An Oracle DBA's Rosetta Stone for Teradata

Dawn McCormick

Senior Database Consultant

Teradata Development Division

Overview of Presentation

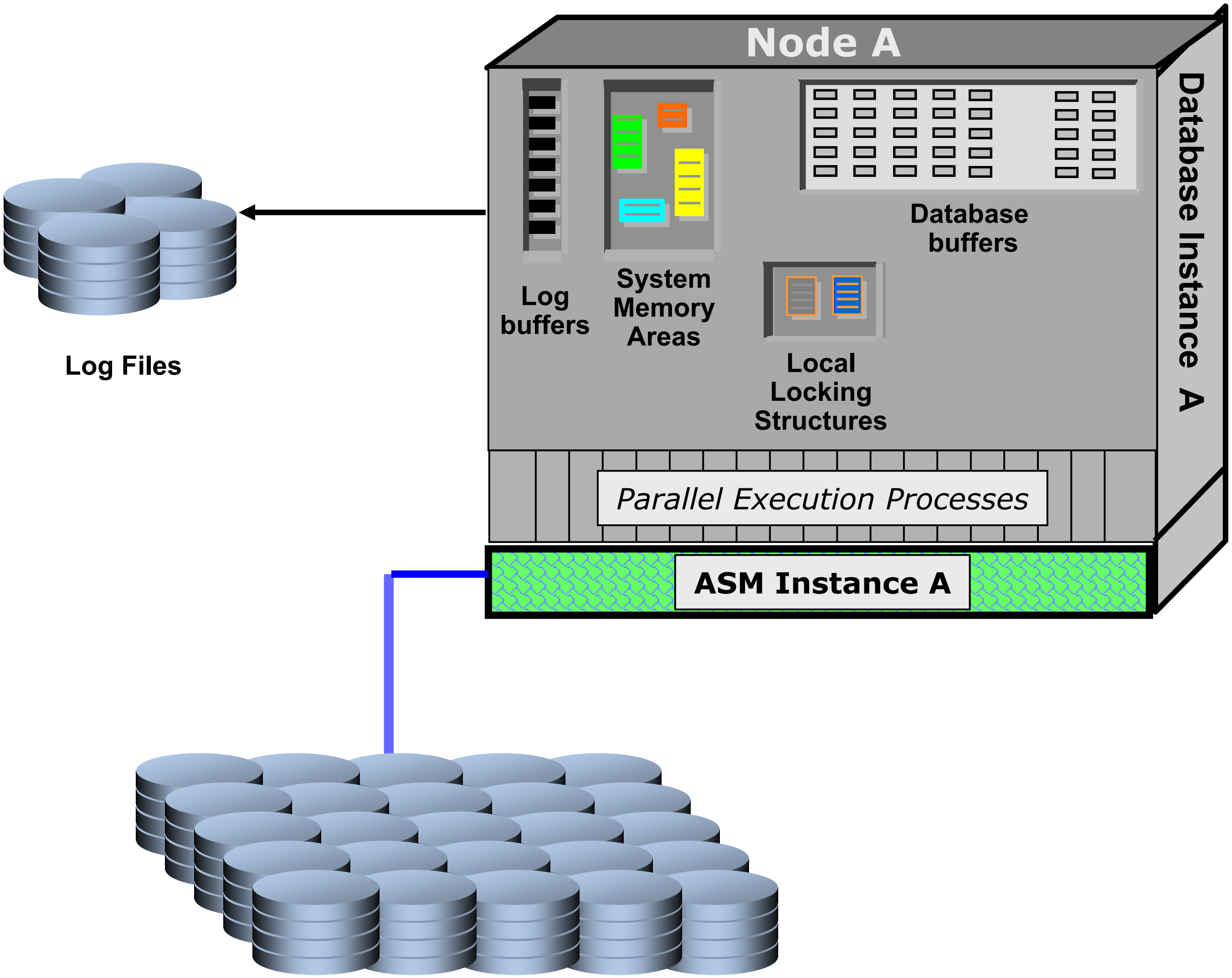
- **Database Architecture**
- **Creating the Data Warehouse System**
- **Connecting Disk Storage to the Database**
- **Table Comparisons**
- **Data Types**
- **Index Comparisons**
- **Materialized Views**
- **Parallelism Comparisons**
- **User Data Maintenance Utilities**

Database Architecture

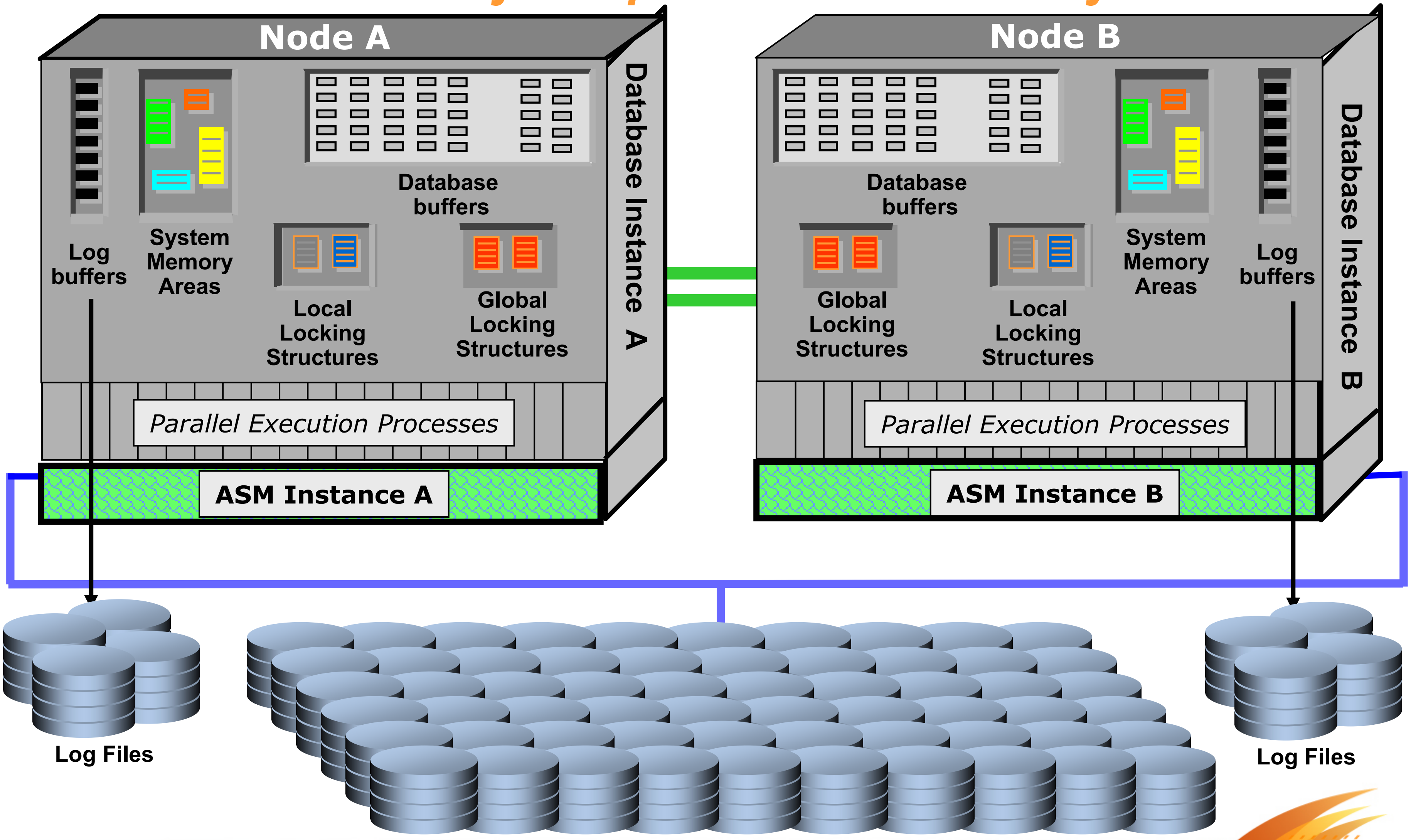
Basic Oracle Database Architecture

- **Oracle single system (node) VLDBs normally uses one database instance**
- **Oracle uses shared everything database architecture for single instance databases**
- **Oracle uses shared disk/loosely coupled shared memory database architecture for multiple node databases**
- **With multi-node Oracle (Real Application Clusters) all nodes see all of the data**
  - Usually one instance per node
  - Data block changes are coordinated between instances via a lock manager over the memory interconnect

Oracle Single Instance = Shared Everything



Oracle Real Application Clusters = Shared Disk/Loosely Coupled Shared Memory



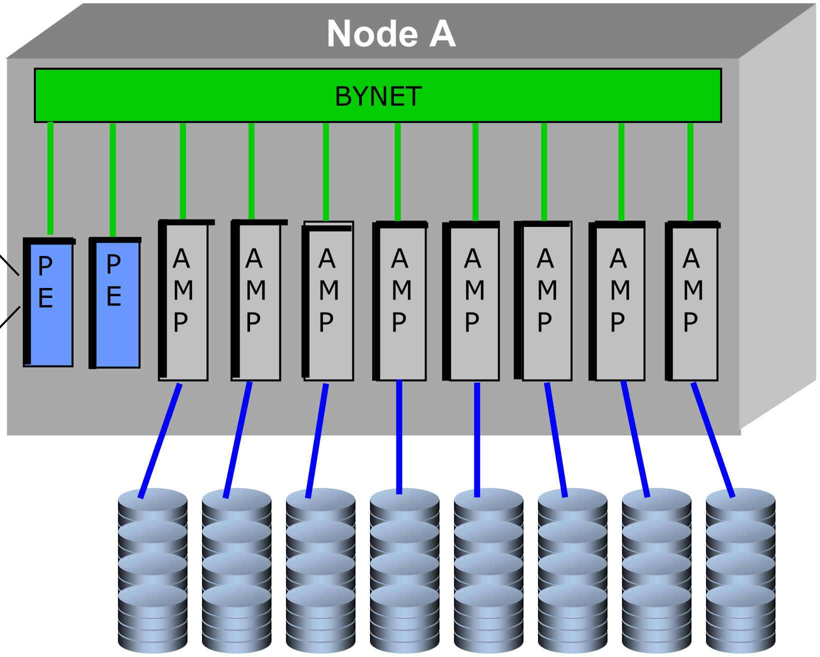
Basic Teradata Database Architecture

- **Teradata uses a shared nothing database model**
- **The database functions have been broken up into multiple repeating software components**
  - Each nodes has multiple Parsing Engines (PEs) that manage user requests
    - PEs manage user sessions
    - PEs parse the user SQL requests
    - PEs perform the optimization of the requests
  - Each node had multiple database engines known as Access Module Processors (AMPs)
  - Data are stored and managed by these AMPs
    - The file system and disk management are performed by the AMPs
    - Each AMP "sees" and manages its portion of the data
  - AMPs perform the data access and manipulation tasks of the database
    - Each AMP works independently of the rest of the AMPs on its part of the database
- **The collection of these software components are the foundation of Teradata's shared nothing software architecture**

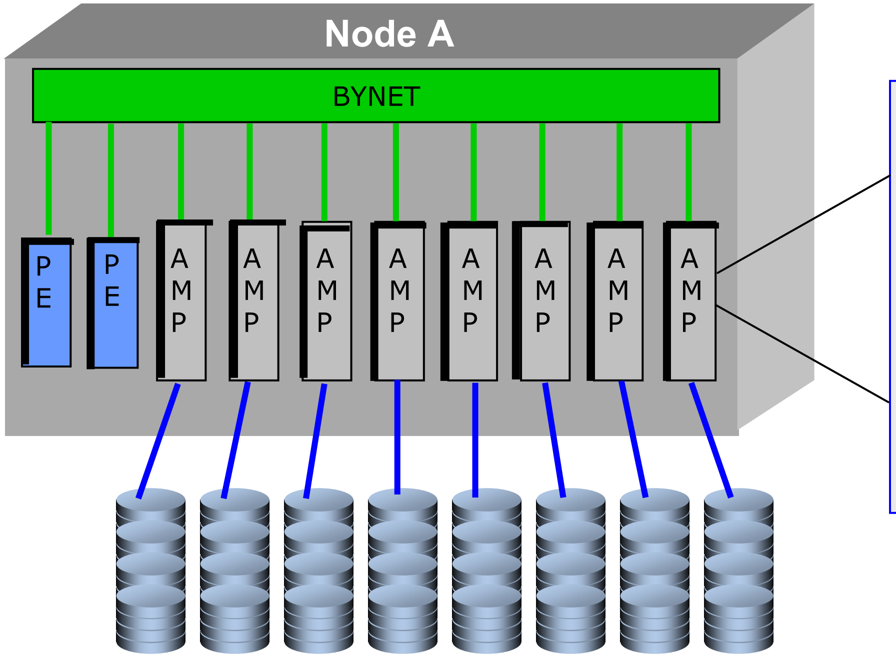
Teradata Single Node = Shared Nothing

Each PE is responsible for:

- Session Control
- SQL Parsing
- Query Optimization
- Package Steps Generation
- Dispatcher



Teradata Single Node = Shared Nothing

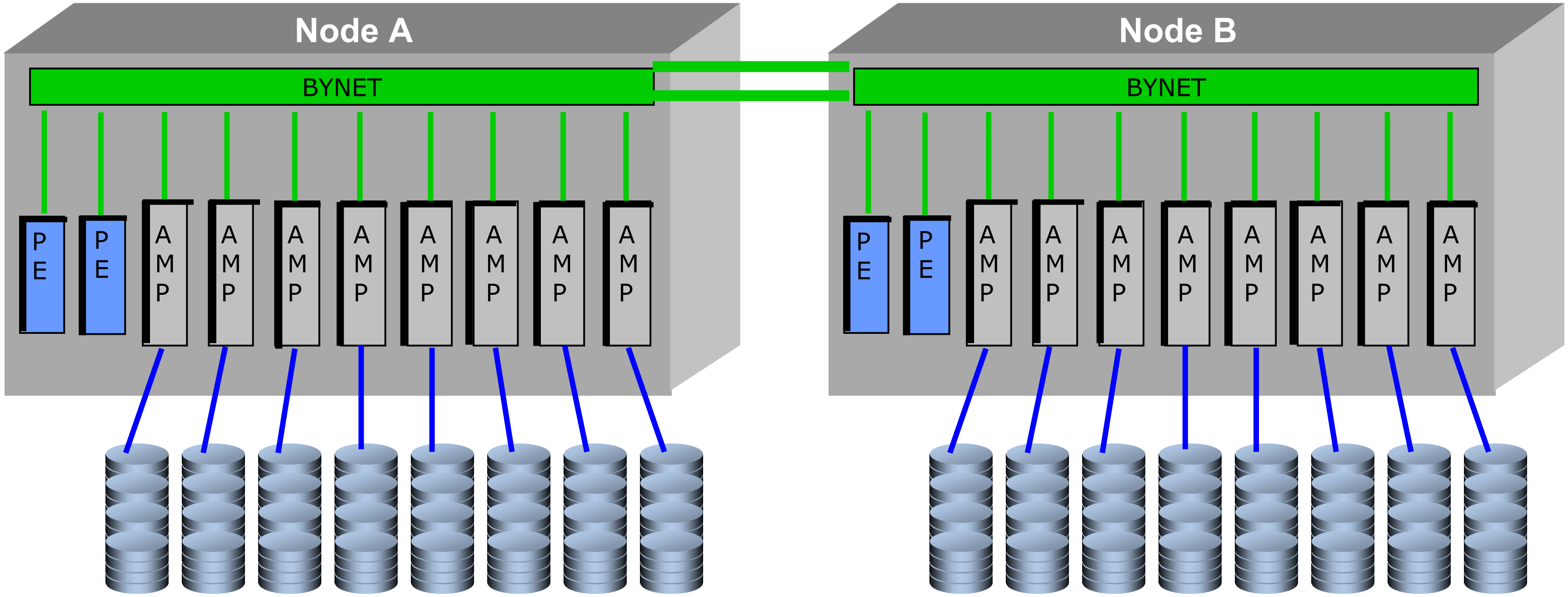


Each AMP has:

- System Memory Structures
- Database Buffers and Processes
- Locking Structures and Processes
- Logging Structures and Processes
- Files System and Disk Management

Each AMP performs data access and data manipulation for user queries

Teradata Multiple Nodes = Shared Nothing



Basic Database Architectural Differences

- **Oracle uses loosely coupled shared memory and shared disk database models**
  - All Oracle RAC instances must see all of the data in the database
  - Oracle instances' data caches must be coordinated to ensure that all instances have a consistent version of the data
  - Data block changes are coordinated between instances via a global lock manager over the memory interconnect
- **Teradata Uses the shared nothing database model**
  - Each AMP manages its piece of the database
  - Data caches are independent, thus eliminating the data cache coordination overhead required to enable the shared model
  - This architecture allows Teradata to scale linearly

Creating the Data Warehouse System

Installing the Data Warehouse System

- **Installing an Oracle data warehouse system is a different effort than installing a Teradata data warehouse system**
- **There are configuration options that must be examined and chosen**
  - DBMS
  - File System
  - Operating System
  - Interconnect
  - Node Hardware
  - Storage Hardware

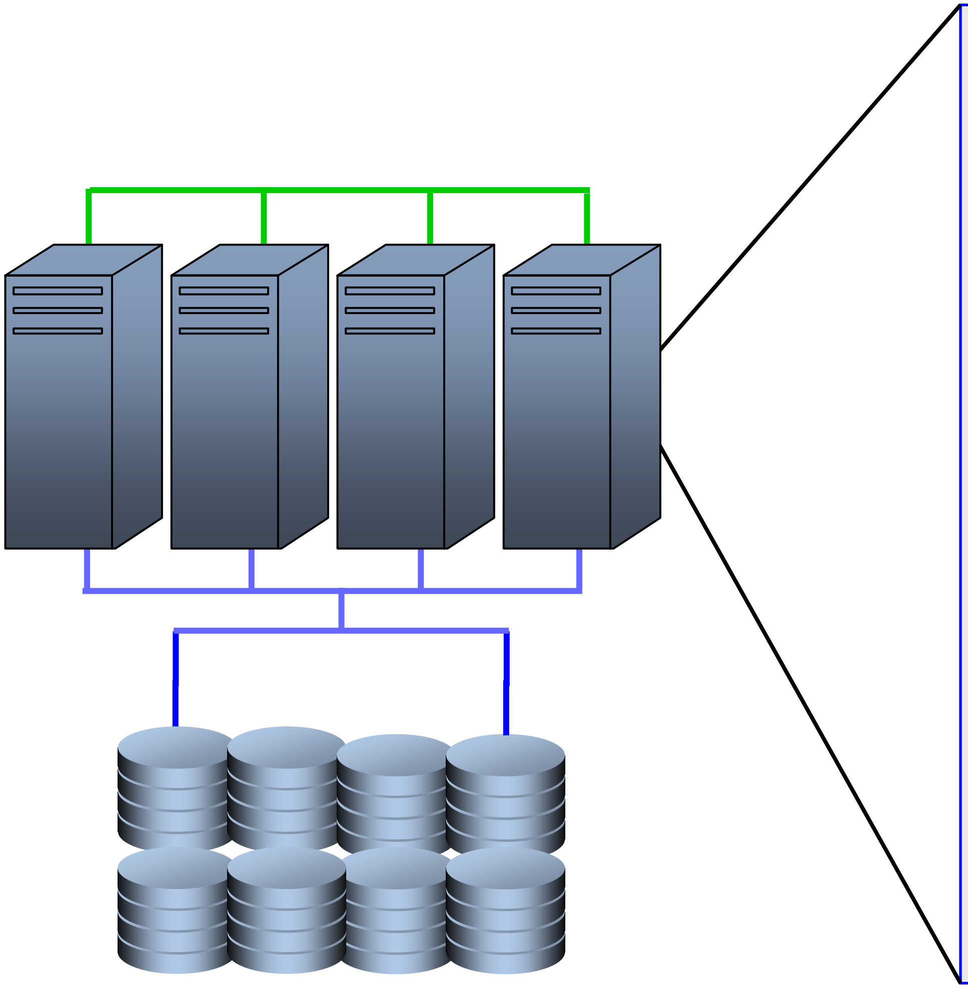
System Platform – Oracle (UNIX)

DBMS	init.ora parameters, partitioning (range, hash, list, composite range-hash, composite range-list, global partitioned indexes, local partitioned indexes), B-Tree indexes, bitmap indexes, parallelism – yes or no, number of parallel execution servers, materialized views, etc.
File System	Raw partitions, Oracle Cluster File System (OCFS), Veritas Cluster File System, Oracle Managed Files, Locally Managed Tablespaces, Dictionary Managed Tablespaces, Automatic Storage Management (ASM), ASM Redundancy Level (Normal, High, External), Automatic Undo Management, and more
Operating System	Solaris, HP-UX, HP Tru64, AIX, Red Hat Enterprise Linux AS/ES 2.1, Red Hat Enterprise Linux AS/ES 3, Suse Linux Enterprise Server 8, Linux Kernel version and errata level, Linux package requirements, Kernel parameters (shared memory, semaphores, etc.), Shell limits for the Oracle user, etc.
Interconnect	RAC, User datagram protocol (UDP) or reliable datagram (RDG) using high-speed network adapters and switches that support TCP/IP, Reliable datagram (RDG) using Memory Channel adapters and hubs, remote shared memory (RMS) using peripheral component interconnect-scalable coherent interface (PCI-SCI) adapters and SCI switches, Oracle OSD Clusterware, Sun Cluster, HP MC/Service Guard, IBM HACMP, etc.
Node Hardware	Dell, IBM, HP, Sun, Blade server or rack mount, x86 or Itanium, PowerPC, SPARC, 2 CPUs/Node or 4 CPUs/Node or n CPUs/Node?, 2 4-CPU nodes or 4 2-CPU nodes?, etc.
Storage Hardware	Dell / EMC SAN, Network Appliance SAN, Network Appliance NAS, IBM ESS, IBM FastT, HP StorageWorks, Sun StorEdge, EMC Symmetrix, RAID 0, RAID 1, RAID 0+1, RAID 5, and more

System Platform – Teradata (UNIX)

DBMS	Primary index, partitioned primary index, secondary index, aggregate join index
File System	Teradata File System
Operating System	UNIX MP-RAS
Interconnect	BYNET
Node Hardware	NCR hardware platform
Storage Hardware	LSI, EMC

You don’t have to worry about configuring the Teradata System



DBMS
Services, Parameters, Partitions

File System
Files, Extents, Stripes
Clusterware

Operating System
Services, Parameters

Interconnect
HW/SW, Protocol

Node Hardware
CPU, Memory, Bus

Storage Hardware
Spindles, Arrays, Channels

Some Beginning Semantics – Databases and Schema

- **An Oracle instance manages one database**
  - Real Application Cluster is an additional software option that allows the sharing of a database between more than one Oracle instance on one or more computer nodes
  - “Schema” group database objects (tablespaces, tables, indexes, etc.) into functional groupings that are created and owned by users
- **The Teradata system manages more than one database**
  - Teradata has a system-wide database named DBC
  - Users create application databases from and allocate space quotas to these application databases from DBC
  - Teradata databases, like schema, organize database objects (tablespaces, tables, indexes, etc.) into functional groupings that are owned by users

Linking Disk Space to the Database: The Oracle World

- **Determine which disks and partitions will be part of the database and become Oracle database files (user data files, temporary files, undo and redo log files, control files, initialization files, etc.)**
- **Determine whether to use Oracle Automatic Storage Management (ASM)**
- **Determine what other file system software will be used**
- **Implement RAID and create the logical disks via your standard cluster file system software**
  - Decide what Oracle files will be stored on this file system
- **Implement Oracle ASM if using it to manage database files**
  - Organized disks into disk groups within ASM
  - Determine what disk groups will support what database objects (tables, indexes, etc.)
- **The administrator function is responsible for configuring Oracle storage to support the demands of data warehousing**

Linking Disk Space to the Database: The Teradata World

- **Disk space is allocated to the Teradata system during system staging prior to the system being delivered**
  - Disks are configured for RAID protection in Logical Units (LUNs)
  - LUNs are partitioned into “slices” (UNIX) or “partitions” (Microsoft Windows)
    - These partitions/slices are known as Pdisks
  - The Parallel Database Extensions (PDE) layer of the database groups pdisks together into Vdisks
  - The PDE assigns Vdisks to AMPs
- **With the completion of the staging process, disk space is evenly distributed to each unit of parallelism (AMP) in Teradata**

Teradata is Responsible for Configuring Storage to Support the Demands of Data Warehousing

Teradata Instance Creation Tasks

- **The Teradata software is installed and configured in the system staging process prior to system delivery**
- **As with all databases, some basic information is needed:**
  - Estimated user data requirement
  - Estimated number of users
  - Estimated system performance requirements
- **Based on this information, the Teradata software is configured:**
  - System storage
  - Number of system nodes
  - Number of PEs
  - Number of AMPs

You Don't Create A Teradata "Instance"
This is Done for You by Teradata

Database Setup/Component Similarities

Database Creation Task	Oracle	Teradata
Script(s) to Create Catalog/Data Dictionary Tables and Views	Catalog.sql, Catclust.sql, Catproc.sql	DIP
Manage “In-Flight Transactional Data	Rollback/Undo Segments	Transient Journal
Database Logs	Redo Logs	Permanent Journal
Temporary Space/Work Space	Temp Space	Spool Space
Default Permanent Space	Default Tablespace	Default Database
Table Block Size	System-wide blocksize setting Optional different table blocksize setting requiring sizing and creation of bufferpools for each additional different blocksize	System-wide blocksize setting optional different table blocksize setting Teradata manages the different blocksizes automatically
Database Startup Information	Spfile, init.ora, OS environment variables, control files, ASM instance information	XCTL, Vconfig.out

Connecting Disk Storage to the Database

Allocating Disk Space to Database Objects in Oracle

- **Tablespaces allocate disk space from assigned data files or ASM disk groups**
- **Table and index partitions use tablespaces to allocate disk space from one or more data files or ASM disk groups**
- **Data files and ASM disk groups may support more than one table/index and more than one table/index partition**
- **Initial, Next Extent Sizes**
- **Minextent, Maxextent Limits**

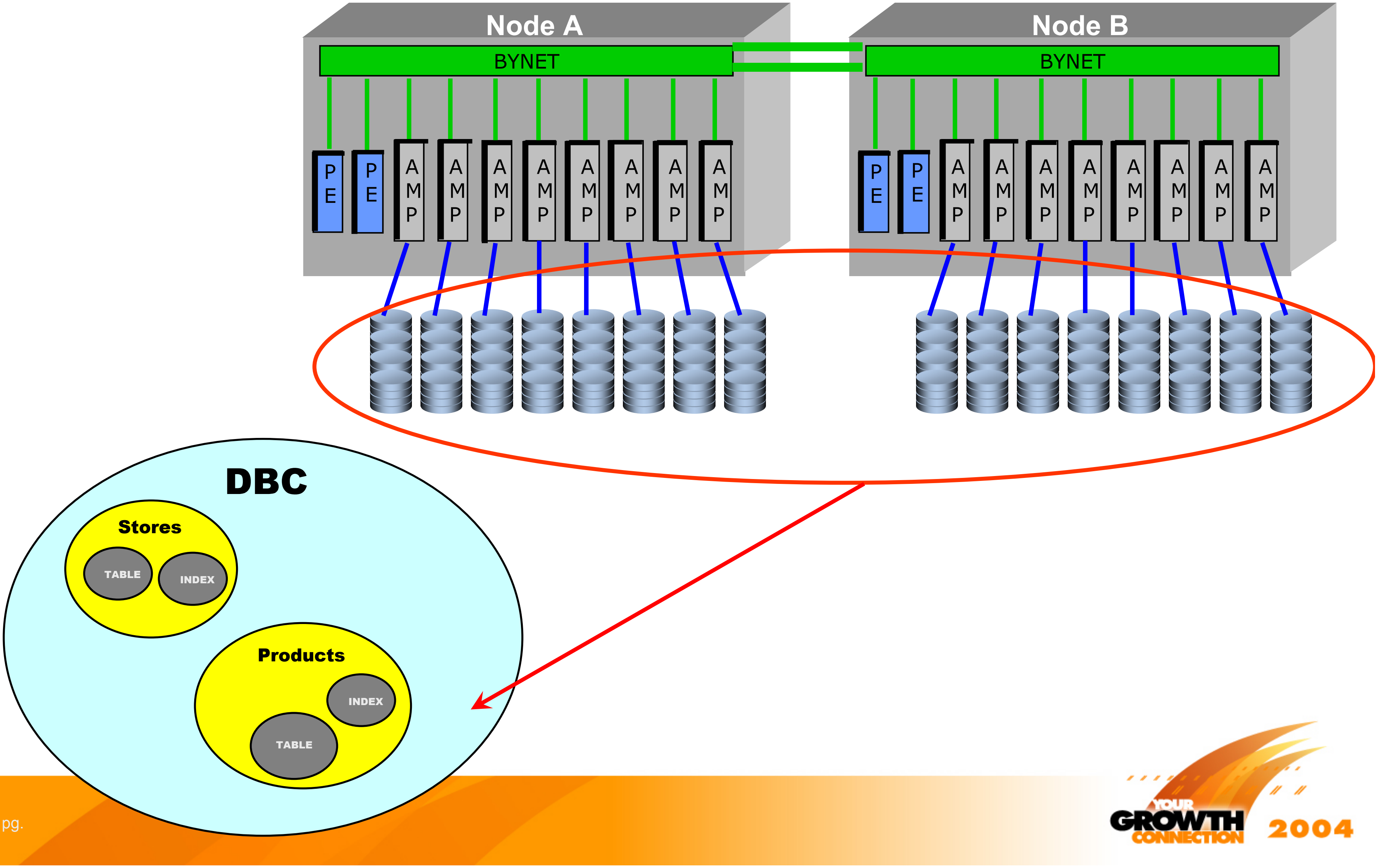
```
CREATE TABLESPACE Tablespace1

DATAFILE '/datadir/datafile6.dat' SIZE 20M

DEFAULT STORAGE (INITIAL 10K NEXT 50K
MINEXTENTS 1 MAXEXTENTS 999) ONLINE;
```

Allocating Disk Space to Database Objects in Teradata

- **All the space in the Teradata system is initially assigned to the DBC (system) database**
- **As user databases are created, each is given a space quota/maximum (named “perm space”) from Teradata system storage**
  - The space is not physically allocated to the user database until data is stored in a table in that database
  - Unused space is available for system-level transient operations
- **User databases allocate space as needed from the system storage until it reaches the defined user database quota/maximum**

```
CREATE DATABASE db1 FROM dbc AS
PERM = 1000000000,
SPOOL = 1000000000;
```

Disk Space and the Teradata Database



Large Disk Space Allocation DDL

Teradata V2R5.1

```
CREATE USER tpcd3000g AS PERM=12000E9,
PASSWORD=tpcd3000g;
```

Execute Once for Each Database

Oracle10g Release 1

```
create tablespace ts_l1
datafile '/dbms/links/line_1' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;

create tablespace ts_l2
datafile '/dbms/links/line_2' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
.
.  - Omitted 333 more partitioning clauses -
.
create tablespace ts_l336
datafile '/dbms/links/line_336' size 9000m reuse
extent management dictionary default storage (initial 1050m next
20m maxextents unlimited pctincrease 0)
;
```

Execute 336 “Create Tablespace” Commands to Store 1 Table

Table Comparisons

Where Are Tables Stored?

- **Oracle tables are created in tablespaces belonging to schema**
  - Tables have space utilization parameters describing how they will use space in tablespaces
- **Teradata tables are created in databases**
  - Disk space is managed by Teradata
  - Disk space is allocated to tables as they need to store data

Simple CREATE TABLE Command Comparison

Oracle

```
CREATE TABLE Table1

( Col1 NUMBER,

  Col2 NUMBER,

  Col3 NUMBER )

TABLESPACE Tablespace1

STORAGE (INITIAL 6144 NEXT 6144

MINEXTENTS 1 MAXEXTENTS 5 );
```

Teradata

```
CREATE TABLE Table1 ,FALLBACK ,
     NO BEFORE JOURNAL,
     NO AFTER JOURNAL
     (Col1 INTEGER,
      Col2 INTEGER,
      Col3 INTEGER)
UNIQUE PRIMARY INDEX ( Col1 );
```

CREATE TABLE Command: Elements Common to Oracle and Teradata

- **All tables have names**
- **All tables contain columns with data types**
- **Constraints can be defined on columns**
- **Referential integrity can be defined**
- **May define global temporary tables**
- **May have a defined data block size**
- **May have triggers**
- **May define freespace**

Additional Teradata CREATE TABLE Parameters

- **A primary index is used to identify a table's partitioning columns to use to distribute data to AMPs**
- **Recovery (fallback) option allows mirroring of table data on another AMP for very high availability systems**
- **Journaling options may be defined for each table**
  - Before Journaling
  - After Journaling
- **All these options are built into the core Teradata database product, available for use "out of the box" and are self-managing**

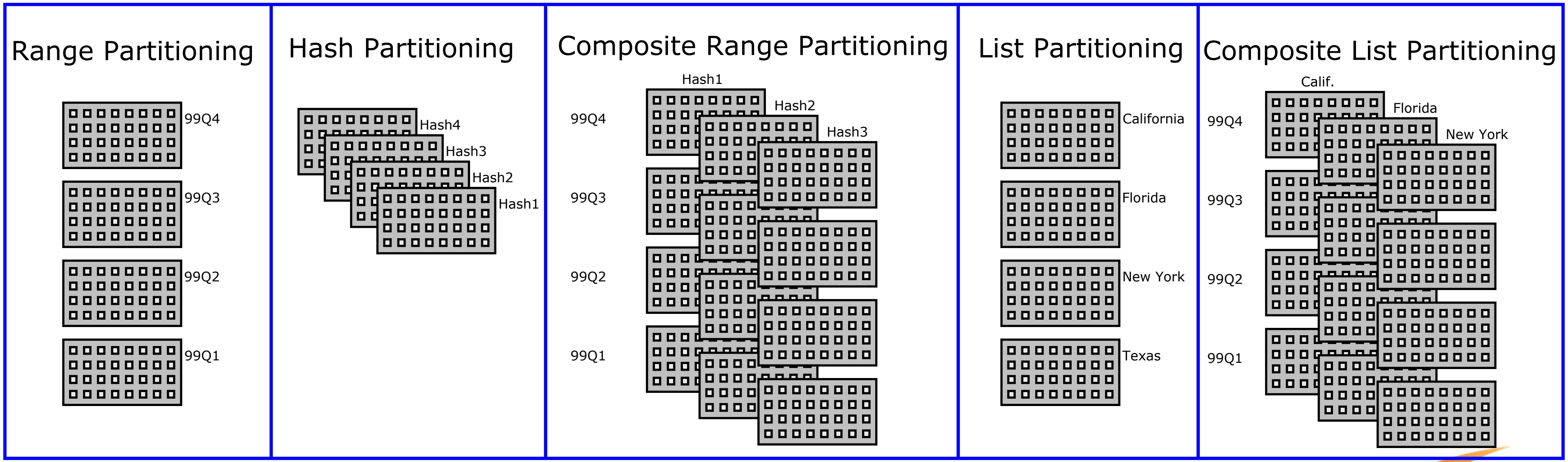
Partitioning

- **Partitioning tables and indexes allow Oracle and Teradata to store lots of data**
  - With Oracle, the process of choosing partitioning methods and partitioning keys is the balancing of query access path, performance, and data load requirements
    - The administrator explicitly manages the partitioning constraints and their relationship to disk storage
  - With Teradata, the hash partitioning algorithm is very good at evenly distributing (loading) data partitions and is the basis for high performance data access and ease of user access

Partitioning Columns are Chosen for Even Data Distribution in Both Teradata and Oracle

Oracle Partitioning Options

- **Administrator designed and implemented**
  - Oracle partitioning decisions - decide on methods as well as partitioning columns, set up the disk environment
  - Partitioning scheme based on trade-off between table maintenance, manageability, and query performance



Oracle10g Release 1 DDL to Create Partitioned Tables

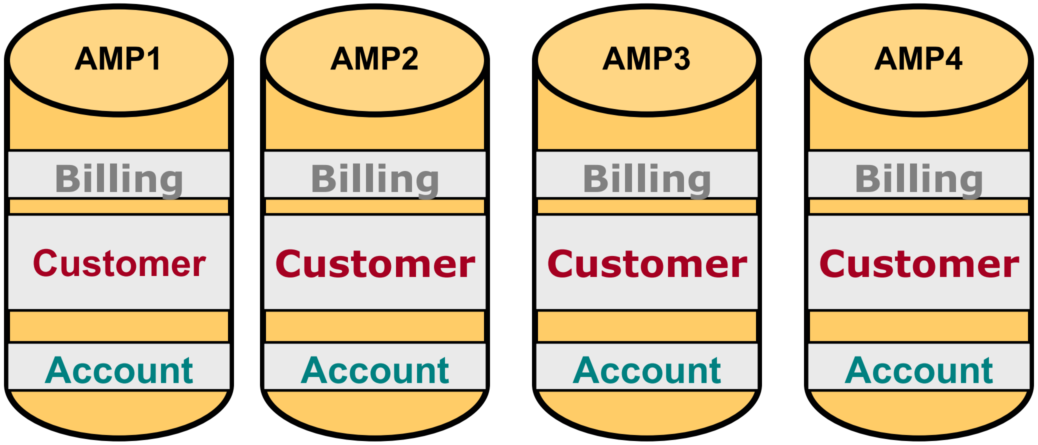
```
create table lineitem(
l_shipdate ,
l_orderkey NOT NULL,
l_discount NOT NULL,
l_extendedprice NOT NULL,
l_suppkey NOT NULL,
l_quantity NOT NULL,
l_returnflag ,
l_partkey NOT NULL,
l_linestatus ,
l_tax NOT NULL,
l_commitdate ,
l_receiptdate ,
l_shipmode ,
l_linenumber NOT NULL,
l_shipinstruct ,
l_comment
)
pctfree 1
pctused 99
initrans 10
storage (initial 260m next 260m freelist
groups 4 freelists 99)
parallel
nologging
partition by range
(l_shipdate)
subpartition by
hash(l_partkey)
subpartitions 16
(
partition item1 values less than
(to_date('1992-01-01','YYYY-MMDD'))
store in (ts_l1,ts_l2,ts_l3,ts_l4)
,
partition item2 values less than (to_date('1992-02-01','YYYY-MMDD'))
store in (ts_l5,ts_l6,ts_l7,ts_l8)
,
partition item3 values less than (to_date('1992-03-01','YYYY-MMDD'))
store in (ts_l9,ts_l10,ts_l11,ts_l12)
,
partition item4 values less than (to_date('1992-04-01','YYYY-MMDD'))
store in (ts_l13,ts_l14,ts_l15,ts_l16)
,
.
.    - Omitted 79 more partitioning clauses -
.
partition item84 values less than (MAXVALUE)
store in (ts_l333,ts_l334,ts_l335,ts_l336) )

as select
l_shipdate ,
l_orderkey ,
l_discount ,
l_extendedprice ,
l_suppkey ,
l_quantity ,
l_returnflag ,
l_partkey ,
l_linestatus ,
l_tax ,
l_commitdate ,
l_receiptdate ,
l_shipmode ,
l_linenumber ,
l_shipinstruct ,
l_comment
from l_et;
```

84 Partitioning Clauses

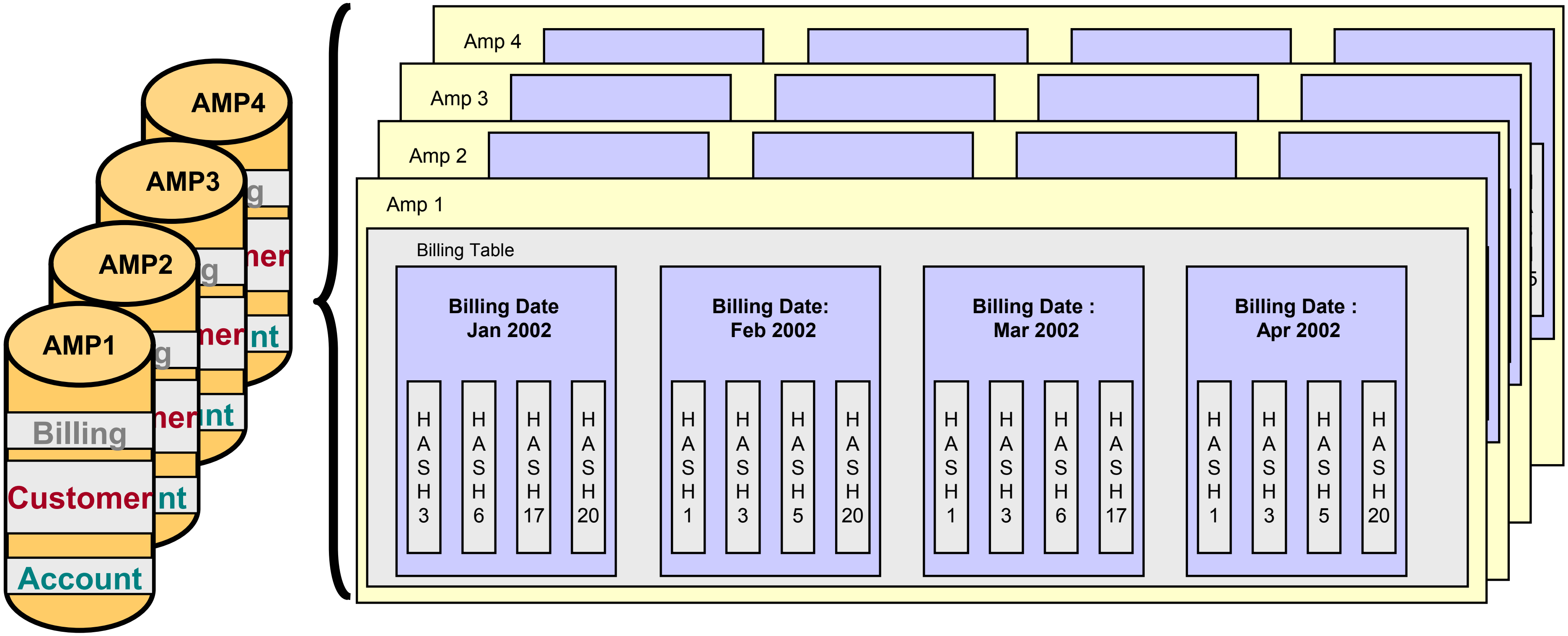
One Table =
84 Partitioning Clauses
336 data files

Partitioning Comparisons - Teradata

- **Teradata partitioning is a fact of the system with hash data distribution based on primary index (partitioning) column values and system managed disk**
  - DBA chooses the columns on which to partition
  - Teradata manages row placements and storage
- **Every unit of parallelism (AMP) may store an equal part of each table**



Teradata Partition Primary Index



PPI Partitioning is Powerful and Flexible

```
CREATE TABLE SalesByMonth
(storeid INTEGER NOT NULL,
productid INTEGER NOT NULL,
salesdate DATE FORMAT 'yyyy-mm-dd' NOT NULL,
totalrevenue DECIMAL(13,2),
totalsold INTEGER,
note VARCHAR(256))
UNIQUE PRIMARY INDEX (storeid, productid, salesdate)
PARTITION BY EXTRACT(MONTH FROM salesdate);
```

```
,productid INTEGER NOT NULL
,orderdate DATE FORMAT 'yyyy-mm-dd' NOT NULL
,totalorders INTEGER)
PRIMARY INDEX (storeid, productid)
PARTITION BY CASE_N(totalorders < 100, totalorders < 1000,
NO CASE, UNKNOWN);
```

```
productid INTEGER NOT NULL BETWEEN 0 and 999,
salesdate DATE FORMAT 'yyyy-mm-dd' NOT NULL,
totalrevenue DECIMAL(13,2),
totalsold INTEGER,
note VARCHAR(256))
UNIQUE PRIMARY INDEX (storeid, productid, salesdate)
PARTITION BY storeid*1000 + productid + 1;
```

Teradata Large Table DDL

```
CREATE MULTISET TABLE LINEITEM,
DATABLOCKSIZE=112
KILOBYTES
(
L_ORDERKEY DECIMAL (15,0) not null
,L_PARTKEY INTEGER not null
,L_SUPPKEY INTEGER not null
,L_LINENUMBER INTEGER not null
,L_QUANTITY DECIMAL(15,2) not null
,L_EXTENDEDPRICE DECIMAL(15,2) not null
,L_DISCOUNT DECIMAL(15,2) not null
,L_TAX DECIMAL(15,2) not null
,L_RETURNFLAG CHAR(1) CASESPECIFIC not null
,L_LINESTATUS CHAR(1) CASESPECIFIC not null
,L_SHIPDATE DATE FORMAT 'yyyy-mm-dd' not null
,L_COMMITDATE DATE FORMAT 'yyyy-mm-dd' not null
,L_RECEIPTDATE DATE FORMAT 'yyyy-mm-dd' not null
,L_SHIPINSTRUCT CHAR(25) CASESPECIFIC not null
,L_SHIPMODE CHAR(10) CASESPECIFIC not null
,L_COMMENT VARCHAR(44) CASESPECIFIC not null)
PRIMARY INDEX (l_orderkey) PARTITION BY
(RANGE_N
( L_ShipDate Between
* AND DATE '1992-12-31',
DATE '1993-01-01' AND DATE '1993-12-31',
DATE '1994-01-01' AND DATE '1994-12-31',
DATE '1995-01-01' AND DATE '1995-12-31',
DATE '1996-01-01' AND DATE '1996-12-31',
DATE '1997-01-01' AND DATE '1997-12-31',
DATE '1998-01-01' AND * ));
```

One Table = 1 Partitioning Clause Defining 7 Partitions in Database Managed Storage

Data Types

Data Types in Oracle and Teradata

Oracle

- CHAR
- VARCHAR2
- NCHAR
- NCHAR2
- NUMBER
- LONG
- LONGRAW
- RAW
- DATE
- BLOB
- CLOB
- NCLOB
- BFILE
- ROWID
- UROWID

Teradata

- CHAR
- VARCHAR
- CHAR VARYING
- LONG VARCHAR
- NUMERIC
- DECIMAL
- DOUBLE PRECISION
- FLOAT
- INTEGER
- SMALLINT
- BYTEINT
- BYTE
- VARBYTE
- GRAPHIC
- VARGRAPHIC
- LONG VARGRAPHIC
- DATE
- REAL

Translating Common Oracle to Teradata Data Types

Oracle	**Teradata**
Varchar2	Varchar
Number(m,n)	DECIMAL(m,n) or NUMERIC(m,n)
Date (includes time)	Date (does not include time)
Date (time portion)	Time
Date	Timestamp

Index Comparisons

Index Usage Comparisons

- **Oracle solutions traditionally have relied heavily on indexes**
  - OLTP workloads required fast access paths to few rows
  - Decision support solutions often use indexes where tactical queries with OLTP-like response time requirements are given more emphasis than throughput performance
- **Teradata solutions have traditionally not used lots of indexes**
  - Teradata's efficient parallel architecture emphasizes throughput performance requirements - a result of it's DSS background

Creating Indexes

- **Similarities between Teradata and Oracle:**
  - Indexes take up space on disk
  - Indexes can be unique and non-unique
  - Indexes and secondary indexes provide alternate ways to access data
- **Differences:**
  - Teradata indexes are not in B-tree structure
    - Hash Subtables
  - Teradata automatically partitions indexes across the AMPs
  - Teradata uses a Primary Index for each table

Oracle Indexes

- **Created in Tablespaces**
- **B-tree or Bitmap Organization**
- **Unique/Non-Unique**
- **Cluster**
- **Function Based**
- **Partitioned**
  - Global/Local, Prefixed/Non-Prefixed

Teradata Primary Indexes (PI)

- **Created through the hashing of its assigned attributes giving it a unique value based on one of 65536 hash-IDs and a uniqueness code**
- **Direct access path to the base table row**
  - Does not require any additional physical structures since index value is hashed into the table and is placed on to disk in hash order
  - Can either be UNIQUE or NON-UNIQUE
- **Access to the base table through the hashing algorithm is a single AMP operation**
- **Since data are distributed among partitions based on the PI, no table can be created without one**

Teradata Secondary Indexes

- **Unique (USI) and Non-Unique Secondary Indexes (NUSI)**
  - USI ensures uniqueness of attributes, especially if the primary index is non-unique
- **Built through a hash map into a subtable**
  - Row ID of the base table is obtained from the subtable and the data retrieved
- **Access Path Impact**
  - USI Operations are most often two AMP operations
  - NUSIs provide all AMP access
    - Can be used to greatly reduce the amount of disk I/O is performed

Materialized Views

Oracle Materialized View

- **Pre-Computed Summary Table**
- **Pre-Joined Tables**
- **Combination of Joined Tables and Aggregates**
- **Kept in-sync with underlying tables only if configured for continuous update**
  - Logs Used to Track Underlying Tables Changes
  - Logs Can be Applied Immediately or At a Later Time

Teradata Join Index

- **A Join Index is created by executing a SELECT statement that retrieves data from tables to satisfy a query and then stores the Join Index structure on disk.**
  - A user query can be satisfied from the Join Index rather than by accessing the table.
  - A Join Index can contain a subset of rows from a single table or it can contain the results from joining several tables together

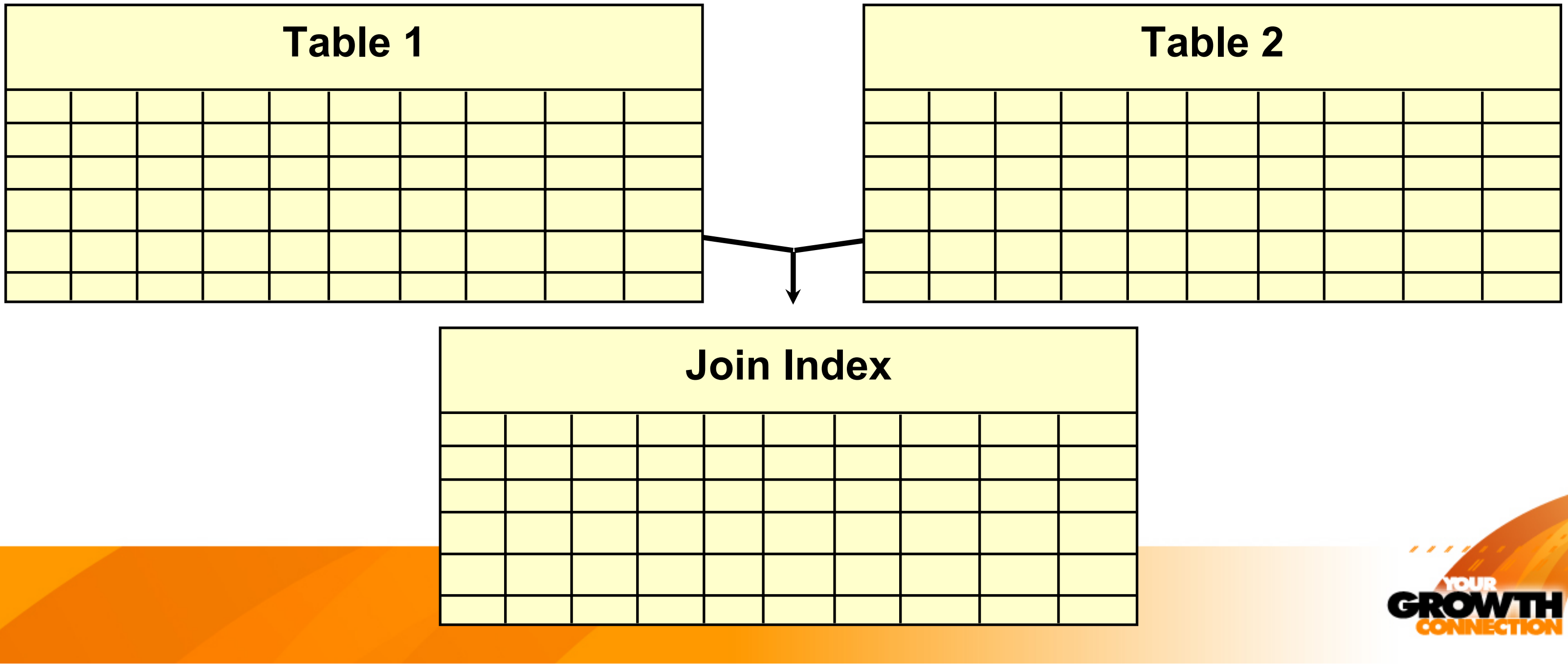


Table 1

Table 2

Join Index

Join Index (JI)

- **A system maintained table based on the pre-JOINed data from 1 or more base tables**
  - Does not provide an access path to the base tables used to create it
- **Use is determined by the optimizer (not directly by the user)**
  - Can not be directly referenced in a query
  - Automatically maintained by the database as the base tables change
- **Has a primary index that can be accessed as a single AMP operation**
- **Pre-JOINing data from base tables takes the work out of the queries that would normally use the base tables**
- **If JI contains aggregated data, large sorts, SUMs and COUNTs are removed from queries that would normally have to do these operations**

Three Types of Join Indexes

- **Multi-table Join Index**
  - Provides the ability to join data into the index from multiple tables using either INNER or OUTER joins
- **Single table Join Index**
  - Allows to access the same columns or a subset of columns as the base table but using an alternative primary index
  - It is not necessarily a self-join and should not be considered as such
- **Aggregate Join Index**
  - Provides the ability to build database maintained summary tables

Parallelism Comparisons

Oracle10g Parallel Processing

- **Parallel Query (Dynamic Parallelism) is an option**
  - Parallelism is a limited resource that must be managed
  - Insufficient parallel degrees available means serial execution or system canceled queries
    - Parallel

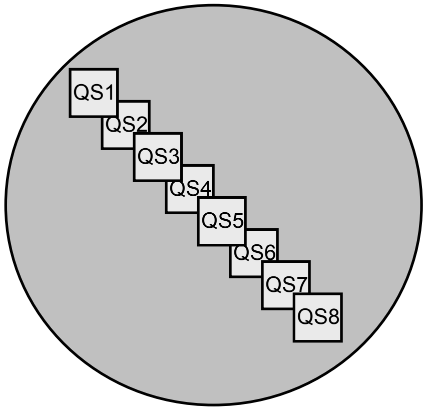
      Table scans, Index scans/reads, Joins, Sorts, Aggregations
    - Not Parallel

      Joins (last level), Final Sort, Final aggregation
  - Amount of parallelism varies according to the resources available at runtime
- **Oracle parallel data access is enabled and managed by the administrator**
  - Partitioning
  - Degrees of Parallelism
  - Number of Query Servers in System
  - Number of Users in the System
  - System, Table, and User Settings

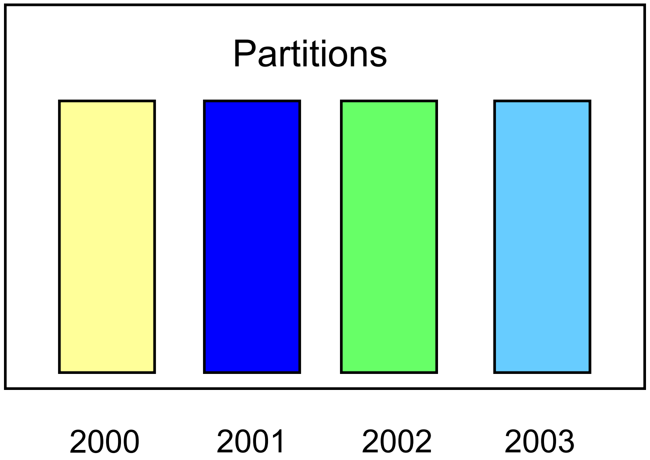
Dynamic Parallelism

Selects,Inserts, Updates, Deletes (Oracle10g Rel. 1)

System Parallelism

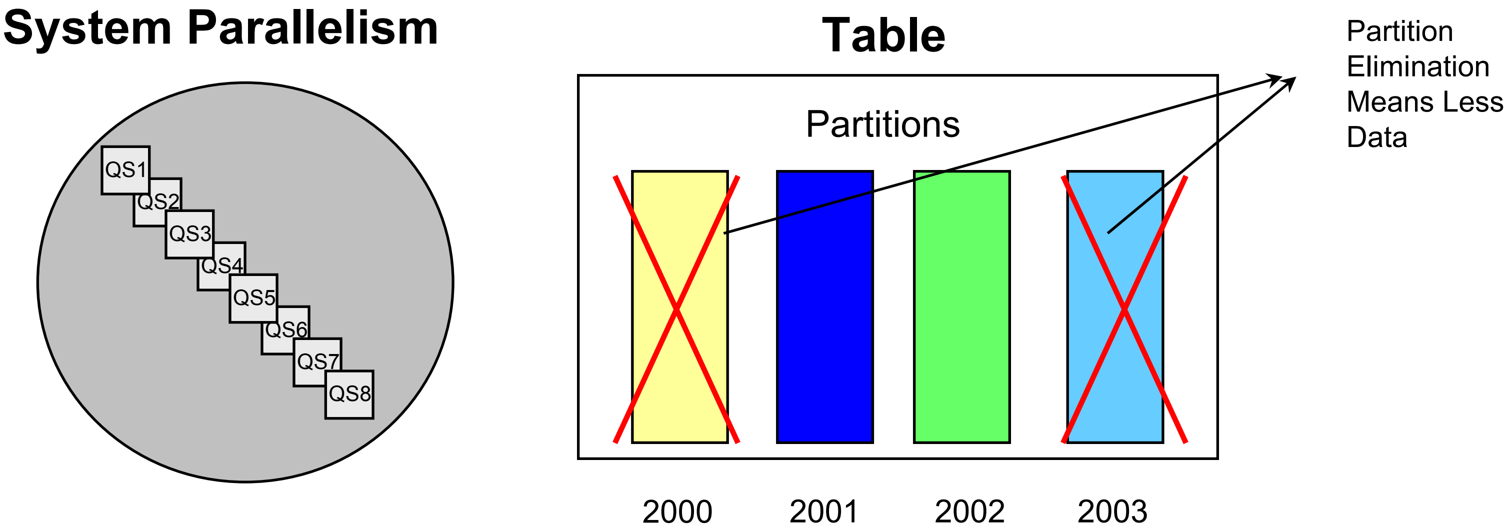


Table



Dynamic Parallelism

Selects,Inserts, Updates, Deletes (Oracle10g Rel. 1)

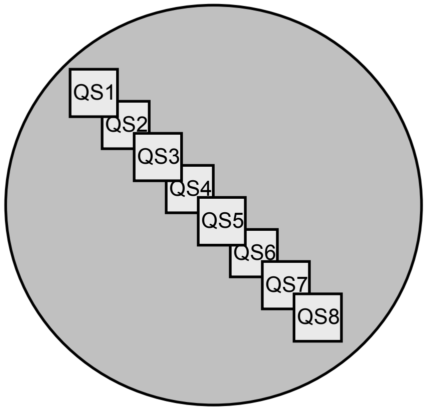


Query only asks for data between “2001” and “2002”

Dynamic Parallelism

Selects,Inserts, Updates, Deletes (Oracle10g Rel. 1)

System Parallelism



QS1

QS2

QS3

QS4

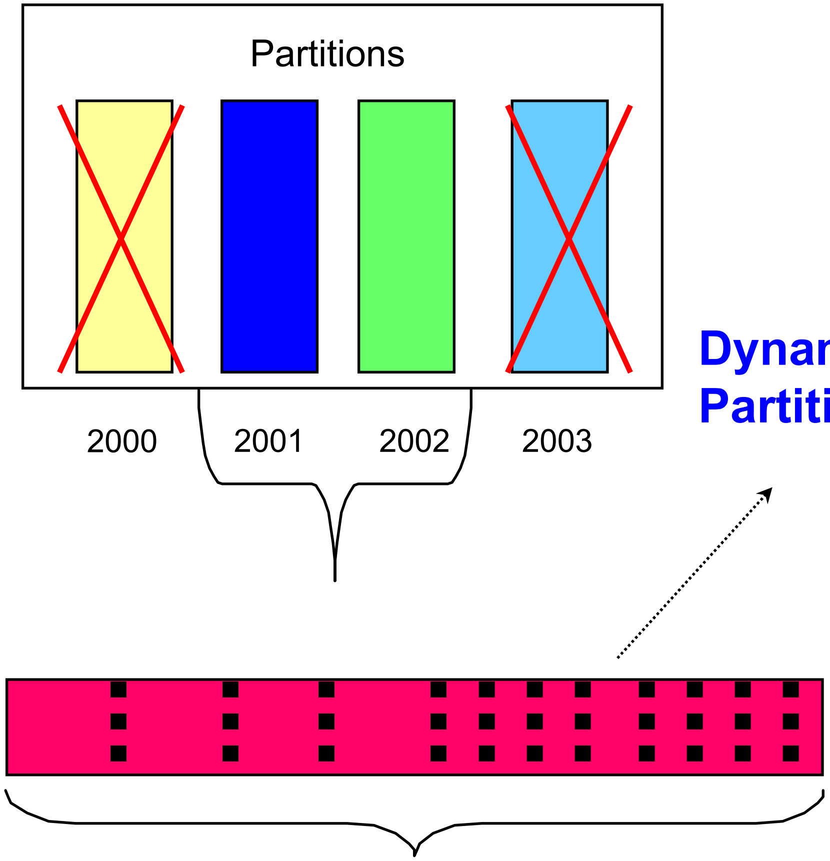
QS5

QS6

QS7

QS8

Table



Partitions

2000

2001

2002

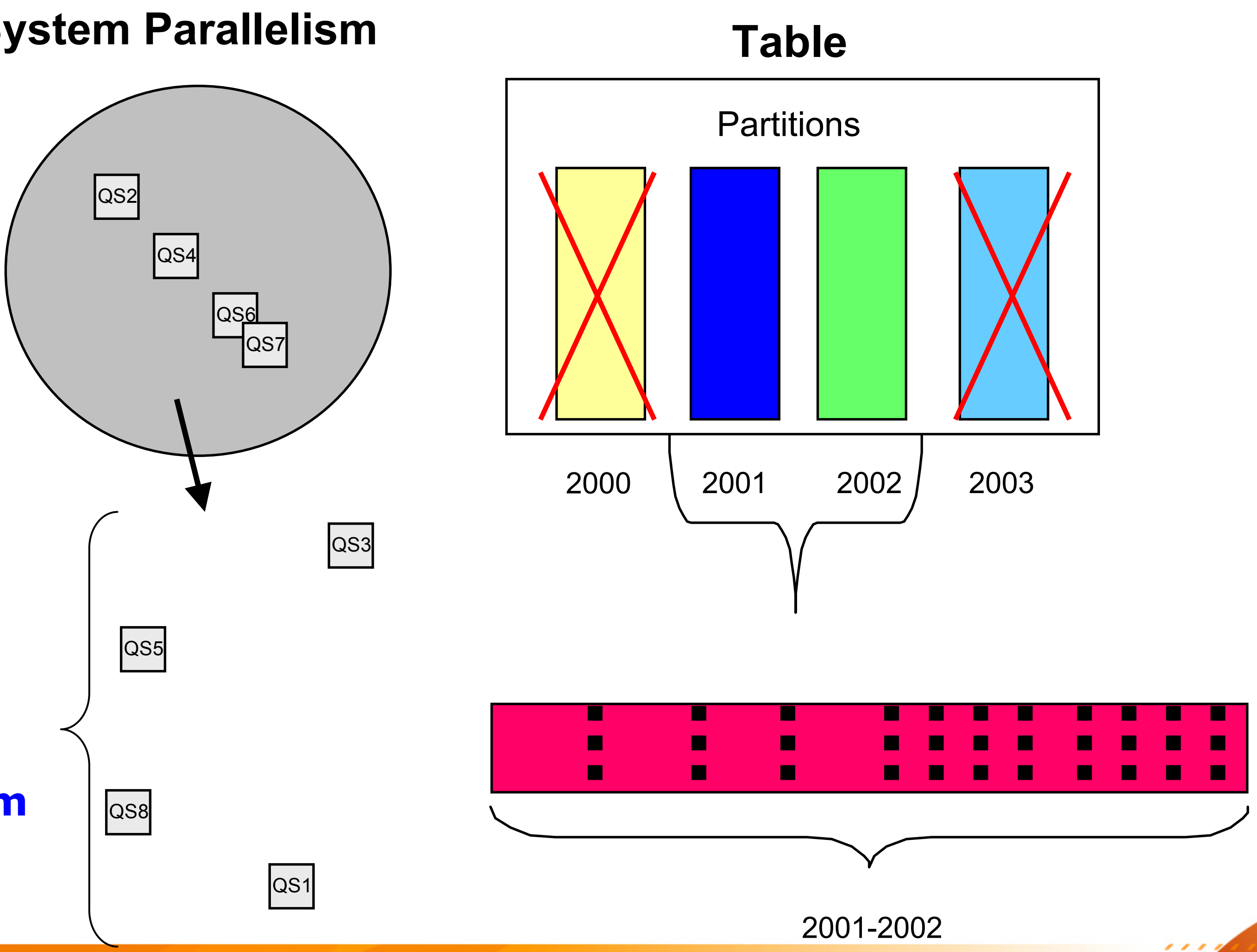
2003

Dynamic Partitioning

2001-2002

Dynamic Parallelism

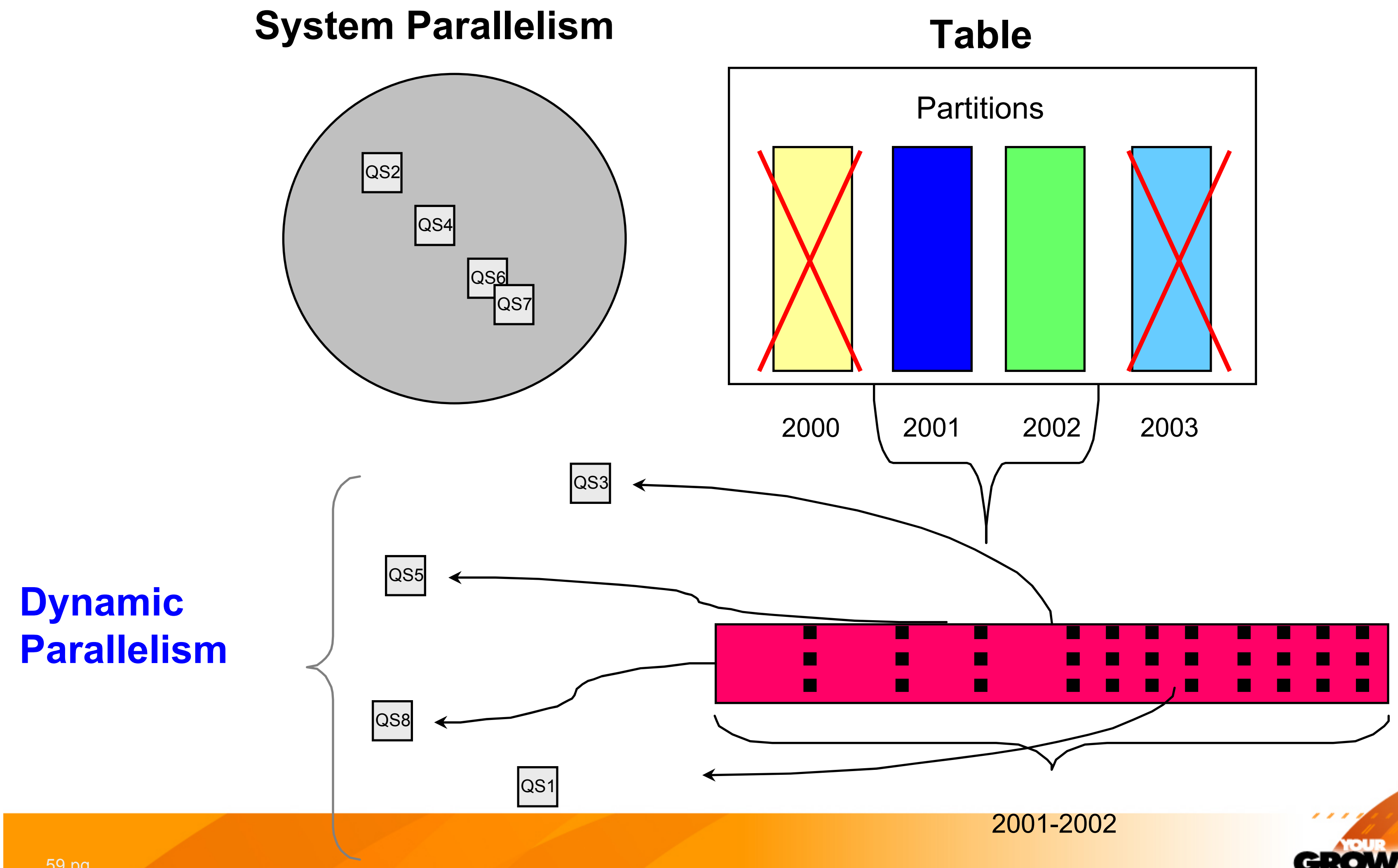
Selects,Inserts, Updates, Deletes (Oracle10g Rel. 1)



Query assigned 4 units of parallelism

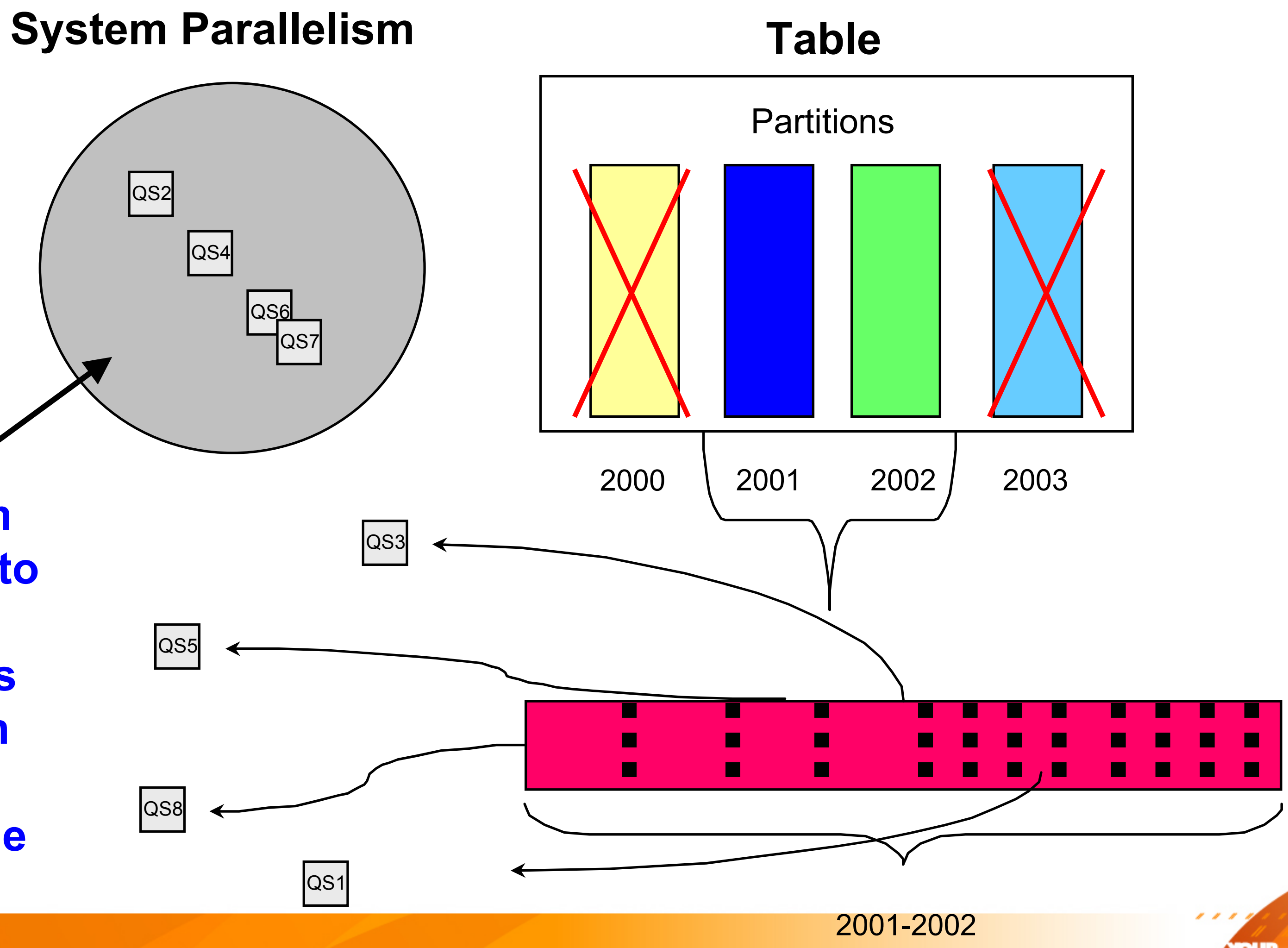
Dynamic Parallelism

Selects,Inserts, Updates, Deletes (Oracle10g Rel. 1)



Dynamic Parallelism

Selects,Inserts, Updates, Deletes (Oracle10g Rel. 1)



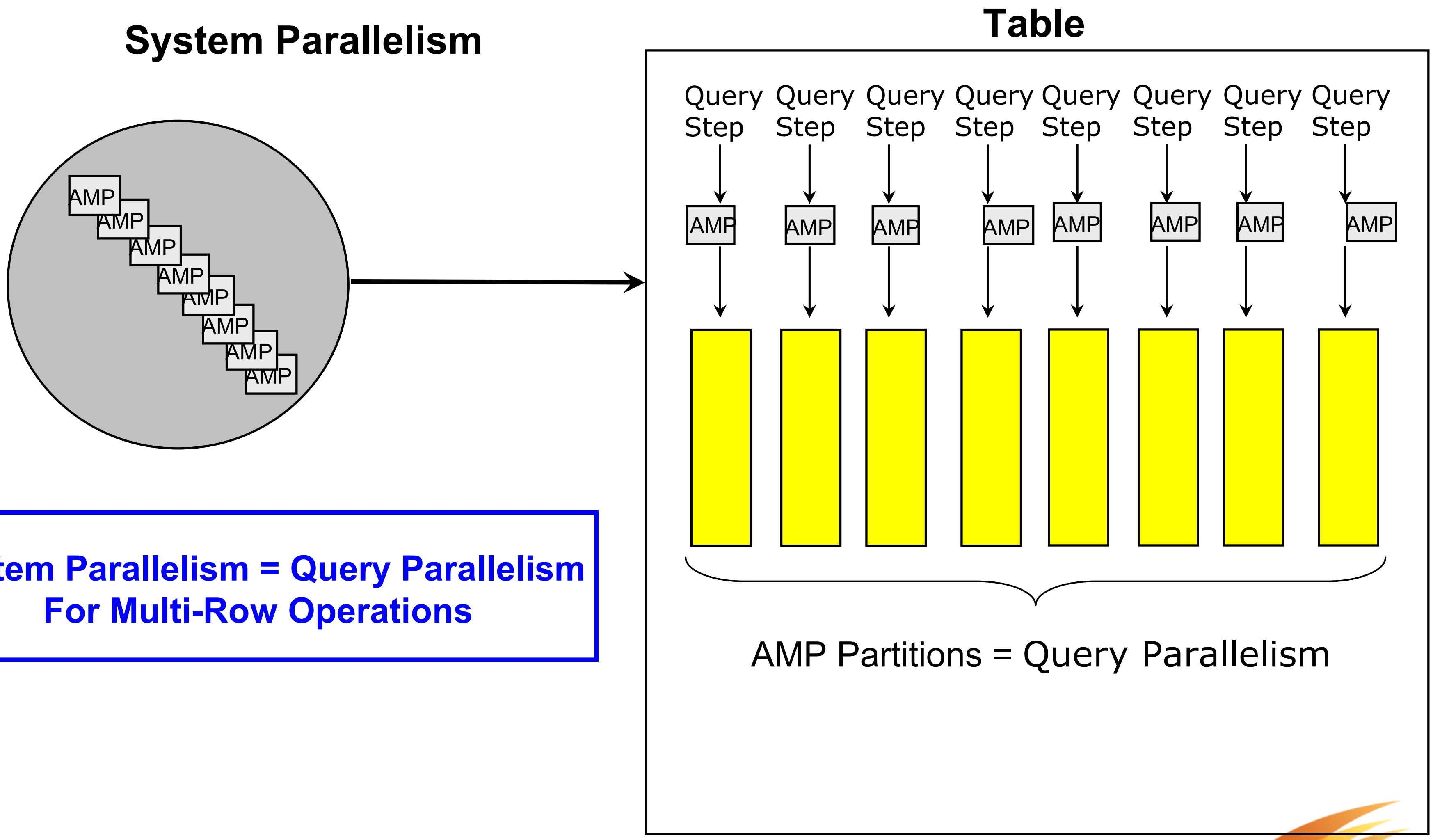
Parallelism allocation to queries means less parallelism for other users in the system

Teradata Parallel Data Access

- **Teradata Parallel Access is Defined By Teradata**
    - Database Managed Partitioning
    - Database Managed Degrees of Parallelism
    - System Configuration Based Parallelism
    - Parallelism Independent of Number of Users
- **Parallelism is Never Conditional**
    - You Don’t “Run Out” of Parallelism
    - Users Can’t Horde Parallelism So Others Can’t Use It

Teradata Pervasive Parallelism

SELECTS, INSERTS, UPDATES, DELETE, UTILITIES, etc.

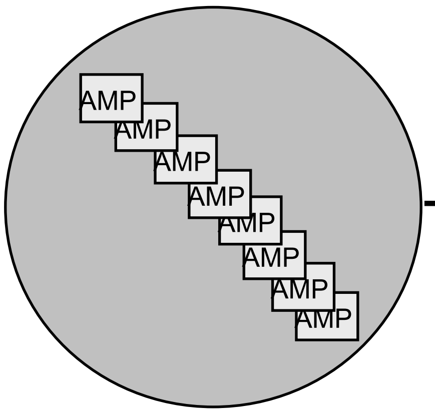


System Parallelism = Query Parallelism
For Multi-Row Operations

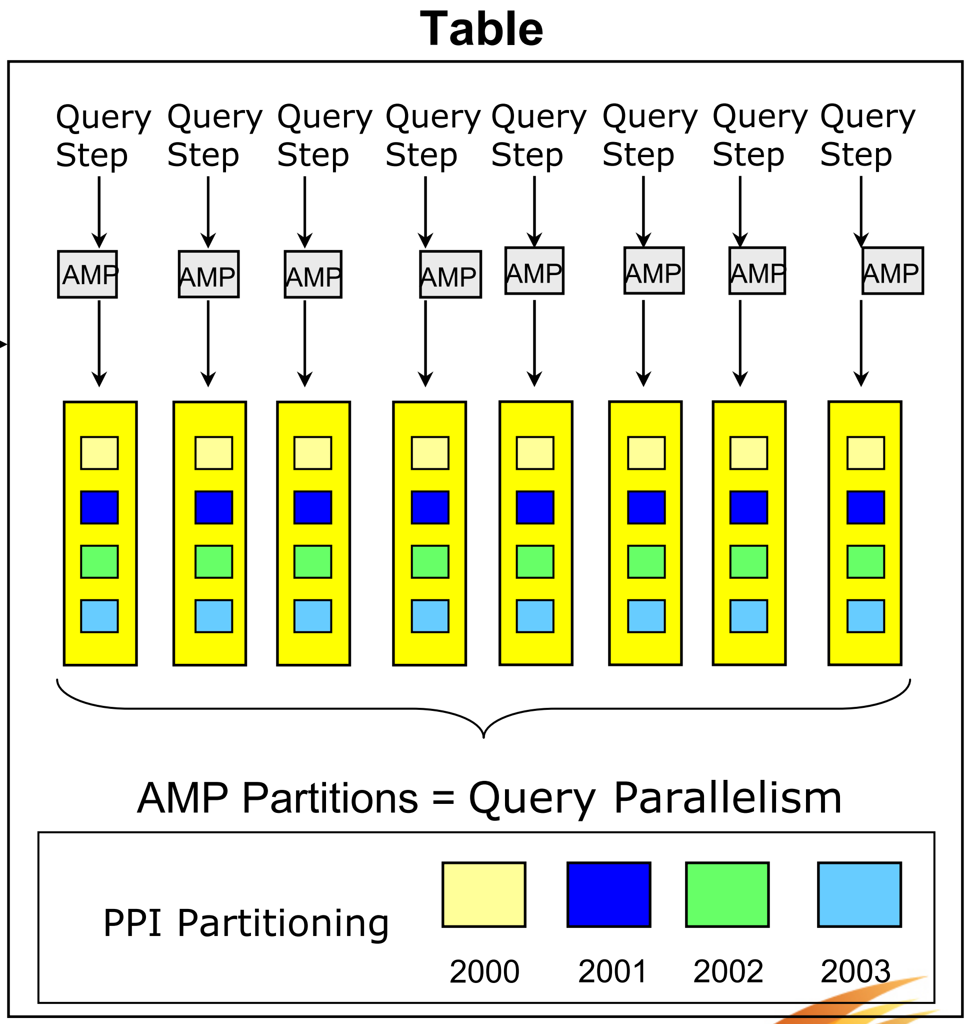
Teradata Pervasive Parallelism

SELECTS, INSERTS, UPDATES, DELETE, UTILITIES, etc.

System Parallelism

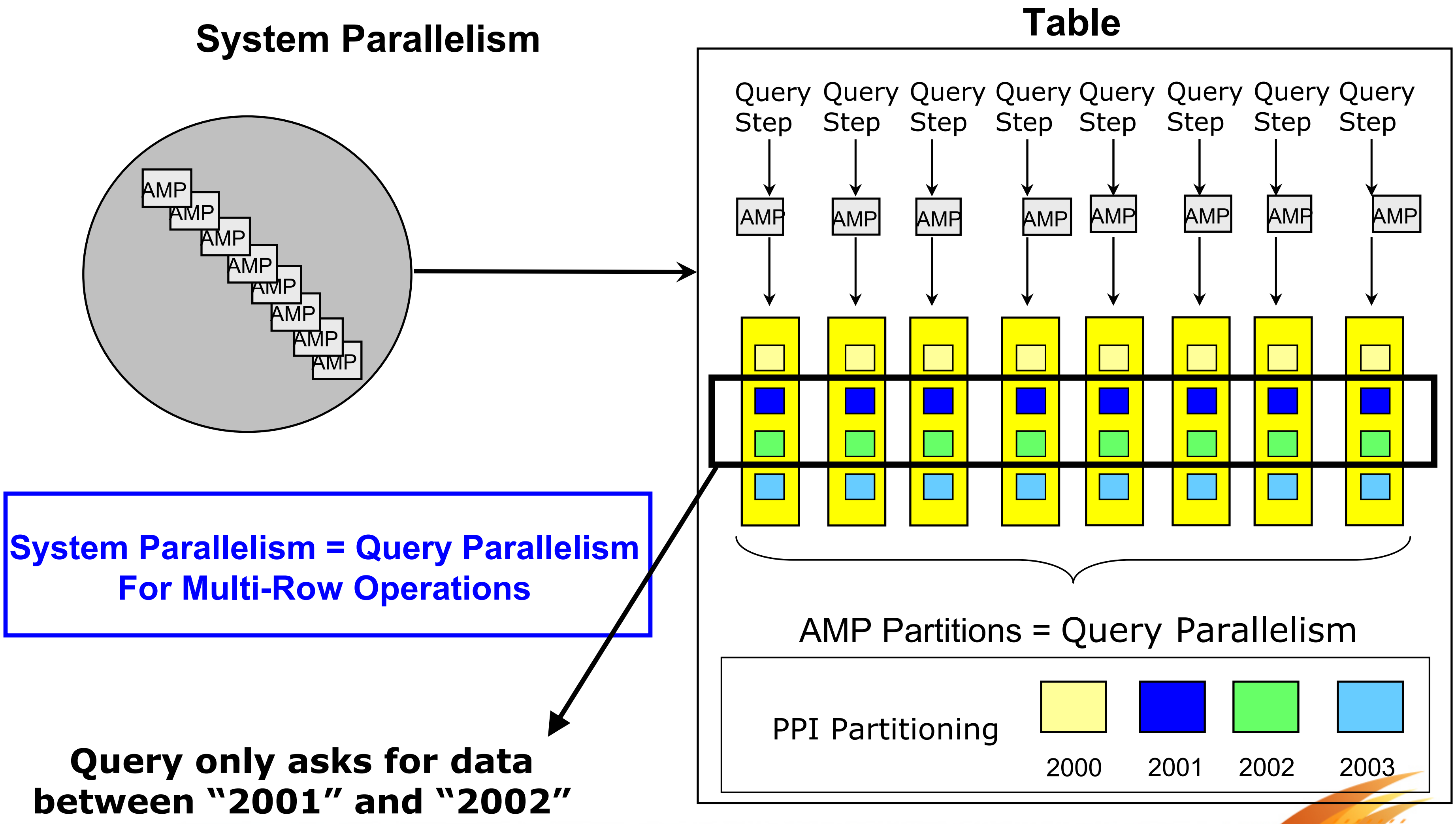


System Parallelism = Query Parallelism For Multi-Row Operations



Teradata Pervasive Parallelism

SELECTS, INSERTS, UPDATES, DELETE, UTILITIES, etc.



System Parallelism = Query Parallelism For Multi-Row Operations

Query only asks for data between "2001" and "2002"

User Data Maintenance Utilities

Utilities to Load and Update Data

Oracle10g

SQL*LOADER

- **Conventional Path**
- **Direct Path**
- **Manually Parallelized through Multiple Load Processes**
  - Multiple Utility Output Files
  - Multiple Input Files

External Table

- **Non-database files defined to the database with DDL so they can be accessed with SQL**
- **Data source for CREATE TABLE … AS SELECT…**
- **You set data access/load parallelism through DDL/SQL**

Data Pump

- **Import/Export Utility**
- **Parallel**
- **Dependent upon Streams**

Teradata

FASTLOAD

- **One Load Process**
  - Can read multiple input files
  - One script, One Set of Output files
  - Optimized, parallelized, block oriented operations
  - Checkpoint restartable
- **Teradata Manages Utility Parallelism Automatically**

Multiload

- **Inserts, Updates, “Upserts”, Deletes**
- **Checkpoint restartable**

TPump

- **Bulk Transactions - Inserts, Updates, Deletes**
- **“Trickled In” to Table Over Time**
  - Allows throttling/controlling the Rate of Change
- **Checkpoint restartable**

Summary

Summary

- **Differences in database architecture result in the differences between the ways Teradata and Oracle databases are created and managed**
- **Teradata is configured and delivered to you ready to use**
- **Many system management tasks are performed by Teradata**
  - Teradata manages disk storage and how it is used by tables and indexes
  - Teradata parallelism is built in
- **Teradata enables you to focus on business requirements – not on system management tasks**
- **You can use your Oracle knowledge to understand and use Teradata**