MeerKAT
64-dish Instrument

Situated in the sparsely populated, semi-desert, Karoo region of South Africa

- 64 dish interferometer
- Precursor to SKA
- 8 km longest baseline
- 4 bands (L, S, X, UHF)
MeerKAT each Dish

- 42 tons
- 19.5 m high
- Offset Gregorian
- 13.5 m diameter
- 3.8 m sub-reflector
- 40 aluminium panels
- Multiple receivers - bands
- Mechanical band selection
- Az: 2 deg/s (185° to 275°)
- El: 1 deg/s (15° to 92°)
- 40 GbE fibre network
- ~35 Gbps data
**CAM**
(Control and Monitoring Sub-system)

- CAM = Glue Links systems
- Abstraction layer
- Not real time
- Domain specific
- Multiple Subarray Obs Scripts (Python)
- Schedule Blocks
- Resource Management

Use KATCP
Client-server
TCP
Requests
Sensors

Monitoring at the Application layer
Agg Sensors
Alarms
Specialised UI
Katportalclient
Archive
Control protocol used for all internal communications

- Text based over TCP/IP
- Client-server
- Tooling (Telnet etc.)
- Used in all subsystems
- Requests
- Sensors
- Advanced sampling strategies
- Open sourced
Need a fast and efficient solution to store sensor samples:

- Not affect operations
- Not only numeric values
- Associated information
- No decimation
- Variable rate (0.1Hz - 10Hz)
- Dynamic sensor creation
- Large values (200KB)
- High volume (72K sample/s)
- Grow without slowdown
Query Objective

Troubleshooting, verify operations and decisions making

- Near realtime
- Complete lifetime
- Resampling for plotting
- Fast-ish queries
- Not affect archiving
- Many simultaneous
- Sensor attributes
- Queries in Karoo and Cape Town
Messaging System

- Simple, high performance open source messaging system
- Clustered and scalable out of the box
- CAM components publish to NATS, archive workers queue subscribe to subjects
- Only for monitoring data, KATCP for control
- Thursday’s poster session THPHA137
Sensor Sample as a Document

- Simplified implementation
- Future-proof implementation
- Store Information (not only data)
- Use JSON

```json
{
    "name": "m000_cryostat_pressure",
    "time": 1505982067.202219,
    "value": 1013.25,
    "status": "nominal",
    "value_ts": 1505977839.44
}
```
Two stage storage pipeline

- Message bus
- Buffer Table
- Archive Table
- Single DB server
- Reduced complexity
- Reduced cost
Buffer
samples_buffer

- Bus2Db
- TimescaleDB
- SSD
- time - Timestamp
- name - Text
- sample - JSONB
- archived - Boolean
Federated Table
FDW in Python3
Ceph Rados
Object per Day per Sensor
Capable of PB
Low DB overhead
Move sample from buffer to archive in the DB; using a stored procedure
Query
Structured Query Language (SQL)

- Complete SQL
- All samples accessible
- Advanced analytics
- Resampling
- Easy to use
- Powerful
PostgreSQL extension

sudo make install
sudo -u postgres psql katstore
CREATE EXTENSION katstore;

- Simpler deployment
- Versioning of procedures and schema
- Testing of SQL
- Depend on Katstore Python module
Conclusion

- Can handle 90k samples per second
- Tested with 200K sensors
- Mean query for 64 concurrent queries for 86400 samples ~ 8 seconds
- Ceph Rados is easy to work with and offers huge scalability
- PostgreSQL is a very powerful database
Questions?
Thank you
SKA South Africa, a Business Unit of the National Research Foundation.

We are building the Square Kilometre Array radio telescope (SKA), located in South Africa and eight other African countries, with part in Australia. The SKA will be the largest radio telescope ever built and will produce science that changes our understanding of the universe.

**Martin Slabber**  
Software engineer  
Email: [martin@ska.ac.za](mailto:martin@ska.ac.za)

**Francois Joubert**  
Software engineer  
Email: [fjoubert@ska.ac.za](mailto:fjoubert@ska.ac.za)

**Toufeeq Ockards**  
Software engineer  
Email: [tockards@ska.ac.za](mailto:tockards@ska.ac.za)