Data Protection

4 ways of protecting data

- Transient Journal.
- Fallback: AMP failure protection.
- RAID 1: Disk failure protection.
- Clique: Node failure protection.

Transient Journal

- Changes of a transaction are recorded here.
- A copy of a row that is being modified is stored temporarily to the transient journal to ensure data integrity.
- This allows to do roll backs in the case of errors.
- Rows are deleted as soon as the transaction is finished.
- The Parsing Engine takes care that all AMPs finish their work before a transaction is committed.

Fallback

- A copy of each row is stored in a different AMP.
- The main AMP and the fallback AMP protect each other.
- They are never stored on the same hardware, to prevent from hardware failure.

RAID 1: Mirroring

- Protection against disk failure.
- Each AMP has exactly one virtual disk assigned, but this might be composed of several physical disks.
- RAID 1 protection creates a copy of these disks.

Clique

- Nodes are organized in groups called *cliques*.
- In the event of node failure, the AMPs of the failed node move to another node in the same clique.
- To avoid overloading the non-failing node, Teradata offers hot standby nodes for taking over the AMPs of the failed node.

Tools and Utilities

One tool to rule them all

- Historically, many tools for handling data import/export:
  - `FastLoad`
  - `MultiLoad`
  - `Tpump`
  - `BTEQ`
- Merged together in a single tool: Teradata Parallel Transporter `TPT`.

Bulk vs transactional load

- `FastLoad` and `MultiLoad` are the most efficient load utilities, as the data is loaded in blocks of 64 kilobytes of data. They skip the transient journal.
- `BTEQ` and `TPump` insert each row separately, using transaction handling (offering rollback functionality).
- Bulk load utilities (`FastLoad` and `MultiLoad`) do not support indexes, triggers or referential integrity checks.

`FastLoad`

- **Main use: to load empty tables at high speed.**
- The target tables must be empty in order to use `FastLoad`.
- Supports inserts only it is not possible to perform updates or deletes.
- Although `FastLoad` uses multiple sessions to load the data, only one target table can be processed at a time.
- Duplicate rows will not be loaded.

`MultiLoad`

- **Main use: Load, update and delete large tables in Teradata in a bulk mode**.
- Multiple tables can be loaded at a time.
- Update and delete also supported.
- Duplicate rows allowed.
- Target table does not have to be empty.

`BTEQ` and `Tpump`

- `BTEQ` was the first tool introduced for loading data, and still quite useful for loading small amounts (a few thousand rows).
- `Tpump` adds support for multiple load sessions and allows to load the data at a given rate (rows per second).

TPT

- The base of TPT for Teradata are the concepts of ***data streams** and **operators**.
- Data streams are not directly accessible from your scripts. They are the pipelines between operators and they are kept in memory. No data is written to the disks.
- Operators read data from a source or write data to a target. Some operators take over more tasks such as dropping and creating of tables.

Advantages of `TPT`

- **Performance:** TPT allows the processes running on the client load server to be scaled and parallel data streams can be created to circumvent performance bottlenecks.
- For example, if I/O is a bottleneck when reading a very large input data file, then one can scale Parallel Transporter to create multiple data flows with multiple readers of the same file or multiple files to create more data throughput for the load.

Advantages of `TPT` (cont.)

- **Ease of use:** There are many features when using the script interface that makes writing a load job much easier.
    - *Example:* One script can extract data from a production Teradata Database and load into a test database. The data will flow in memory among the parallel processes on the client load server. With the stand-alone tools one would have to write two scripts in two different languages and put a named pipe in between the two tools to pass data.
- **Downside:** Need to learn the scripting language, which is a bit weird.

More on `TPT`

- https://www.dwhpro.com/tpt-teradata-parallel-transporter/
- Example scripts: https://www.dwhpro.com/teradata-tpt-export-import-script-examples/