near-line archived data

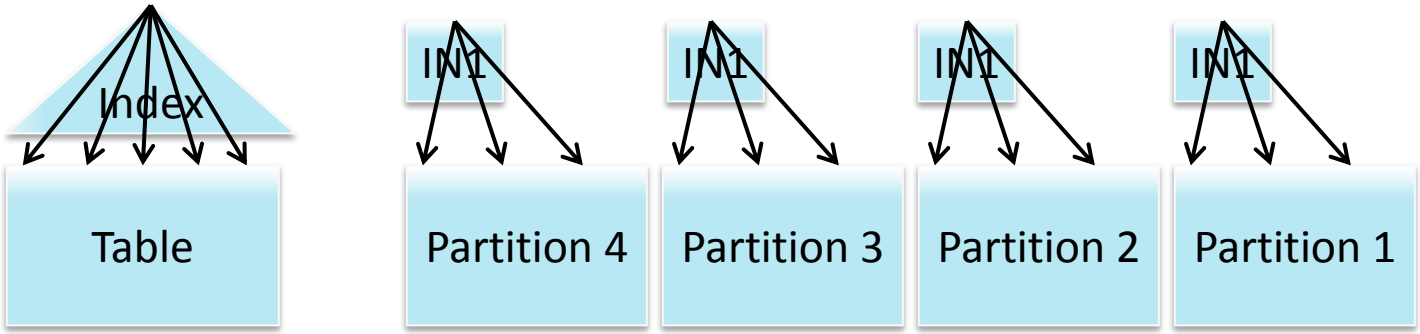
Oracle Partitioning

A.K.A. Carving Up A Large Object Into Manageable Pieces

- Partitioning enables *efficient* data purging/archiving

Oracle Partitioning

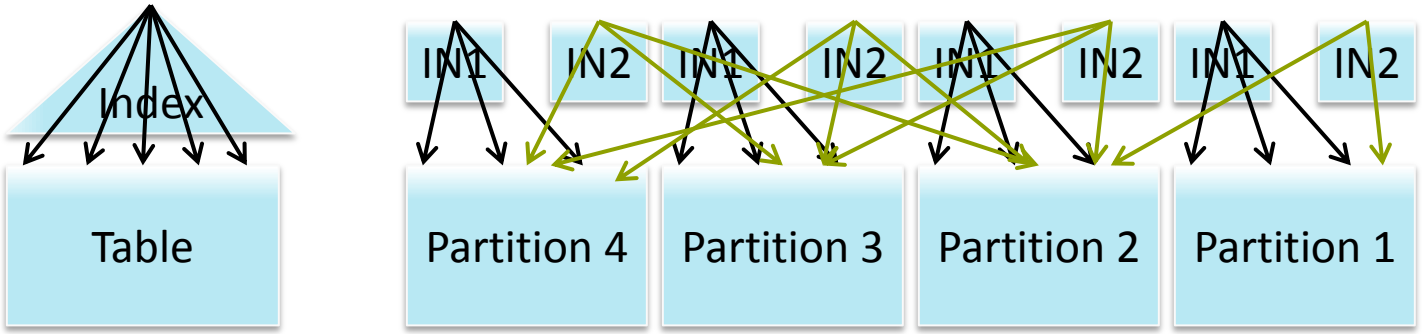
A.K.A. Carving Up A Large Object Into Manageable Pieces



- Partitioning enables *efficient* data purging/archiving
- IN1 Local index partitioned by part_key: single partition probe

Oracle Partitioning

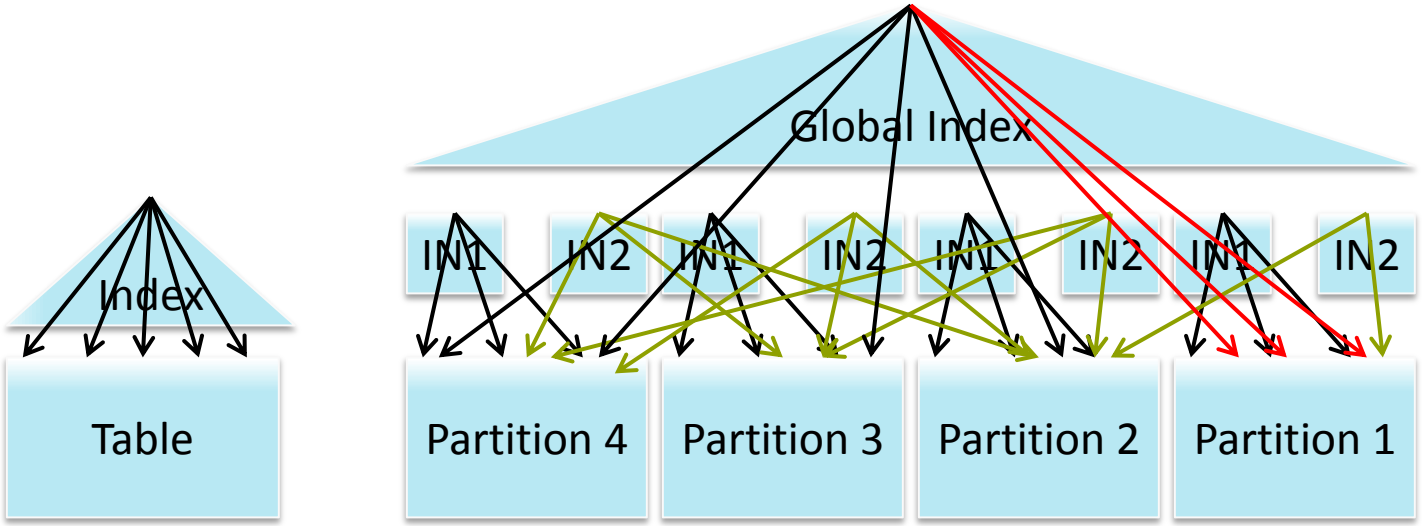
A.K.A. Carving Up A Large Object Into Manageable Pieces



- Partitioning enables *efficient* data purging/archiving
- IN1 Local index partitioned by part_key: single partition probe
- IN2 Local index not accessed by part_key: multi-partition probe

Oracle Partitioning

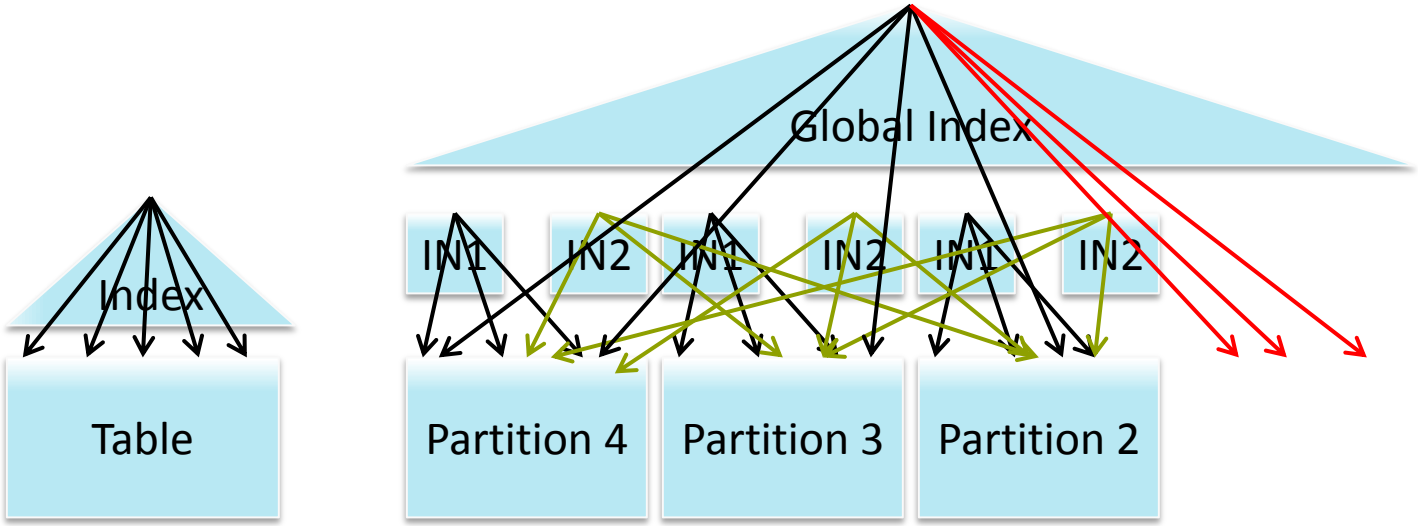
A.K.A. Carving Up A Large Object Into Manageable Pieces



- Partitioning enables *efficient* data purging/archiving
- IN1 Local index partitioned by part_key: single partition probe
- IN2 Local index not partitioned by part_key: multi-partition probe
- Global index: Index rows deleted during partition maintenance

Oracle Partitioning

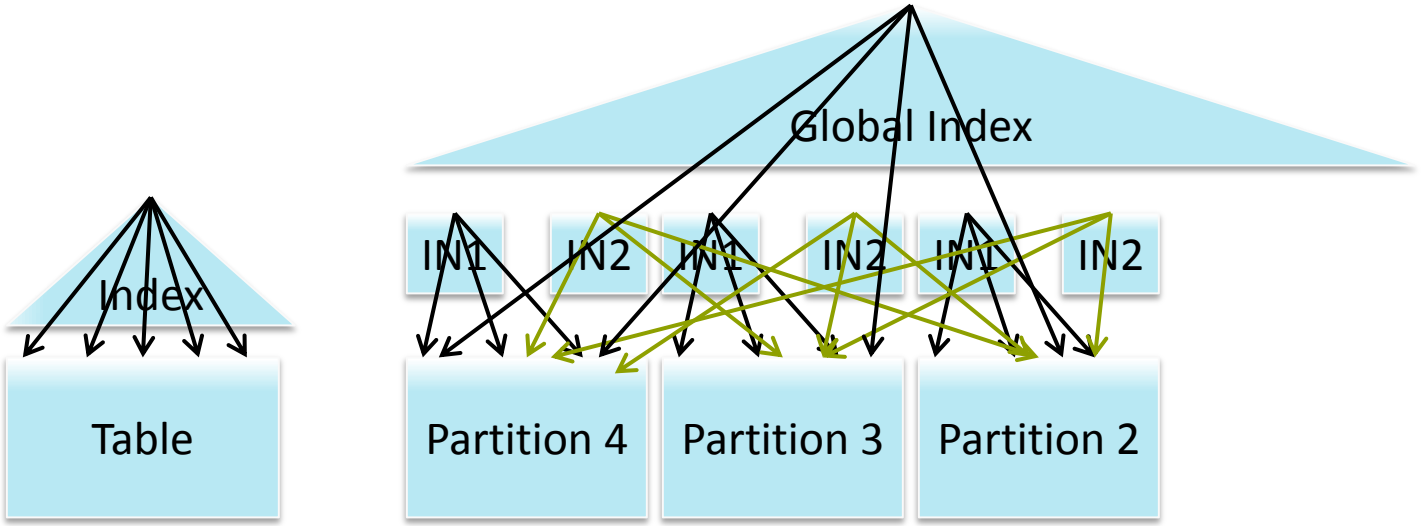
A.K.A. Carving Up A Large Object Into Manageable Pieces



- Partitioning enables *efficient* data purging/archiving
- IN1 Local index partitioned by part_key: single partition probe
- IN2 Local index not partitioned by part_key: multi-partition probe
- Global index: Index rows deleted during partition maintenance
  - Index row deletes async'ed in Oracle DB 12c

Oracle Partitioning

A.K.A. Carving Up A Large Object Into Manageable Pieces



- Partitioning enables *efficient* data purging/archiving
- IN1 Local index partitioned by part_key: single partition probe
- IN2 Local index not partitioned by part_key: multi-partition probe
- Global index: Index rows deleted during partition maintenance
  - Index row deletes async'ed in Oracle DB 12c
- Range partitioning by Time, sub-partitioning by hash is a typical pattern
  - Choice of partitioning key is key to creating the right type of indexes
  - Surrogate Key can be Time/Date based

Oracle Partitioning

- ILM needs ability to segregate data by these access patterns
  - Needs physical separation at lowest level possible
  - A method to divide (or “partition”) this data by access
  - Typically driven by Time (or Date/Time)
- Table and Index partitioning is a must for ILM:
  - Partitioned objects have physically distinct segments
  - Difference shown in OBJECT_ID and DATA_OBJECT_ID
  - Local indexes preferred!
- Most objects have Time-oriented lifecycle
  - Range partitioning by Time is most normal pattern
  - Ideal if partitioning key is a number representing time (or date)
  - Time + Sequence = A Key unique and partitionable by time

Oracle Partitioning

```
create table TEST_TIMEDID
(TIMEDID  NUMBER not null,
VCOL1    VARCHAR2(100) not null,
VNUM1    NUMBER not null,
STATUS   CHAR(1),
CREATED_EPOCH_TIME NUMBER not null,
UPDATED_EPOCH_TIME NUMBER)
partition by RANGE (TIMEDID) -- Partition Width is 6 months
(partition lc_2013_01_01 values less than (9713772914540544O), -- 2013/01/01 00:00:00
partition lc_2013_06_30 values less than (9825098466852864O), -- 2013/06/30 00:00:00
partition lc_2013_12_27 values less than (9936424019165184O), -- 2013/12/27 00:00:00
partition lc_pmax values less than (maxvalue));
```

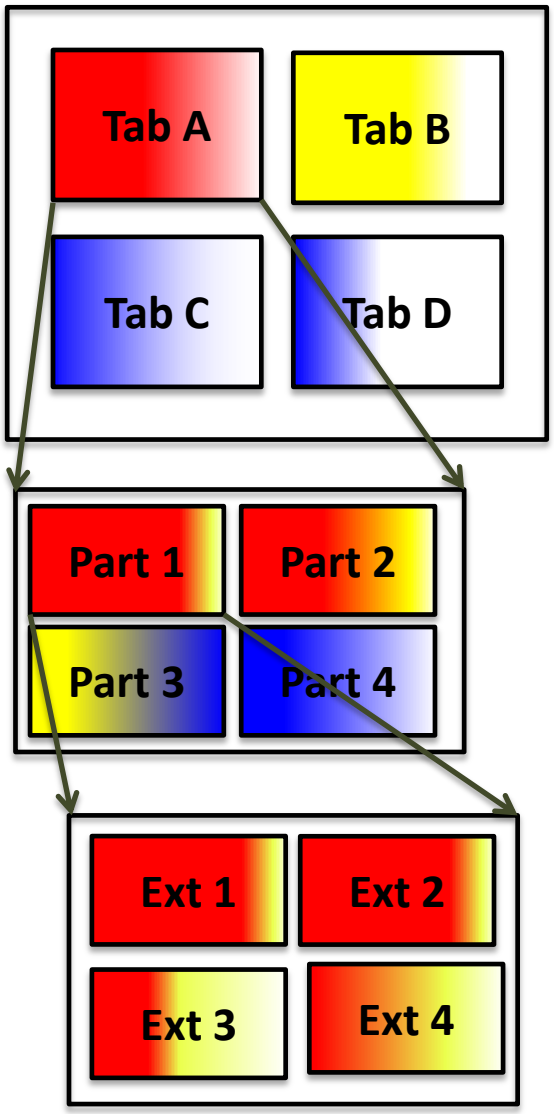
- Create a Sequence Start 1 Max 4294967295, CYCLE
- Get EpochTime (using V$TIMER) – Div by 100 for secs
- Shift up 32 bits - Multiply by 1000000000
- Add the NEXTVAL

- TIMEDID = Epoch Second + Running Oracle Sequence
- Epoch Sec = No. of seconds since Jan 1, 1970 midnight UTC
- http://www.epochconverter.com
  - Epoch time convertor: Epoch to Date/Time and vice versa
- Time model is extensible for multiple sources – just insert a number representing source: TIMEID + Source + Sequence

Understanding data access patterns

- New in Oracle Database 12c : Heat Map
  - DB level heat map showing tables/partitions being used
  - Block/Extent level last modification
  - Detailed statistics of access
  - Low overhead (no cost for object level, <5% for block level)
  - Combined with other licensed options to be effective
- Object (and partition) level tracking pre Database 12c
  - High level usage map in V$SEGMENT_STATISTICS
  - Persisted in AWR (DBA_HIST_SEG_STAT/STAT_OBJ)
  - Partial key/bind values in V$SQL_BIND_CAPTURE
  - Derive approximate change time from SCN_TO_TIMESTAMP(ORA_ROWSCN)

Oracle Database 12c : Heat Map



Tables at Database Level

Partitions at Table Level

Segments at Partition Level

Freshly Inserted, Highly active, Updated by Batch

Infrequent Updates, frequent reads

Infrequent Reads, No updates (never?)

Infrequent Reads, No updates (never?)

Oracle Database 12c : Heat Map

- Set HEAT_MAP = ON to enable in-memory tracking
- Setup heat map using DBMS_* programs
  - DBMS_ILM_ADMIN to setup tracking parameters
- View in-memory stats using V$HEAT_MAP_SEGMENT
- Flushed to DBA_HEAT_MAP_SEGMENT and DBA_HEAT_MAP_SEG_HISTOGRAM
- Use DBMS_HEAP_MAP package to view as well
- Sets you up to implement ADO (Automatic Data Optimization)
  - Possible to create rules to implement data retention and other policies

Oracle Database 12c: ADO

- Automates compression and movement of data
- Uses Heat Map data collected prior
- Implemented using DBMS_ILM package
- Creates “in-database” archiving using compression
  - Needs license
  - Does NOT go across databases
- Exposed via DBA_ILM% views
  - DBA_ILMDATAMOVEMENTPOLICIES: Data movement related attributes
  - DBA_ILMEVALUATIONDETAILS: Evaluation of ADO poliicies
  - DBA_ILMOBJECTS: Mapping of ILM policies to objects
  - DBA_ILMPARAMETERS: Parameters defined by DBMS_ILM* packages
  - DBA_ILMPOLICIES: Details of ADO policies
  - DBA_ILMRESULTS: ADO Execution details
  - DBA_ILMTASKS: ADO Execution details

Oracle Database 12c: ADO Examples

```
/* Add a row-level compression policy after 30 days of no modifications */
ALTER TABLE sales MODIFY PARTITION sales_q1_2002
  ILM ADD POLICY ROW STORE COMPRESS ADVANCED ROW
  AFTER 30 DAYS OF NO MODIFICATION;

/* Add a segment level compression policy for data after 6 months of no changes */
ALTER TABLE sales MODIFY PARTITION sales_q1_2001
  ILM ADD POLICY COMPRESS FOR ARCHIVE HIGH SEGMENT
  AFTER 6 MONTHS OF NO MODIFICATION;

/* Add a segment level compression policy for data after 12 months of no access */
ALTER TABLE sales MODIFY PARTITION sales_q1_2000
      ILM ADD POLICY COMPRESS FOR ARCHIVE HIGH SEGMENT
      AFTER 12 MONTHS OF NO ACCESS;

/* Add storage tier policy to move old data to a different tablespace */
/* that is on low cost storage media */
ALTER TABLE sales MODIFY PARTITION sales_q1_1999
  ILM ADD POLICY
  TIER TO my_low_cost_sales_tablespace;
```

ADO and Heat Map Restrictions

- ADO and Heat Map not supported in a CDB database
- Row-level policies for ADO are not supported for Temporal Validity
- Partition-level ADO and compression supported if partitioned on the end-time columns
- ADO does not perform checks for storage space in a target tablespace when using storage tiering
- ADO is not supported on tables with object types, materialized views, IOTs and Clustered tables
- ADO concurrency (the number of simultaneous policy jobs for ADO) depends on the concurrency of the Oracle scheduler.
- ADO Policies are only run in the maintenance windows
- Supplemental logging restrictions
- *“ADO has restrictions related to moving tables and table partitions”??*

Oracle DB 11g: “Back-porting”

- Some high level information available 10g+ and 11g
  - V$SEGMENT_STATISTICS: Tracks access
    - Stats such as “physical reads” & “db block changes”
    - Persisted in AWR (DBA_HIST_SEG_STAT/STAT_OBJ) with timestamp
    - May not be recorded for all objects in a busy database
  - Derive row-level access using bind values
    - Stored in V$SQL_BIND_CAPTURE/DBA_HIST_SQLBIND
    - Manual work to derive access patterns
  - Changed blocks record time in ORA_ROWSCN
    - Derive approximate change time from SCN_TO_TIMESTAMP(ORA_ROWSCN)

Segment Level Stats – AWR (Global/Single)

(use STATSPACK in case you don't have License for AWR!)

```
Segment Statistics (Global)          DB/Inst: TEST/TEST_2  Snaps: 94734-94735
-> % Total shows % of statistic for each segment compared to the global cluster-wide total
   (logical reads, physical reads, gc [cr/cu] blocks [recv/serv])
-> % Capture shows % of statistic for each segment compared to the  total captured
   by AWR for all segments during the snapshot interval
-> Captured Segments account for   84.3% of Total Logical Reads:     311,580,725
-> Captured Segments account for   94.0% of Total Physical Reads:      14,995,258
<snip> -- Other stats include "Physical Read Requests", "UnOptimized Read Requests", "Optimized  Read Requests"
<snip> -- "Direct Physical Reads", "Physical Writes", "Physical Write Requests", "CR Blocks Served/Received"
"Direct Physical Writes", "Table Scans", and "Current Blocks Served/Received"
```

Statistic	Tablespace Owner	Object Name	Subobject Name	Obj. Type	Value	%Total	%Capture
db block changes	TESTDBA	TST_DATA_L	TST_LMTNSCHDACTIONS_TIONS_Q_18	TABLE	1,985,760	N/A	32.2
	TESTDBA	TST_IDX_LM	TST_LMTNSCHDACTIONS_TIONS_Q_18	INDEX	1,151,456	N/A	18.6
	TESTDBA	TS_TXN_DAT	TST_LMTNSCHDACTIONS_IONS_DM_18	INDEX	1,126,832	N/A	18.2
	TESTDBA	TEST_FLOW4	TESTMENT_FLOW T_FLOW_P18	TABLE	1,047,792	N/A	17.0
	TESTDBA	TST_DATA_L	TST_LMTNSCHDACTIONS_	TABLE	863,600	N/A	14.0
gc cr blocks received	TESTDBA	TST_DATA_L	TST_LMTNSCHDACTIONS_	TABLE	570,409	10.9	37.7
	TESTDBA	TST_MAP	TST_MAP	TABLE	356,468	6.8	23.6
	TESTDBA	TEST_FLOW4	TESTMENT_FLOW T_FLOW_P18	TABLE	253,758	4.9	16.8
	TESTDBA	TESTTABL_I	TESTTABL_INFO	TABLE	204,984	3.9	13.6
	TESTDBA	TS_TXN_DAT	PAYMENT_FLOW_RISK _TEST_P342	TABLE	126,949	2.4	8.4
gc cr blocks served	TESTDBA	TST_DATA_L	TST_LMTNSCHDACTIONS_	TABLE	570,409	10.9	37.7

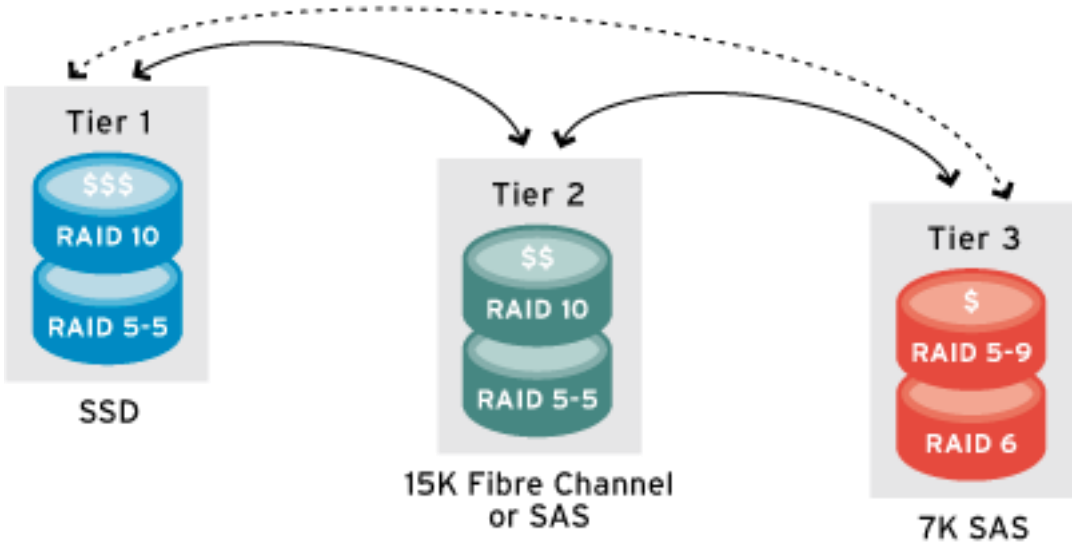
<snipped until end>

- Table/Index/Partition names are truncated…. ☹ Access the data directly!
- DBA_HIST_SEG_STAT and DBA_HIST_SEG_STAT_OBJ

COLLABORATE14
TECHNOLOGY AND APPLICATIONS FORUM FOR THE ORACLE COMMUNITY

Storage Tiering: An essential component

- Most Storage vendors provide some form of tiered storage
  - SAN Array tiers should be mapped to ASM diskgroups (DG's)
  - Create "Compressed", "Archive" tablespaces on these DG's
  - Use ADO to compress/move the required partitions
  - In pre-12c, use available compression methods
  - HCC in Exadata, ZFS Storage Appliance and Pillar Axiom



Off-database: “Transparent Online archive”

- Move archived data to another database
  - Implemented using third-party archive tools
  - E.g. HP RIM, IBM InfoSphere Optim, etc.
  - Most originated from OuterBay (HP acquired 2006 => RIM)
  - Essentially for Oracle E-Business Suite; Now for XML as well
- Main issue: Reduced availability (dependent on >1 database)
  - Essentially based on Database links
  - Separate access path for archived data
    - Mitigates availability concern for critical access paths
    - Not suitable for chatty applications; Low use cases only
- Needs to keep up with the main (DDL, changes, formats, etc.)
- May be built in-house with some effort

Please fill in your feedback form

Link up with me on LinkedIn

John Kanagaraj, PayPal, an eBay Inc. Company