

# Beyond Conventional Data Warehousing

Florian Waas  
Greenplum Inc.

# Takeaways

- The basics
  - Who is Greenplum? What is *Greenplum Database*?
- The problem
  - Data growth and other recent trends in DWH
  - A look at different customers and their requirements
- The solution
  - Teaching an old dog new tricks: using an RDBMS for massively parallel data processing
- Conclusion

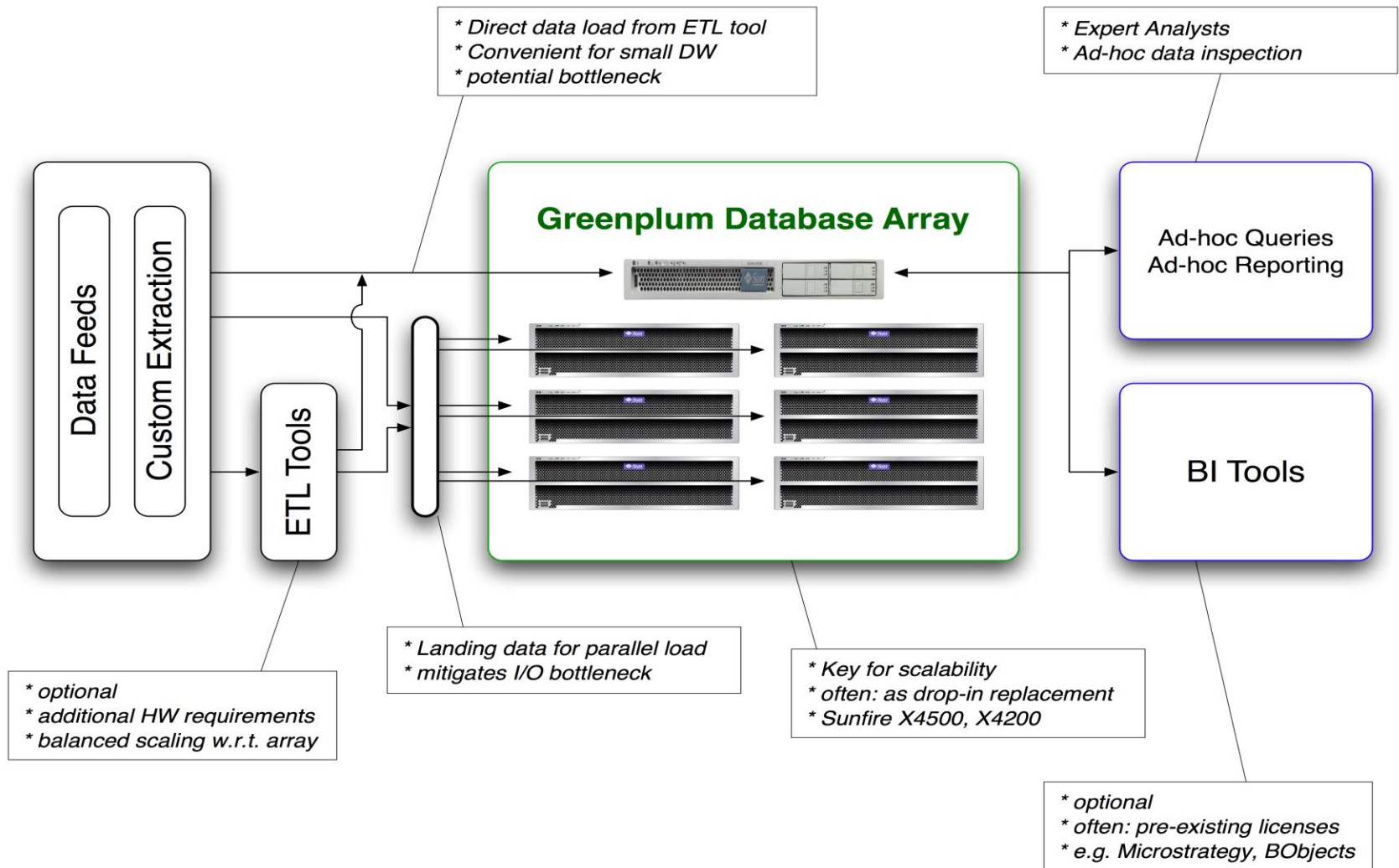
# Greenplum Inc.

- **What:** High-performance database software for Business Intelligence and Data Warehousing
- **Where:** Based in San Mateo, CA
- **When:** Founded in June 2003; product GA in February 2006
- **Who:** Technical pioneers in data warehousing (Teradata, Microsoft, Oracle, Informix, Tandem, PostgreSQL, ...)
- **Strategic Partner:** Powers the Sun Data Warehouse Appliance.

# Greenplum Database

- DBMS
  - highly scalable
  - fault-tolerant
  - high-performance
- Based on Postgres
- Shared-nothing architecture
- Commodity hardware
- Currently supported on Solaris, Linux

# Architecture



# Sample Hardware Configuration – Sun Fire X4500 Data Server



- 2 dual core AMD processors
- 48 Hitachi Deskstar SATA II 7200 rpm 500GB drives
- 6 Marvell 8-port Serial-ATA 2.0 Storage Controllers
- Leverages Hyper-transport architecture to achieve high-

# Trends in DWH: Data Growth

- Growth of customer base
  - E.g. phone carriers in Asia
- Additional data sources
  - E.g. click-stream and ad-impression data
- Data processing – “Data Bunnies”
  - E.g. intermediate results of analysis, aggregated/expanded
- Data will continue to outpace Moore’s Law

# Trends in DWH: Customers

- New Customers
  - Not your typical DB customers
  - No pre-existing DB infrastructure
  - Atypical data: logs, click-stream, etc.
- “Weight” of data less significant
  - E.g. CDR – call detail records
  - Click-stream vs. sales/transaction records
  - Often: Reflects behavior, not deliberate purchase decision
  - “bankers vs. teens”
- Analysis as service

# Trends in DWH: Analysis

- Turn-around on reporting
  - Similar/same requirements despite increased data volume
- Automated/on-line decision making process
  - E.g. ad placement in social network applications
- Advanced data analysis processes over massive amounts of data
  - E.g. Bayesian classification

# Requirements

- Petabyte-scale storage
- High-performance query processing
- Fault-tolerance/high-availability
- Constant loading activity
- “Richer processing capabilities”
  - Leverage parallelism automatically
  - Cannot move data (size, privacy concerns)
  - Integrate with existing programming environments
  - Not strictly a DWH requirement

# Leveraging Greenplum Database

- GPDB designed for
  - Scalability
  - High-performance query processing
  - Fault-tolerance
- How to address processing needs?

# How to use GPDB for Data Processing

- Typical installation 10s to 100s of CPU cores
- 100s GB memory
- 100s TB disk space
- Often largest individual system in data center
- Slack resources during off-peak times

# Example: ETL (1)

- **Customer's System**
  - 40 nodes
  - 160 CPU cores
  - 1 TB main memory
  - 3.6 TB/h load rate
- **ETL jobs**
  - 18 hours to process 1 day's worth of data
  - 5 serial streams
  - Load time < 1hr

## Example: ETL (2)

- ETL crucial in daily processing
- Mainly data cleaning: string manipulations, conversions, etc.
- Hard to parallelize effectively, load-balance
- Hard to recover if falling behind
  - E.g. glitches in ETL logic, data contamination
- Desired run time < 4hrs

## Example: ETL (3)

- Load “dirty” raw data directly into GPDB
  - Trade-off: raw data bulkier
- Rewrite ETL logic in SQL
  - Cleaner program
- Run SQL statements on GPDB
  - Automatic parallelization, fully transparent
  - Max degree of parallelism
  - Run time < 3 hrs

# Solution

- Leverage existing query processing infrastructure
- Rewrite procedural logic in SQL
- Enjoy benefits of SQL
  - Automatic parallelization
  - Add UDFs and UDAggs in other languages as needed, e.g. Java, C#, etc.

# Challenges

- Query Processing does not mean *read-only*
- Database Technology suffers image problem
- SQL is difficult
  - Declarative programming perceived as non-intuitive
  - SQL dialects (portability issues)
  - Too powerful – overwhelming
- Requires special skillset/expertise

# Summary

- Database technology for DWH addresses scalability, fault-tolerance and performance needs
- Users are looking for additional mileage from large-scale DWH installations
- ELT, and tools like UDFs, UDAggs become more attractive
- Existing database technology to be revamped into massively parallel processing engine

# Beyond Conventional Data Warehousing

Florian Waas  
Greenplum Inc.