Microsoft

Database Design

Agenda

- DDL
- Creating a Database
- Understanding AUTOGROW
- Database Sizing
- Tempdb
- Transferring Ownership

Database DDL

Sample DDL

```sql
CREATE DATABASE PDW_2012
WITH
( AUTOGROW = ON
, DISTRIBUTED_SIZE = 1024.0GB
, REPLICATED_SIZE = 20GB
, LOG_SIZE = 10GB
);
DROP DATABASE PDW_2012;
```

DDL Points to Note

- File sizes specified as a decimal
- File size unit of measure is GB only
- Collation changes at database level not allowed
- Appliance default collation: Latin1_General_100_CI_AS_KS_WS
- AUTOGROW setting & allocation units cannot be seen via metadata

DDL Points to Note

Handled by PDW:

- Filegroup specifications
- File placement

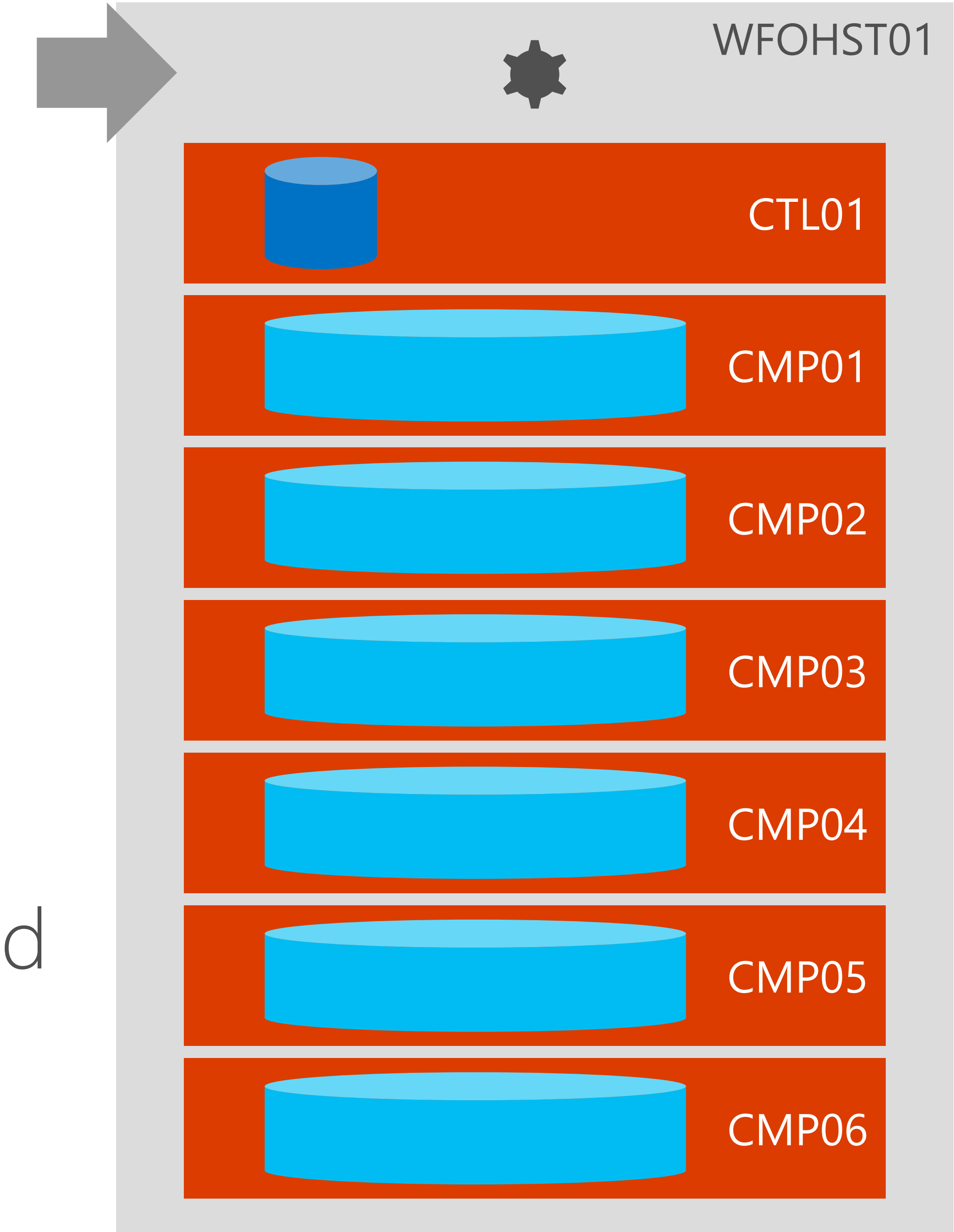
Creating a Database

Creating a Database

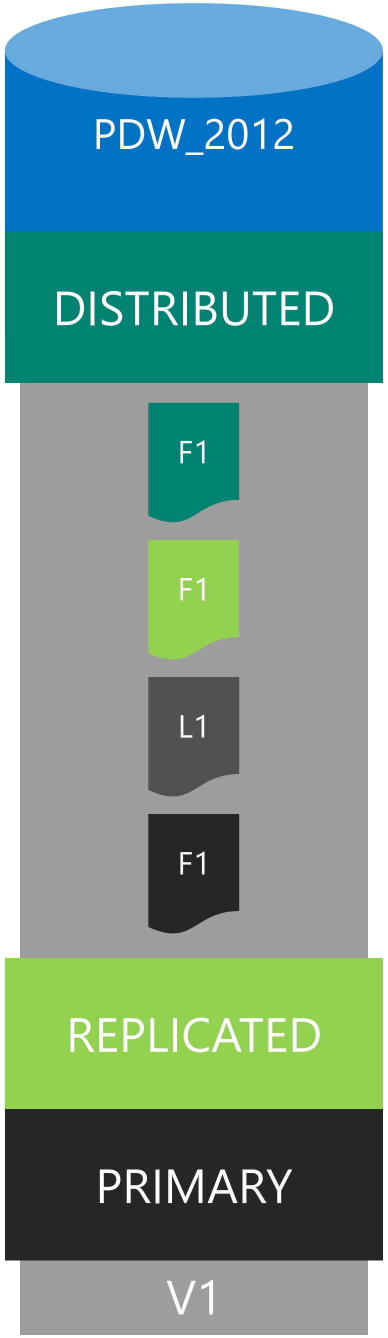
User Creates Database

- PDW creates a logical shell database on the Control node
- PDW creates a physical application database on each Compute node

Total number of databases created = # of Compute nodes + 1



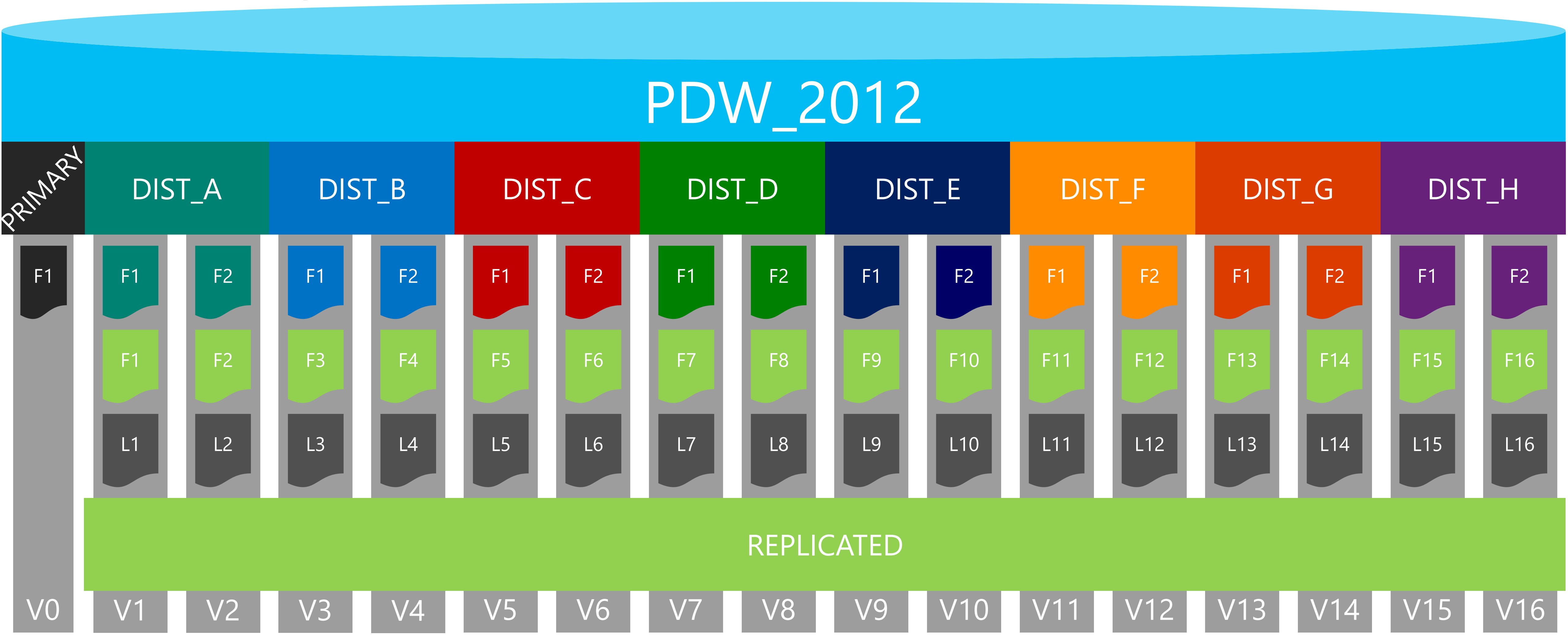
Control Node Database



Database Extended Properties

Name	Value
pdw_distributed_size	Value specified in DDL
pdw_is_autogrow	True or False
pdw_is_valid_full_backup	True or False
pdw_log_size	Value specified in DDL
pdw_physical_name	DB_<GUID>
pdw_replicated_size	Value specified in DDL

Compute Node Database



PDW_2012

PRIMARY	DIST_A		DIST_B		DIST_C		DIST_D		DIST_E		DIST_F		DIST_G		DIST_H	
F1	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16
	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13	L14	L15	L16
	REPLICATED															
V0	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16

Physical Database Configuration

- Simple Recovery Model
- MAXDOP 4

Database Autogrow

AUTOGROW

- Is not discoverable from PDW catalog views
- sys.database_files shows configuration of the shell database on the Control node
- There is no sys.pdw_nodes_database_files catalog view
- Primary filegroup handled differently to other filegroups: always ON for primary

Compute Node AUTOGROW - ON

FILE	TYPE	VALUE	COMMENT
ROWS	max_size	-1	Each data file is not limited by size
ROWS	growth	512	Each data file grows by a fixed amount of 512 8KB pages (4MB)
ROWS	is_percent_growth	0	Table grows in fixed amounts not by percent
LOG	max_size	268435456	Each log file will grow to a max of 2TB
LOG	growth	10	Each log file will grow by 10%
LOG	is_percent_growth	1	Log file grows by percentage increments

Compute Node AUTOGROW - OFF

FILE	TYPE	VALUE	COMMENT
ROWS	max_size	-1	Not relevant as growth = 0
ROWS	growth	0	Data files do not grow
ROWS	is_percent_growth	0	Not relevant as growth = 0
LOG	max_size	268435456	Not relevant as growth = 0
LOG	growth	0	Log files do not grow
LOG	is_percent_growth	0	Not relevant as growth = 0

Sizing

Allocating Space – REPLICATED_SIZE

Base on a single copy of all replicated tables

- Each Compute node

Allocating Space – REPLICATED_SIZE

Allocation Amounts

Total storage

- Value of REPLICATED_SIZE* # of Compute nodes

Per replicated data file:

- Value of REPLICATED_SIZE/ 16

Allocating Space – DISTRIBUTED_SIZE

Divided evenly across the appliance

- Across Compute nodes
- Across distributions
- Across data files allocated to the distribution filegroups

Allocating Space – DISTRIBUTED_SIZE

Allocation Amounts

Per Compute node:

- Value of DISTRIBUTED_SIZE / # of Compute nodes

Per distribution:

- (Value of DISTRIBUTED_SIZE / # Compute nodes)/ 8

Per distributed data file:

- (Value of DISTRIBUTED_SIZE / # Compute nodes)/ 8/ 2

Allocating Size – LOG_SIZE

Divided evenly across the appliance

- Across Compute nodes
- Across data volumes

Allocating Size – LOG_SIZE

Amount allocated

Per Compute node:

- Value of LOG_SIZE/ # of Compute nodes

Per data volume:

- (Value of LOG_SIZE/ # of Compute nodes)/ 16

Re-sizing User Databases

ALTER DATABASE

- Sets one property at a time
- Can be smaller than original value
- Also used for switching AUTOGROW ON | OFF

DBCC SHRINKLOG(SIZE <target> | DEFAULT)

- Target combined size on all nodes

General Sizing Advice

Pre-size your Database!

- Enable Instant File Initialization
- Allocate sufficient capacity for 6-12 months
- Keep sufficient spare capacity to rebuild your largest table
- Create log twice the size of your largest transaction
- AUTOGROW ON – avoids 3a.m. alarm call

Example

```
CREATE DATABASE PDW_2012
WITH
( AUTOGROW = ON
, DISTRIBUTED_SIZE = 1024.0GB
, REPLICATED_SIZE = 20GB
, LOG_SIZE = 10GB
);
```

Q. How much storage has been allocated for distributed tables? replicated tables? the transaction log?

A. 1TB

A. 120GB

A. 10GB

Per	Capacity (6 Nodes)
Compute Node	1.67GB
Log File	0.104GB

Guidance

Insufficient Capacity Consequences

Restricts ability to rebuild tables / indexes

Prevents automatic data re-distribution

- Extend appliance
- Restoring a database

Files using AUTOGROW get heavily fragmented

- Impacts performance

Remember Skew!

- File layout is important for understanding skew
- When data files for a distribution are full then the database is full
- When volume is full appliance is full

Data Loading & Sizing

- When loading data internal tables are created
- Internal tables hold a complete copy of the data being loaded
- You can nominate a specific database for these internal tables to isolate them from user database
- Failure to nominate a database will result in the internal tables being created inside target database
- Data load pattern can impact your database sizing!

General Loading Recommendations

- Create (and use) a staging database
- Size database to twice the size of largest load
- Size log to twice the size of largest load or batch
- If parallel loading load size includes all loads
- Staging database size likely to be different for initial/historic load and incremental load

FASTAPPEND load mode does not use internal tables.
Remember this for Data Loading Patterns segment!

Tempdb

Temporary Databases

Two Temporary Databases

- tempdb
- pdwtempdb

pdwtempdb

- User database
- Centralized resource for the appliance
- Holds user-defined temporary tables
- Isolates temporary tables into separate db
- Helps with sizing and fragmentation
- Does not have same optimisations as tempdb

pdwtempdb Database Sizing

Database Pre-sized at

- 1280MB per distributed data file
- 128MB per replicated file
- 64MB per log file
- Autogrows at 10% (unlimited)
- Re-created when appliance is re-started

Cannot be shrunk by DBCC SHRINKLOG or ALTER DATABASE

Contents visible in SSDT (tempdb)

pdwtempdb on 6 Compute Node PDW

6 Compute Nodes = 96 Data Volumes

pdwtempdb CREATE DATABASE size

- Distributed_Size = 122880MB or 120GB
- Replicated_Size = 12288MB or 12GB
- Log_Size = 6144MB or 6GB

tempdb

- System database created by SQL Server
- Exists on Compute and Control nodes
- Spread across all data volumes
- Used for data movement
- Tables created in tempdb are called Q tables
- Also used by normal SQL Server operations such as sorts and spills

tempdb on Control Node

One data file created

One log file created

Database Pre-sized at

- Data file 1000MB
  - autogrows at 1000MB (unlimited)
- Log file 100MB
  - autogrows at 100MB (unlimited)
- Re-created when appliance is re-started

Cannot be shrunk by DBCC SHRINKLOG or ALTER DATABASE

Contents visible in SSDT (tempdb)

tempdb on Compute Nodes

One file created per data volume (16 total)

Database Pre-sized at

- 1000MB per data file
  - Data file AUTOGROWs at 1000MB (unlimited)
- 100MB per log file
  - Log file AUTOGROWs at 100MB (limited 2097152MB)
- Re-created when appliance is re-started

tempdb on 6 Compute Node PDW

Control Node

- Data file = 1000MB
- Log file = 100MB

Compute Nodes (96 Volumes)

- Data files = 96000MB or 93.75GB
- Log files = 9600MB or 9.375GB

SSDT and tempdb

- tempdb shown in SSDT is a logical view
- Representation for the appliance
- Q Tables – tempdb
- #tables - pdwtempdb

Ownership

Database Ownership

- Login used to create the database is the owner of that database
- If login needs to be dropped then ownership must first be transferred

The sa account cannot be dropped or disabled in PDW. It is therefore a good option for database ownership

Transferring Database Ownership

- Use the ALTER AUTHORIZATION command

```
ALTER AUTHORIZATION ON DATABASE::
PDW_2012 TO sa;
```

Microsoft

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.