Gilead Sciences Usage of Database In-Memory

Ajay Poondla
Gilead Sciences
About Ajay Poondla

- 15+ years of experience working in Data & Analytics space.
- Worked in different roles which include building a roadmap for platforms, Enterprise Architecture, implementing solutions, streamlining processes for different clients.
- Currently manage DevOps team supporting Data & Analytics platforms at Gilead.
OBIA Architecture at Gilead Sciences

- Informatica Client
- DAC Client
- GCWPROD
- Database replication service (netapp snap)
- Source Systems
- Database Tier
  - OLTP
  - DRP
- Hyperion
- Datawarehouse for OBIEE System
- GCWPROD 1
- GCWPROD 2
- OBIEE Servers
- End user

@<twitter handle>
Our Business Challenge

• Report/Analytics requirements change frequently: new filters, brand new reports, new data sources, ad-hoc queries, 3rd party data etc.

• Slow queries are huge productivity drain through the org

• Pain flows downhill: End user->Biz analyst->Developer->DBA->Storage/Networking
  – 20-30% DBA time spent in this loop

• Cycle repeats especially around M/Q/Y close
Gilead Sciences
Supporting Real Time Analytics in Finance Organization

Our Use Case
• Sustain M/Q/Y end key critical objectives & SLAs
• **1 Hour SLAs** - Hourly data refresh against multiple subject areas
• 100 % Uptime to avoid interruption during M/Q/Y End financial close periods
• Stake holders run **adHoc query/report demands & regular dashboards**
• **Consistent performance** for end users across all the Geo’s

20+% time spent in service requests and incident resolution for analytics platform
Gilead Sciences
Supporting Real Time Analytics in Finance Organization

Solution: Database In-Memory
• 1TB of data, ~250-300GB of In-Memory usage
• Dell 48-core server w/ 500GB (no Exadata)
• Data fully loaded to in-memory
• Dropped all indexes, constraints
• 1000+ end users/analysts running 6000 reports on system

• Resolved high pain-point for financials system
• 50%-2000% gains (2x-6x typical)
• In-Memory reason for move to 12c
Our Workflow

- Drop indexes, constraints
- Load all data into memory
- Confirm data is loaded into memory
- Integrate with DAC post ETL job
- After every ETL we run the job which does the snap of data and loads the data into memory on the reporting database
- The end to end process takes about 15 minutes