Stanford Linear Accelerator Center

Oracle Storage for the Channel Archiver

Managing Channel Archiver data with Oracle partitions Overview

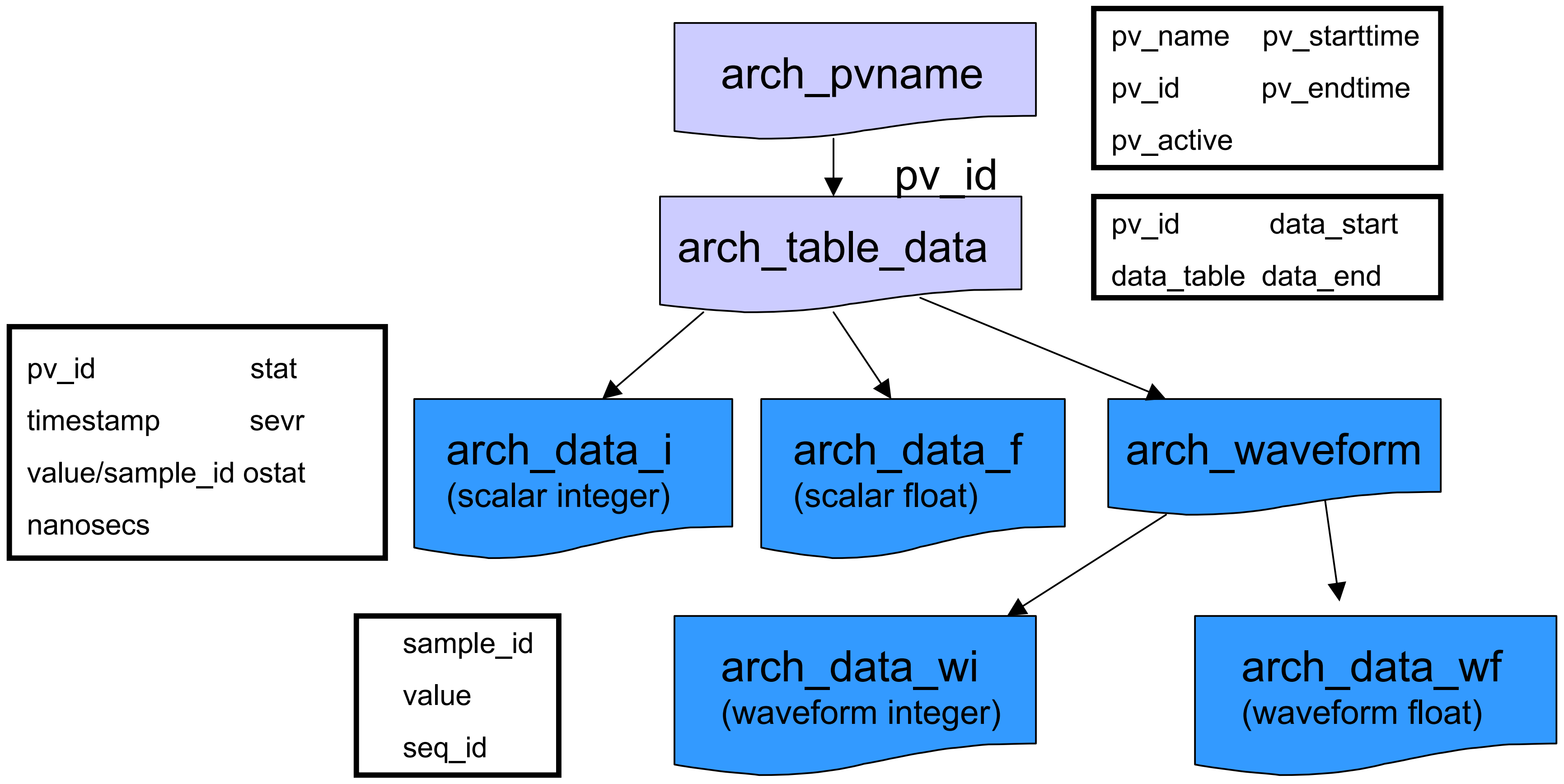
Topics

- Goals
- Oracle table design for Channel Archiver data
- Partition management algorithms
- Partition compaction algorithms
- Oracle support tables
- Oracle views to retrieve data
- Status of implementation

Goals

- Increase data storage and retrieval performance
- Use a commercial RDB with its associated data management tools
- Support the current functionality of the Channel Archiver
- Allow flexibility for each site to manage their data their own way.

Table Structure Overview



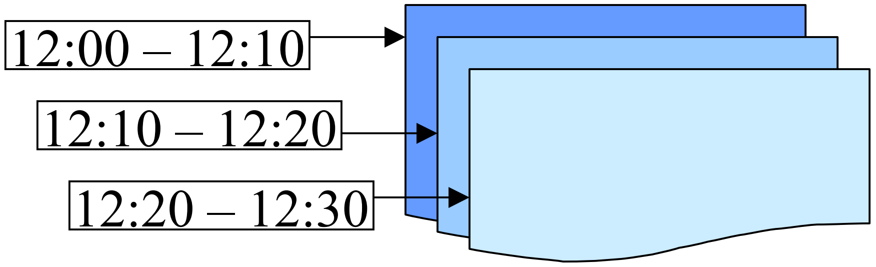
Partitioning Syntax

```
CREATE TABLE ARCH_DATA_I

(     timestamp              date,
      pv_id                  number(38),
      value                  number(38),
      nanosecs               number(9),
      stat                   number(8),
      sevr                   number(8),
      ostat                  number(16) )

  PARTITION BY RANGE (timestamp)

      ( partition MAY0702_0001 values less than
           (TO_DATE('05/07/2002 00:10:00','mm/dd/yyyy hh24:mi:ss')),

        partition MAY0702_0002 values less than
           (TO_DATE('05/07/2002 00:20:00','mm/dd/yyyy hh24:mi:ss')),

        partition bin values less than (MAXVALUE) );
```

Partitioning of Oracle Tables

The arch_data_f, arch_data_i and arch_waveform tables will be partitioned into small (~10 minutes but can be specified) time intervals for the day. These tables are NOT indexed.

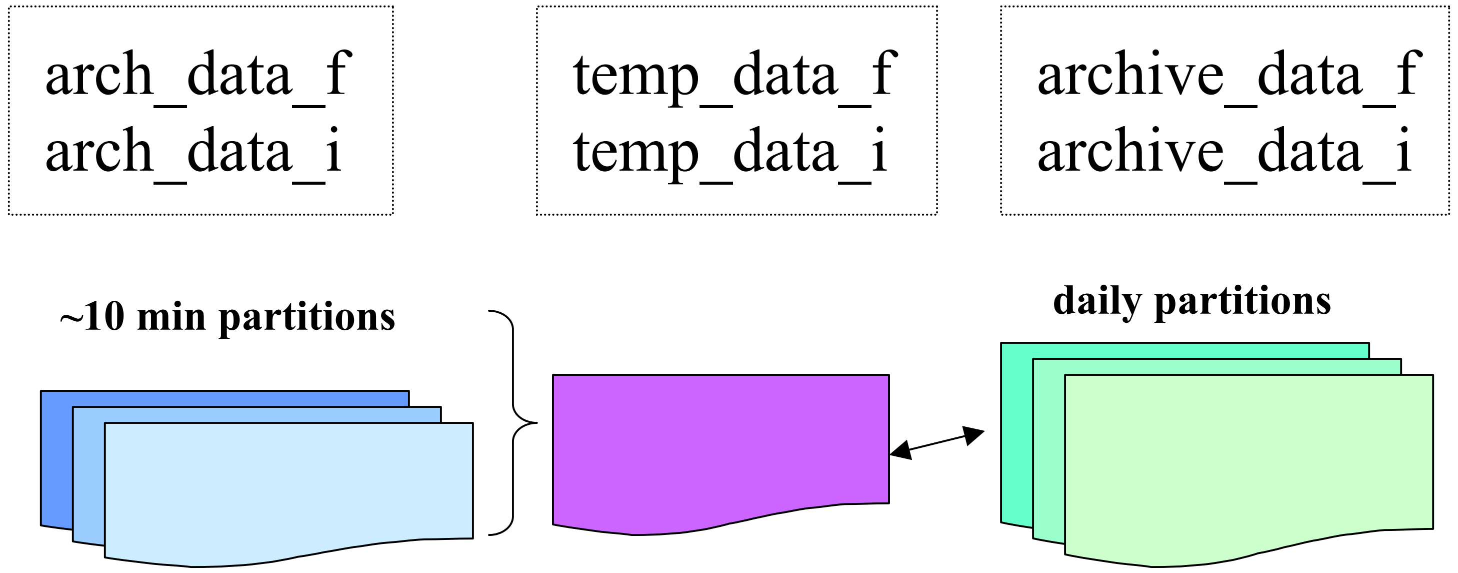
EX:



Oracle will track which partition to store the data in so no additional overhead is performed by the Archive Engine.

Daily Processing of Partitions

(Scalar Data – similar processing for waveforms)



All partitions for yesterday are copied to a temporary table

The data is validated and indexes are created

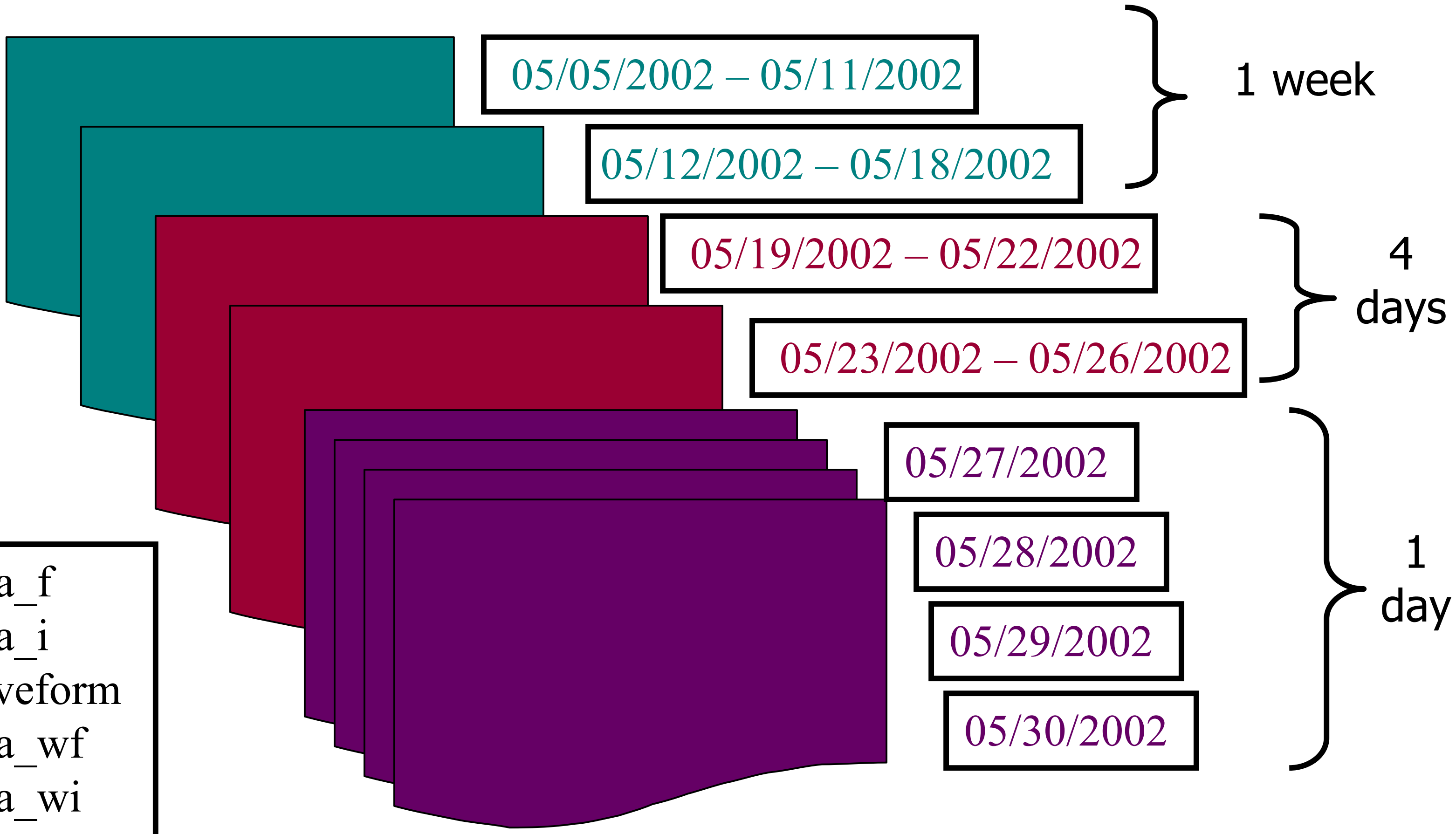
The temporary table, along with indexes, is exchanged with a partition in the archived data tables

Partition Compaction Algorithm

- Each night a partition compaction script will run which processes archive_XXX tables and associated indexes.
- The compaction algorithm uses the arch_part_durations table to determine the way in which partitions will be compacted.
- Eventually, it will also be used to handle the "rolling out" of partitions from the current location into a near-storage device.

Oracle Partition Compaction

EX:



archive_data_f
archive_data_i
archive_waveform
archive_data_wf
archive_data_wi

Oracle Views for Retrieval

- Since there are two tables for each data type, an Oracle view retrieves data from both tables for each data type
- The retrieval SQL queries the view instead of querying the tables directly.
- The views are created read-only
- Views allow flexibility as to which data the user has access
- Views allow access to scalar and waveform data to be the same

Status of Implementation

- We have an Oracle machine available (4x450 MHz processors) and sufficient disk space.
- The OCI interface to Oracle is working well
- The support scripts are written but some of the partition management functions are still being worked out
- We have a limited number of licenses for the Oracle partitioning option
- We are using the libIO code delivered by Thomas Birke; more work is needed

Testing Results

- Retrieval is still the choke point
- Smaller partitions reduce access time
- About one year’s data from several hundred PVs now in the database
- 4 days worth of one scalar is fetched in 2-3 seconds

Problems

- We know of no good browser
  - Python browser
  - CGI browser
- Some additional libIO support needed before we can stop recording old-style files
- Computer-center support for backup still an interesting question.

Additional Notes

- We have tried to keep most of the processing flexible so other labs can use it “out of the box”.
- Other labs may use bits and pieces of the Oracle table processing algorithms and are not required to handle their data the same way we plan on handling our data at SLAC. The only hard and fast requirement is for the *initial* table structure to be the same so the Channel Archiver knows where to store the data.
- We are open to any suggestions and ideas for improvement.