POZNAN UNIVERSITY OF TECHNOLOGY

Data Warehouse Appliance: Main Memory Data Warehouse

Robert Wrembel
Poznan University of Technology
Institute of Computing Science
Robert.Wrembel@cs.put.poznan.pl
www.cs.put.poznan.pl/rwrembel

Lecture outline

Teradata Data Warehouse Appliance

SAP Hana

IBM Netezza

Oracle Exadata

EMC Greenplum Appliance

DW Appliance

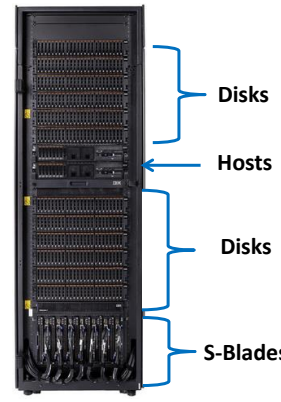
- Self-contained integrated solution stack of
  - hardware
  - operating system
  - RDBMS
  - storage
- Optimized for data warehouse workloads
- Comes out of the "box" preconfigured and tuned
- Hardware is designed to work with a particular software and the software is tuned to work with this hardware

IBM appliance

- IBM PureData System for Analytics = Netezza
- Models
  - N1001
  - N2001
  - N3001
  - C1000
  - TwinFin12
  - TwinFin24
  - SailFish

IBM Netezza

- The key hardware components include the following:
  - hosts (servers)
  - storage arrays (disks)
  - snippet blades (S-Blades = Snippet Processing Units - SPUs)
    - each S-Blade owns several disks which reside in a storage array within the same rack



IBM Netezza

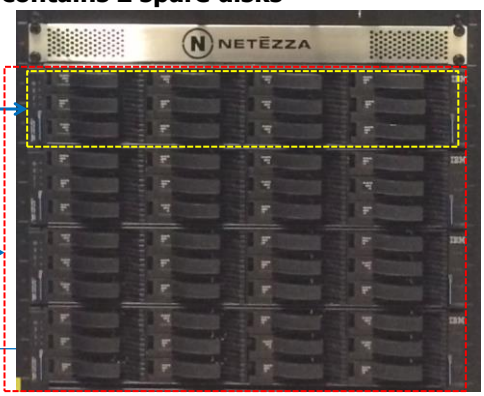
- Host
  - **Linux OS** (Red Hat Enterprise Linux)
  - administration and security
  - system monitoring
  - workload management
  - consolidating and returning query results
  - query optimization
  - data loading
  - data distribution to disks
  - one is active
  - one is spare (standby, backup)
  - failover from active to standby

IBM Netezza

- Storage array = storage group
  - composed of n disk enclosures
  - disk enclosure = 12 or 24 disks (depending on an appliance)
  - one appliance includes at least 1 storage group (array)
  - typically, 1 storage group contains 2 spare disks

Disk enclosure = 12 disks

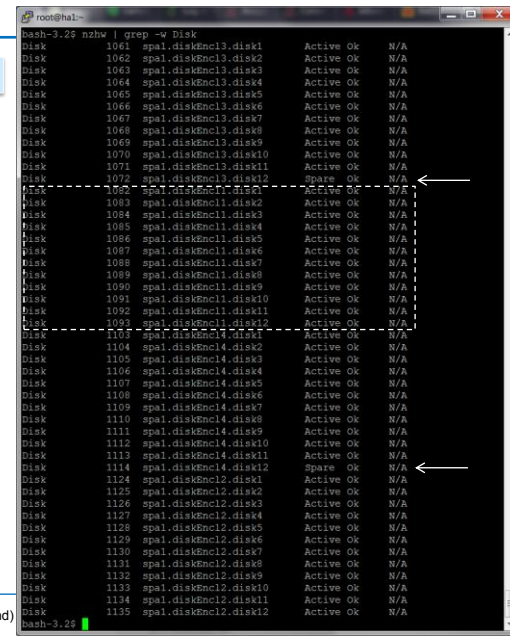
Storage array (4 disk enclosures)



IBM Netezza

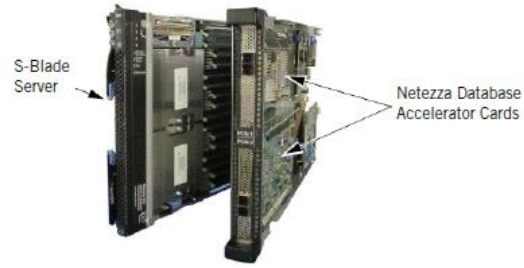
```
[nz@ha1 ~]$ nzhw | grep -w Disk
```

- 48 disks (2 spare)
- 4 enclosures with 12 disks each



IBM Netezza

- S-Blade
  - for processing data from disks
  - CPU + Netezza Database Accelerator card
  - contains: FPGA query engines, memory, and I/O
  - S-Blade manages its own disks



IBM Netezza

- A field-programmable gate array (FPGA)
  - an integrated circuit designed to be configured (programmed) after manufacturing
  - contains an array of programmable logic blocks, and reconfigurable interconnects that allow the blocks to be "wired together", like logic gates
  - the logic blocks can be configured to perform functions → based on a given input produce a given output
  - logic blocks include memory

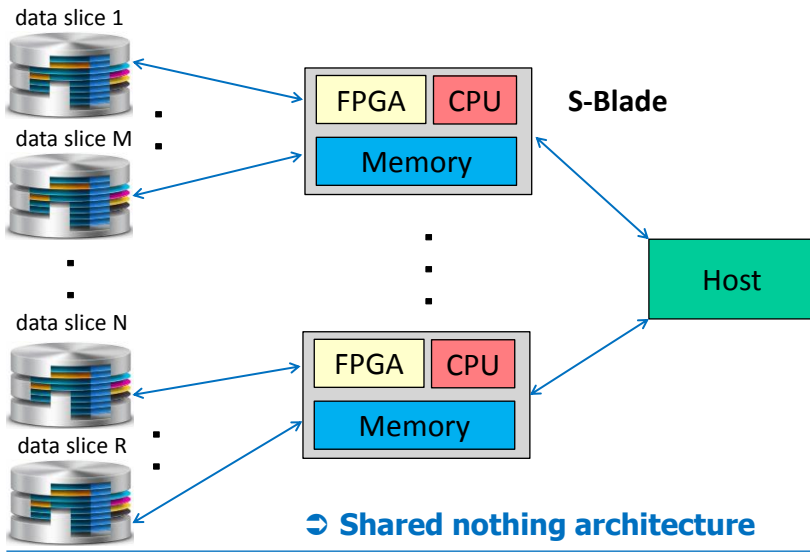
IBM Netezza

- **SPU identifiers**
  - **1007**
  - **1008**
  - **1009**
  - **1010**
  - **1011**
  - **1012**
  - **1013**

```
[nz@ha1 ~]$ nzstats show -type spu

HW ID Memory      Free Memory Used Memory % Used Memory
----- ----------- ----------- ----------- -------------
 1007 24599552 KB 21403648 KB  3195904 KB        12.99 %
 1008 24599552 KB 21404672 KB  3194880 KB        12.99 %
 1009 24600576 KB 21911552 KB  2689024 KB        10.93 %
 1010 24599552 KB 21978112 KB  2621440 KB        10.66 %
 1011 24600576 KB 21977088 KB  2623488 KB        10.66 %
 1012 24599552 KB 21976064 KB  2623488 KB        10.66 %
 1013 24599552 KB 21975040 KB  2624512 KB        10.67 %
```


IBM Netezza



- **Shared nothing architecture**

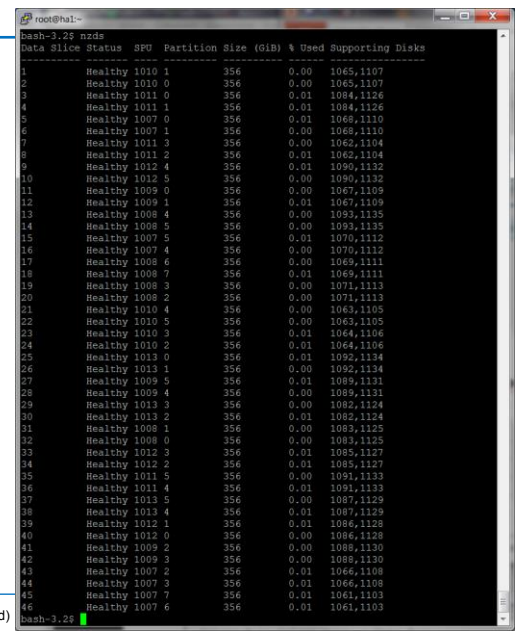
Data slice

- Data slice ⇨ a disk zone allocated for storing data of a table
- #data slices = #disks in an appliance
- Data are stored in extents
  - an extent is the smallest unit of disk allocation = 3MB
  - a data slice may be composed of multiple extents

IBM Netezza

- Data slices

```
[nz@ha1 ~]$ nzds
```



IBM Netezza

➲ # data slices = # active disks → 46

Data Slice	Status	SPU	Partition	Size (GiB)	% Used	Supporting Disks
5	Healthy	1007	0	356	0.01	1068,1110
6	Healthy	1007	1	356	0.00	1068,1110
15	Healthy	1007	5	356	0.01	1070,1112
16	Healthy	1007	4	356	0.00	1070,1112
43	Healthy	1007	2	356	0.01	1066,1108
44	Healthy	1007	3	356	0.01	1066,1108
45	Healthy	1007	7	356	0.01	1061,1103
46	Healthy	1007	6	356	0.01	1061,1103
13	Healthy	1008	4	356	0.00	1093,1135
14	Healthy	1008	5	356	0.00	1093,1135
17	Healthy	1008	6	356	0.00	1069,1111
18	Healthy	1008	7	356	0.01	1069,1111
19	Healthy	1008	3	356	0.00	1071,1113
20	Healthy	1008	2	356	0.00	1071,1113
31	Healthy	1008	1	356	0.00	1083,1125
32	Healthy	1008	0	356	0.00	1083,1125
11	Healthy	1009	0	356	0.00	1067,1109
12	Healthy	1009	1	356	0.01	1067,1109
27	Healthy	1009	5	356	0.01	1089,1131
28	Healthy	1009	4	356	0.00	1089,1131
41	Healthy	1009	2	356	0.00	1088,1130
42	Healthy	1009	3	356	0.00	1088,1130
1	Healthy	1010	1	356	0.00	1065,1107
2	Healthy	1010	0	356	0.00	1065,1107
21	Healthy	1010	4	356	0.00	1063,1105
22	Healthy	1010	5	356	0.00	1063,1105
23	Healthy	1010	3	356	0.01	1064,1106
24	Healthy	1010	2	356	0.01	1064,1106
..	.....	.	...	....	.....	

one data slice on 2 disks → replication

each SPU manages its own disks → no disk sharing between SPUs

Data distribution

➲ Table data are distributed into all available data slices

- chacking allocation

```
nzstats show -type tableDataSlice
```

```
SELECT DATASLICEID, COUNT(*)
FROM table_name GROUP BY DATASLICEID;
```

➲ Table data distribution into data slices

- hashing
- random (round-robin)

```
CREATE TABLE tab-name
(...)
DISTRIBUTE ON {(col1, ...) | RANDOM}
```

Data distribution

- Distribution key should
  - have a large number of distinct values, distributed evenly
  - not be used in WHERE

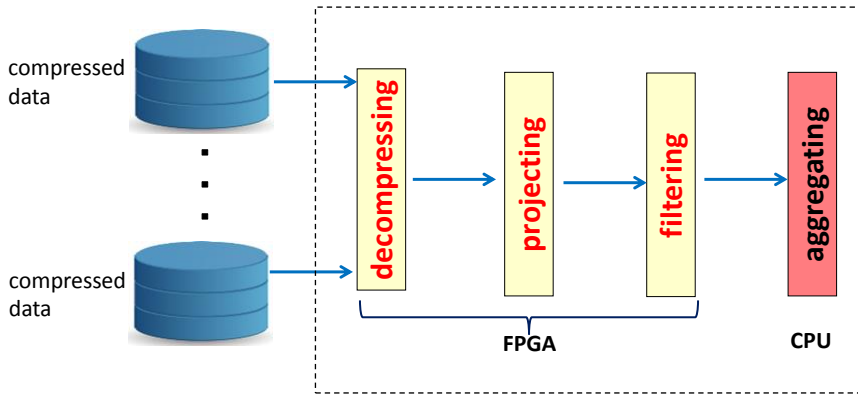
Data allocation

- Allocating rows from PK table - FK table for joins
- Case 1
  - PK rows are stored on different data slices than FK rows
  - at least one table (the smaller - PK) needs to be redistributed to S-Blades for joining with FK local data slices
- Case 2
  - colocated PK-FK tables (on the same data slice)
  - PF-FK rows can be joined locally at a given S-Blade

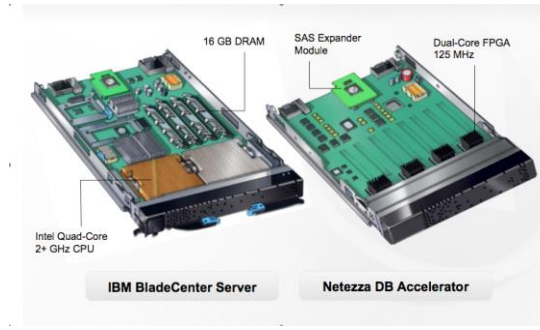
Disk mirroring

- All user data and temp space is mirrored to another disk (2 copies of the same data item)
- Automatic failover
  - transactions continue on a mirror disk
  - failed drives are automatically regenerated on a spare disk

S-Blade



S-Blade



- S-Blade tasks
  - decompression
  - data filtering
  - data projection
  - SQL operations
  - joins
  - aggregations
  - sorts
  - analytical algorithms (data mining, prediction)

Interfaces

- Moving data in / out
  - SQL
  - JDBC
  - ODBC
  - OLE-DB
- ETL tools
  - DataStage
  - Informatica
  - AbInitio
  - ...
- Analytics
  - Cognos
  - MicroStrategy
  - SAP Business Objects
  - ...

SQL

ALTER AGGREGATE	CREATE DATABASE	DROP HISTORY CONFIGURATION	SET AUTHENTICATION
ALTER CATEGORY	CREATE EXTERNAL TABLE	DROP PROCEDURE	SET CATALOG
ALTER COHORT	CREATE FUNCTION	DROP KEYSTORE	SET CONNECTION
ALTER DATABASE	CREATE GROUP	DROP LIBRARY	SET HISTORY CONFIGURATION
ALTER FUNCTION	CREATE HISTORY CONFIGURATION	DROP SCHEMA	SET SCHEMA
ALTER GROUP	CREATE PROCEDURE	DROP SECURITY LEVEL	SET SESSION
ALTER HISTORY CONFIGURATION	CREATE KEYSTORE	DROP SEQUENCE	SET SYSTEM DEFAULT
ALTER PROCEDURE	CREATE LIBRARY	DROP SYNONYM	SET TRANSACTION
ALTER KEYSTORE	CREATE MATERIALIZED VIEW	DROP TABLE	SHOW
ALTER LIBRARY	CREATE SCHEMA	DROP USER	SHOW AGGREGATE
ALTER SCHEMA	CREATE SECURITY LEVEL	DROP VIEW	SHOW AUTHENTICATION
ALTER SECURITY LEVEL	CREATE SEQUENCE	EXECUTE	SHOW CATEGORY
ALTER SEQUENCE	CREATE SYNONYM	EXECUTE AS	SHOW COHORT
ALTER SESSION	CREATE TABLE	EXECUTE PROCEDURE	SHOW CONNECTION
ALTER SYNONYM	CREATE TABLE AS	EXPLAIN	SHOW CRYPTO KEY
ALTER TABLE	CREATE USER	EXTRACT	SHOW FUNCTION
ALTER USER	CREATE VIEW	GENERATE STATISTICS	SHOW HISTORY CONFIGURATION
ALTER VIEW	DELETE	GRANT	SHOW KEYSTORE
BEGIN	DROP AGGREGATE	GROOM TABLE	SHOW LIBRARY
CALL	DROP CATEGORY	INSERT	SHOW PROCEDURE
COMMENT	DROP COHORT	RESET	SHOW SECURITY LEVEL
COMMIT	DROP CONNECTION	REVERT	SHOW SYSTEM DEFAULT
CREATE AGGREGATE	DROP CRYPTO KEY	REVOKE	TRUNCATE
CREATE CATEGORY	DROP DATABASE	ROLLBACK	UPDATE
CREATE COHORT	DROP FUNCTION	SELECT	USER
CREATE CRYPTO KEY	DROP GROUP	SET	

UDX

- **User Defined eXtensions**
  - **programs (implemented in C++) callable from SQL**
  - **precompiled for Netezza**
  - **executed in parallel on S-blades**
- **Next generation of UDX → Analytic Executables (AE)**
  - **implemented in C, C++, Java, Fortran, Python, Perl, R**
  - **by default unavailable → must be explicitly installed from the INZA package**

Analytic Executables

```
isntalling INZA package
./inzaPackageInstaller.sh

installing additional libraries
$ ./main_installer.sh install /export/home/nz/instalation

installing UDF
$ ./compile.sh IRIS

using from SQL

sql=> SELECT * FROM example_train, TABLE WITH FINAL (
PUT_HT_TRAIN( attrib_a, attrib_b, class) );

sql=> SELECT * FROM example_test, TABLE WITH FINAL (
PUT_HT_PREDICT_SEQ( attrib_a, attrib_b) );
```


Netezza: data structures

- Materialized view (MV)
- Zone map (ZM)
- Cluster based table (CBT)

Netezza: MV

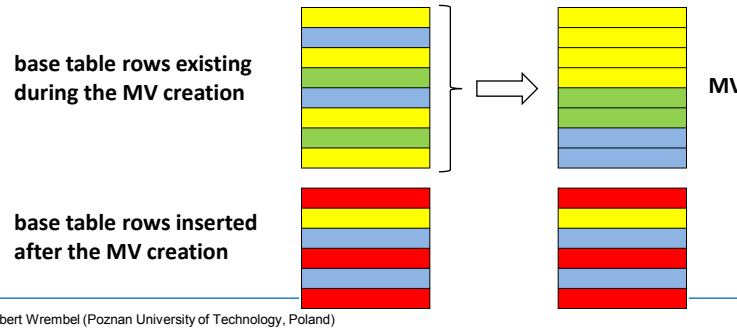
- Used for query rewriting
- Stored as a table
- Divided into data slices that are co-located on the same disk as the corresponding base table data slices

```
CREATE MATERIALIZED VIEW v-name AS
SELECT ... FROM tab-name [ORDER BY ...]
```

- Some restrictions
  - only one table in the FROM clause
  - the WHERE clause cannot be used
  - the columns in the projection list must be columns ⇨ not allowed expressions (aggregates, mathematical operators, SQL functions, DISTINCT, ...)
  - the columns in the optional ORDER BY clause must be one or more columns in the projection list

Netezza: MV

- Inserting rows into a base table
  - automatic refreshing
  - new rows are appended to the MV ⇨ two areas in the MV:
    - the sorted records generated when the view was created
    - the unsorted records that have been inserted into the base table after the MV was created
  - resorting by manual refreshing



Netezza: MV

- Suspending MV ⇨ making it inactive
- Refreshing MV
  - manually ⇨ the REFRESH option

```
ALTER VIEW MV-name MATERIALIZE {REFRESH | SUSPEND}
```

  - automatically ⇨ setting a refresh threshold
    - the threshold specifies the percentage of unsorted data in the materialized view, value from 1 to 99 (default 20)
    - the same threshold is set for all MVs in the system
    - admin privilege required to run the below command

```
SET SYSTEM DEFAULT MATERIALIZE THRESHOLD TO <number>
```

- The system creates zone maps for all columns in a MV that have data types integer, date, or timestamp

Netezza: ZM

- ZM - Zone Maps
  - similar to SMA
  - created automatically for every extent
  - by default created for columns of type
    - integer (byteint, smallint, integer, bigint)
    - date
    - timestamp
  - created automatically for columns used in the ORDER BY clause of a materialized view
  - for a given attribute ZMs store MIN and MAX value of the attribute in an extent
  - maintained automatically by the system

Netezza: ZM

extent1

Index	Date	Open	Close
BZWBK	21-03-2006	148,50	147,00
BZWBK	20-03-2006	148,50	147,00
BZWBK	17-03-2006	151,50	148,00
BZWBK	16-03-2006	150,00	149,50

extent2

Index	Date	Open	Close
BZWBK	15-03-2006	151,50	149,00
BZWBK	14-03-2006	149,00	148,50
BZWBK	13-03-2006	152,50	146,00
BZWBK	10-03-2006	152,00	146,00
BZWBK	9-03-2006	154,50	154,00

extent3

Index	Date	Open	Close
BZWBK	6-03-2006	170,00	168,00
BZWBK	5-03-2006	169,50	163,50
BZWBK	2-03-2006	170,50	168,00
BZWBK	1-03-2006	166,00	165,00

ZM for extent 1

Index		Date		Open		Close	
Min	Max	Min	Max	Min	Max	Min	Max
BZWBK	BZWBK	16-03-2006	21-03-2006	148,50	151,50	147,00	149,50

ZM for extent 2

Index		Date		Open		Close	
Min	Max	Min	Max	Min	Max	Min	Max
BZWBK	BZWBK	9-03-2006	15-03-2006	149,00	154,50	146,00	154,00

ZM for extent 3

Index		Date		Open		Close	
Min	Max	Min	Max	Min	Max	Min	Max
BZWBK	BZWBK	1-03-2006	6-03-2006	166,00	170,50	163,50	168,00

Netezza: CBT

- Clustered Base Table (CBT) ⇨ data are organized / clustered by 1 to 4 attributes (clustering / organizing keys)
- A way to rearrange data within all the extents in the same data slice, by values of selected attributes
- Organizing keys are used to group records within the table (store them in one or more nearby extents)
- Netezza creates zone maps for the organizing keys

```
CREATE TABLE tab-name
(...)
[ORGANIZE ON (org-key1, ..., org-key4)]
```

Netezza: CBT

```
create table orders_cbt
(
o_orderkey integer not null,
o_custkey integer not null,
o_totalprice decimal(15,2) not null,
o_orderdate date not null,
...
)
distribute on (o_orderkey)
organize on (o_orderdate, o_totalprice);
```

- Cluster keys can be changed

```
ALTER TABLE tab-name
(...)
[ORGANIZE ON (NONE | org-key1, ..., org-key4)]
```

- Materialized views cannot be build on CBTs

Netezza: CBT

- Important
  - creating or altering a table to become a CBT doesn't change its physical storage until the groom command is used
- CBT is useful for reducing access time of:
  - ORDER BY leading clustering attributes
  - WHERE on leading clustering attributes
  - GROUP BY leading clustering attributes

Netezza: CBT

- ZMs created for organizing keys of type:
  - integer
  - date
  - timestamp
  - char
  - varchar
  - nchar
  - nvarchar
  - numeric
  - float
  - double
  - bool

Netezza: CBT

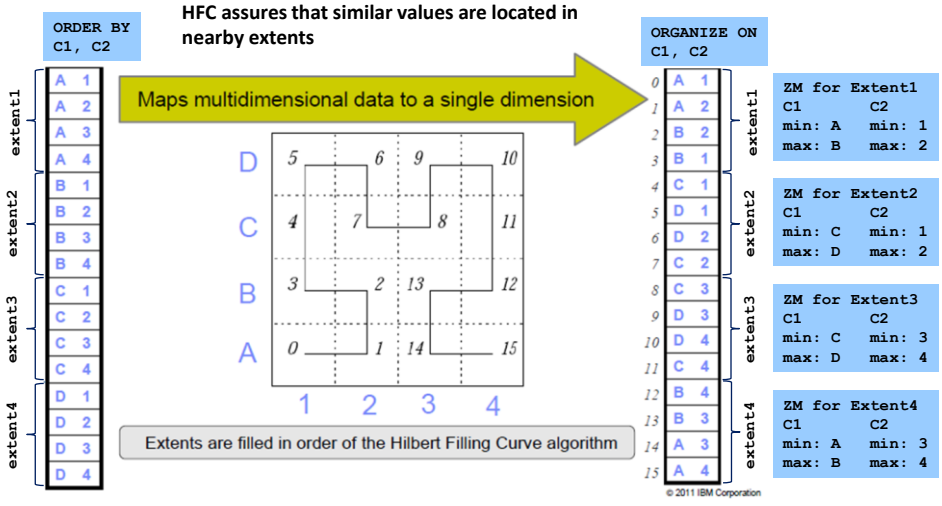


Figure adapted from IBM teaching materials

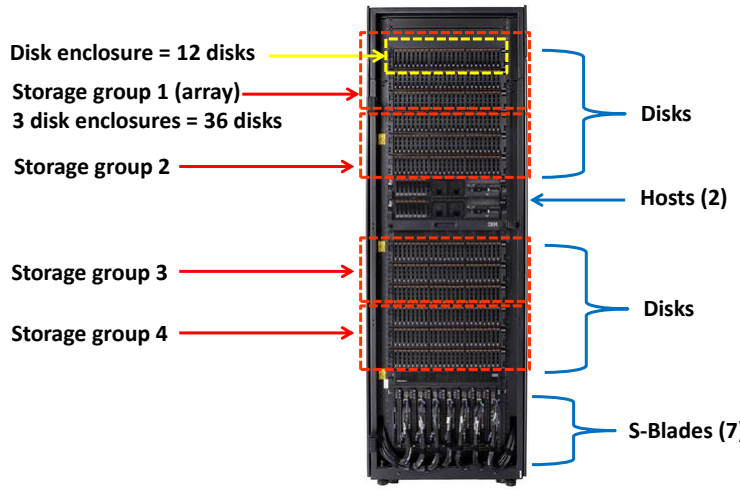
IBM PureData System for Analytics N1001

1-rack

Model	N1001-002	N1001-005	N1001-010
S-Blades	4	7	14
Snippet Processors	24	48	96
Disks [Spares]	24 [2]	48 [2]	96 [4]
User Data Storage (raw)	8TB	16TB	32TB

N1001: storage array contains 2 or 4 disk enclosures, each disk enclosure has 12 disks

IBM PureData System for Analytics N2001



IBM PureData System for Analytics N2001

Model	N2002-002	N2001-005	N2001-010	N2001-020
S-Blades	2	4	7	14
Snippet processors	32	64	112	224
Disks [Spares]	48 [4]	144 [16]	288 [34]	576 [68]
User Data Storage (Raw)	8 TB	24 TB	48 TB	96 TB

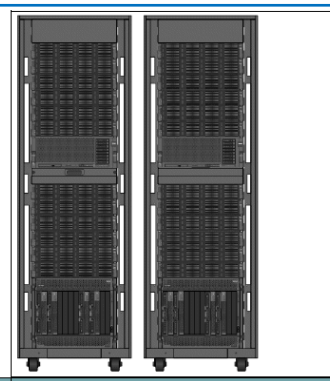
IBM PureData System for Analytics N3001

Model	N3001-002	N3001-005	N3001-010	N3001-020
S-Blades	2	4	7	14
Snippet processors	32	64	112	224
Disks [Spares]	48 [4]	144 [16]	288 [34]	576 [68]
User Data Storage (Raw)	8 TB	24 TB	48 TB	96 TB

N2001 and N3001

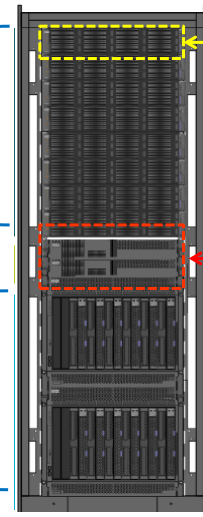
- In the N2001 and N3001 model families, each disk enclosure has 24 disks
- There are 12 disk enclosures in each full rack, or 6 enclosures in a half-rack model
- Each rack is one storage array

Netezza C1000 Systems



Model	C1000-8	C1000-16
S-Blades	8	16
Snippet Processors	64	128
Disks [Spares]	288 [16]	576 [32]
User Data Storage (Raw)	288 TB	576 TB
	2-racks	4-racks

Netezza TwinFin™ 12



Disk enclosure

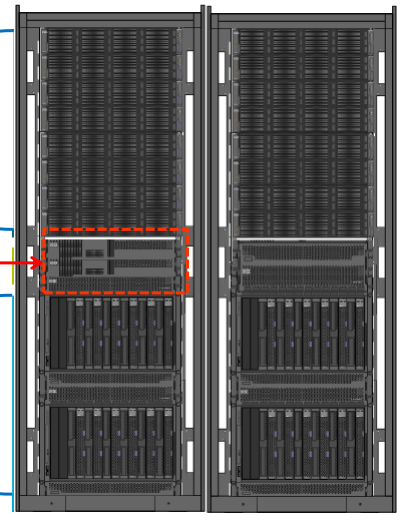
8 disk enclosures
12 disks/enclosure
disk capacity: 1TB
8[de] * 12[d] * 1TB = 96TB

Hosts (2)
one active, one passive
CPU: 2 Intel Quad-Core 2.6GHz
7x146GB SAS Drives
24 GB RAM
Red Hut Linux 5 64-bit

12 S-Blades
1 blade includes:
CPU: 2 Intel Quad -Core 2GHz
4 125MHz FPGA
16GB DDR2 RAM
Linux Kernel 64-bit

Data load speed: 1TB/h

Netezza TwinFin™ 24



2 * 8 disk enclosures
12 disks/enclosure
disk capacity: 1TB
2 * 8[de] * 12[d] * 1TB = 192TB

2 hosts

2 * 12 S-Blades

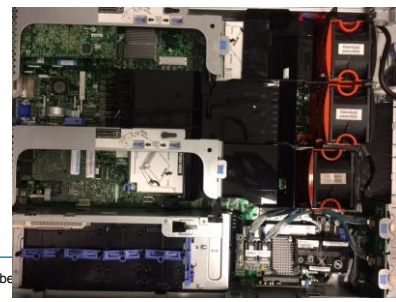
Data load speed: 2TB/h

Netezza @PUT

- appliance: IBM PureData System for Analytics
- model: N1001-005
- active S-blades: 7
- storage
  - 4 disk enclosures
  - 1 disk enclosure: 12 disks x 1TB
  - 2 spare disks for the whole appliance
- Total storage: 46TB (+2TB spare)
- user data (with compression): 64TB
- power consumption: 4kW

Netezza @PUT

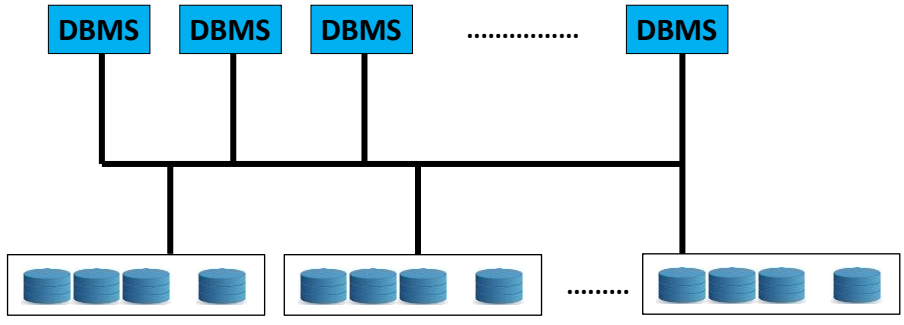
- Host
  - CPU: 2x Intel Xeon Quad-Core E5620, 2.4GHz
  - RAM: 24GB
- S-blade
  - CPU: 2x Intel Xeon Quad-Core E5620 2.4GHz (8 cores)
  - FPGA: 8 cores (4x double-engine modules)
  - RAM: 24GB (12x 2GB)



Oracle Exadata - Architecture

database servers

max 8 DB servers



storage servers
(Exadata cells)

max 14 storage servers

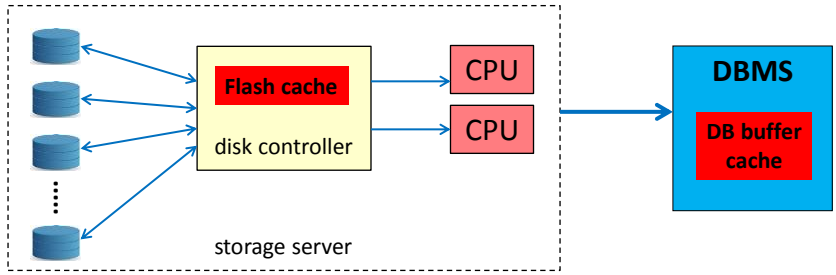
➲ **Shared disk architecture**

Oracle Exadata - Features (1)

- ➲ **Suitable for OLTP and OLAP**
- ➲ **Storage server**
  - **2 CPU Intel Xeon**
  - **Smart Scan module ⇨ similar to Netezza's S-Blade**
    - **parallel reads from disks**
    - **uncompressing**
    - **filtering**
  - **flash memory ⇨ used as cache for query intensive data**
    - **each storage server includes 4PCI flash cards of total capacity 3.2TB**
    - **max flash capacity 14*3.2 = 44.8TB (X4-2 series)**
  - **data compression**
  - **data distribution to all disks**

Oracle Exadata - Features (2)

- **DB server**
  - **runs under Oracle Linux or SUN Solaris**
  - **processes prefiltered data by Smart Scan modules**
- **InfiniBand switches connect DB servers and storage servers**
  - **40GB/s**
- **Max data load rate 20TB/h (full rack X4-2 series)**

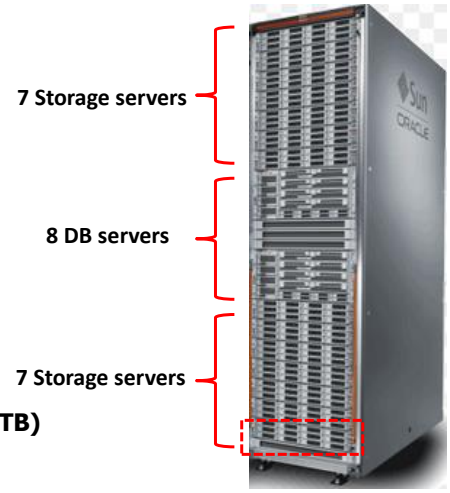


Oracle Exadata - Models (1)

Exadata Database Machine X3-2 Full Rack	Exadata Database Machine X3-8 Hardware
8 x Database Servers	**2 x Database Servers**
Each with: – 2 x Eight-Core Intel® Xeon® E5-2690 Processors – 256GB Memory – Disk Controller HBA with 512MB Battery Backed Write Cache – 4 x 300GB 10,000 RPM SAS Disks	Each with: – 8 x Ten-Core Intel® Xeon® E7-8870 Processors (2.40 GHz) – 2 TB Memory – Disk Controller HBA with 512MB Battery Backed Write Cache – 8 x 300GB 10,000 RPM SAS Disks
14 x Exadata Storage Servers X3-2	**14 x Exadata Storage Servers X3-2**
With: – 12 x 600GB 15,000 RPM High Performance SAS disks or 12 x 3TB Performance SAS disks or 12 x 2TB 7,200 RPM High Capacity SAS disks Includes: – 168 CPU cores for SQL processing – 22.4TB Exadata Smart Flash Cache	With: – 12 x 600GB 15,000 RPM High Performance SAS disks or 12 x 3TB 7,200 RPM High Capacity SAS disks Includes: – 168 CPU cores for SQL processing – 22.4TB Exadata Smart Flash Cache

Oracle Exadata - Models (2)

- Quarter rack (X4-2)
  - 2 DB servers
  - 3 storage servers
- Half rack (X4-2)
  - 4 DB servers
  - 7 storage servers
- Full rack (X4-2)
  - 8 DB servers
  - 14 storage servers
  - Disk types (X4-2)
    - high performance (1.2TB)
    - high capacity (4TB)



Other

- Teradata
  - shared nothing architecture
  - 5 Models
    - Data Mart Edition ⇨ up to 6TB
    - Data Mart Appliance ⇨ up to 8TB
    - Extreme Data Appliance ⇨ up to 234PB
    - Data Warehouse Appliance ⇨ up to 21PB
    - Active Enterprise Data Warehouse ⇨ up to 61PB
  - SUSE Linux
  - data compression
- SAP Hana
  - row store + column store
  - compression
  - partitioning

Literature

- The slides about IBM Netezza were prepared based on the official IBM materials:
  - "IBM Pure Data Systems for Analytics" - workshop
  - Netezza technical documentation
    - IBM Netezza Database User's Guide. IBM Netezza 7.2.x
    - IBM Netezza System Administrator's Guide. IBM Netezza 7.2.x
    - IBM Netezza Getting Started Tips. IBM Netezza 7.2.x
- The slides about Oracle Exadata were prepared based on:
  - Oracle Exadata Database Machine X4-2 (Oracle data sheet)
  - The Teradata Data Warehouse Appliance. Technical Note on Teradata Data Warehouse Appliance vs. Oracle Exadata