TPC-DI
The First Industry Benchmark for Data Integration

Meikel Poess, Tilmann Rabl,

Hans-Arno Jacobsen, Brian Caufield

VLDB 2014, Hangzhou, China, September 4

Data Integration

- Data Integration (DI) covers a variety of scenarios
  - Data acquisition for business intelligence, analytics and data warehousing
  - Data migration between systems
  - Data conversion
  - etc.
- All of the above require:
  1. Data Extraction from Multiple Sources
  2. Data Transformation from Multiple Formats into a Target Format
  3. Consolidating data into one or more Target Systems

Why a Data Integration Benchmark

- Vendors
  - Publish performance numbers without always providing detailed information on how performance numbers were obtained
  - Compare performance numbers that might not be comparable
- Results are of little value for customers who would like to evaluate data integration tools across vendors
- Situation is similar to the 1980's when vendors compared performance of OLTP systems using a variety of workloads and metrics and which eventually resulted in the creation of the TPC.
- See "The History Of DebitCredit and the TPC" by Omri Serlin

JasperETL can be used for both analytic decision support system tasks such as updating data warehouses or marts, as well as for operational solutions such as data consolidation, duplication, synchronization, quality, migration, and change data capture. Performance tests indicate performance up to 50% faster than other leading commercial ETL tools.

PowerCenter 8 running across 64 CPUs loaded 1TB into Oracle in just 45 minutes, compared to 95 minutes for PowerCenter 7. Additional tests that targeted flat files took just 32 minutes, a new world record based on published benchmarks. As in past benchmarks, PowerCenter 8 exhibited near-perfect linear scalability across 16-, 32- and 64-CPU HP Integrity Superdome server configurations. Informatica

Microsoft and Unisys announced a record for loading data into a relational database using an Extract, Transform and Load (ETL) tool. Over 1 TB of TPC-H data was loaded in under 30 minutes. SSIS

Outline

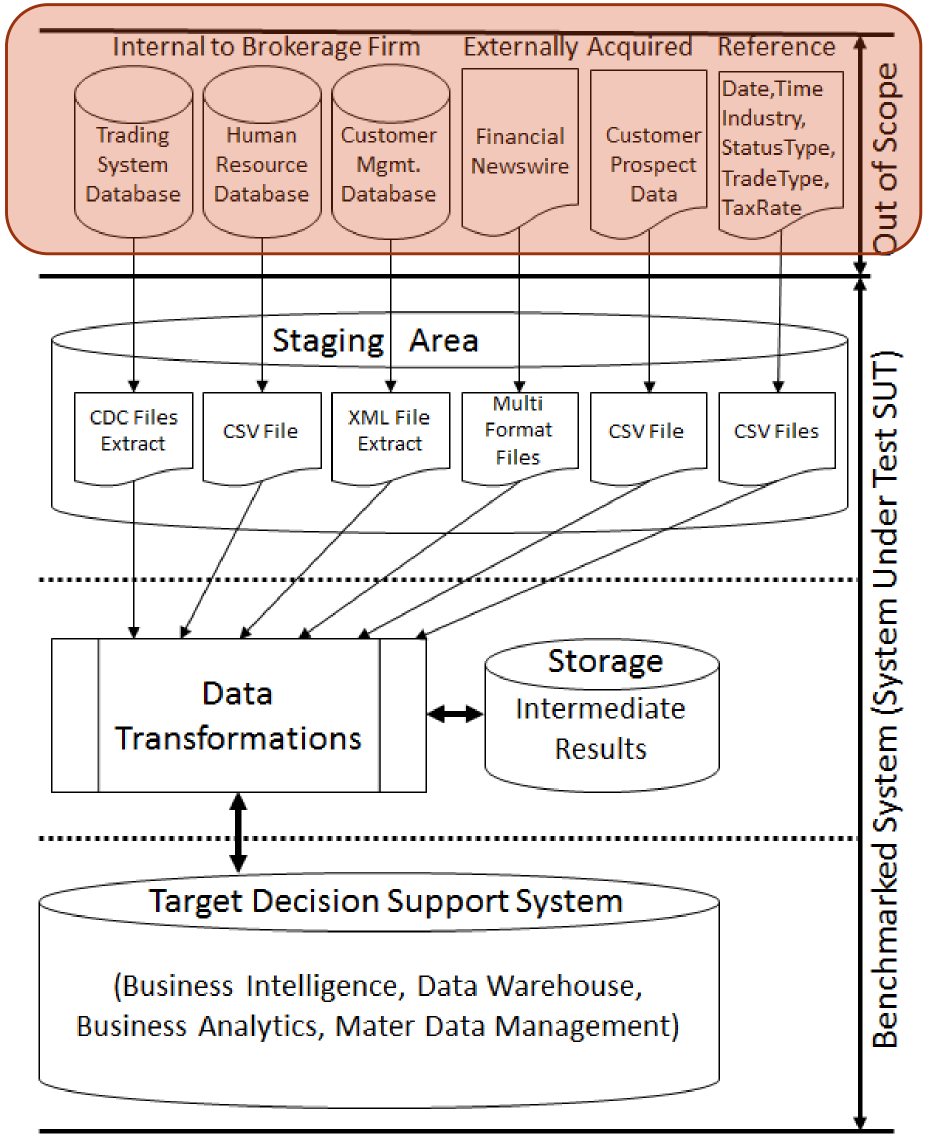
- Scope of TPC-DI
- General Concepts
- Data Model
  - Source Model
  - Target Model
  - Data Set
- Transformations
- Metric and Execution rules
- Experimental Results

General Concepts

- TPC-DI uses data integration of a factious Retail Brokerage Firm as model:
  - Main Trading System
  - Internal Human Resource System
  - Internal Customer Relationship Management System
  - Externally acquired data
- Operations measured use the above model, but are not limited to those of a brokerage firm
- They capture the variety and complexity of typical DI tasks:
  - Loading of large volumes of historical data
  - Loading of incremental updates
  - Execution of a variety of transformation types using various input types and various target types with inter-table relationships
  - Assuring consistency of loaded data
- Benchmark is technology agnostic

Scope of TPC-DI

- Out-Scope
  - Extraction of data from operational systems
  - Transport of data into a staging area
  - Data of source systems is provided by a data generator, based on PDGF
- In-Scope
  - Reading of data from staging area
  - Data transformation and their insertion into target system
  - Storing of intermediate results
  - Verification of transformed data



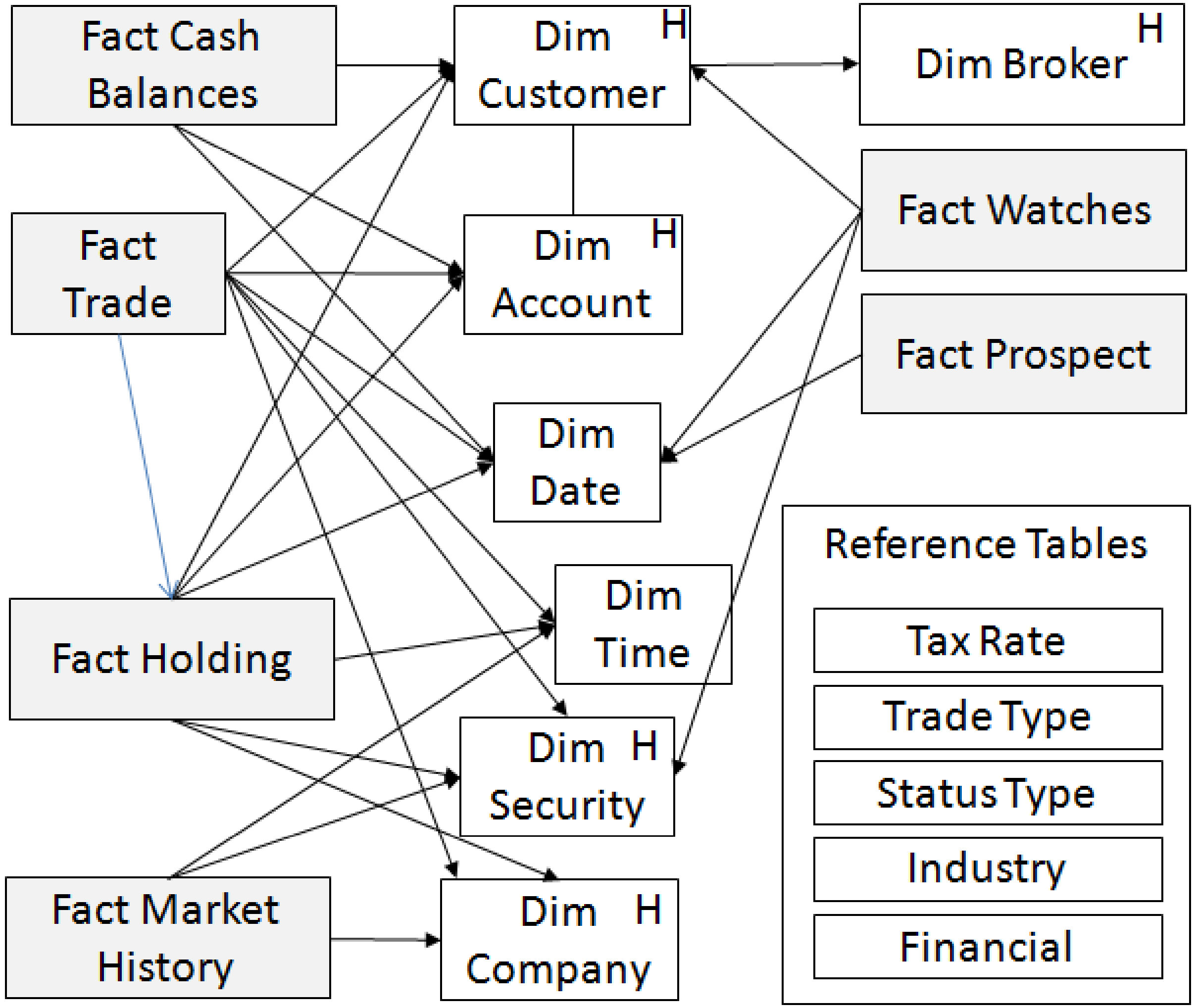
Source Schema

- 18 different source tables
- Various formats
  - CSV (Comma Separated)
  - CDC (Change Data Capture)
  - Multi Record
  - DEL (Pipe Delimited)
  - XML
- Some used only for the Historical Load
- Some used only for the Incremental Load
- Some used in both Historical and Incremental Loads

Source Table	Format	H	I
Account.txt	CDC		✓
CashTransaction.txt	DEL/CDC	✓	✓
Customer.txt	CDC		✓
CustomerMgmt.xml	XML	✓	
DailyMarket.txt	DEL	✓	✓
Date.txt	DEL	✓	
Time.txt	DEL	✓	
FINWIRE	Multi-record	✓	
HoldingHistory.txt	DEL	✓	✓
HR.csv	CSV	✓	
Industry.txt	DEL	✓	
Prospect.csv	CSV	✓	✓
StatusType.txt	DEL	✓	
TaxRate.xt	DEL	✓	
TradeHistory.txt	DEL	✓	
Trade.txt	DEL/CDC	✓	✓
TradeType.txt	DEL	✓	
WatchItem.txt	DEL/CDC	✓	✓

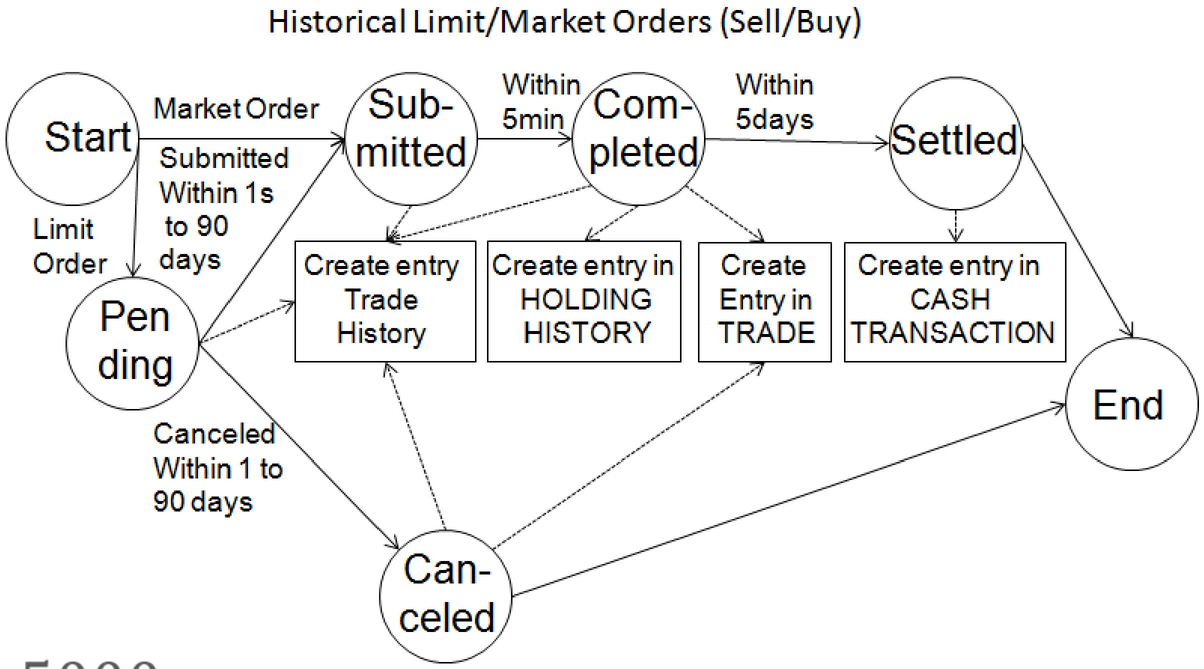
Target Schema

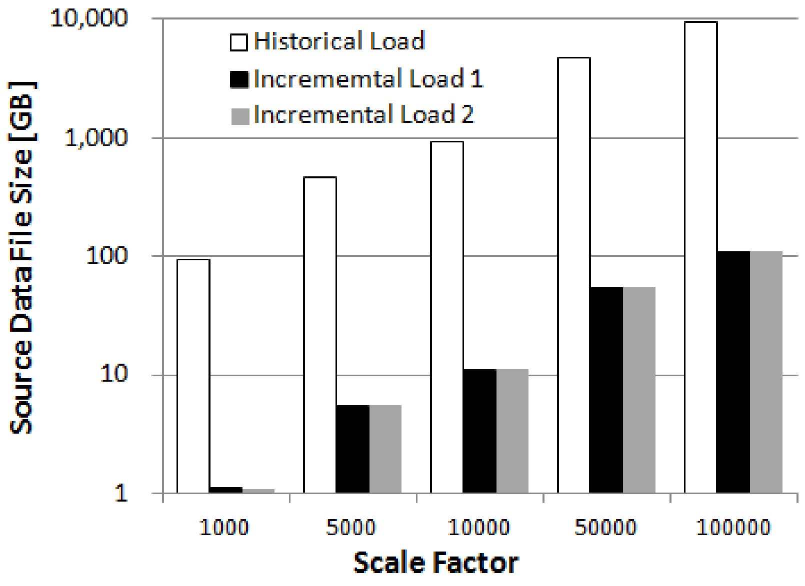
- Dimensional schema
- 5 fact tables
- 9 dimension tables
- 5 reference tables (dimensions in the strict sense of a star schema)



Data Set

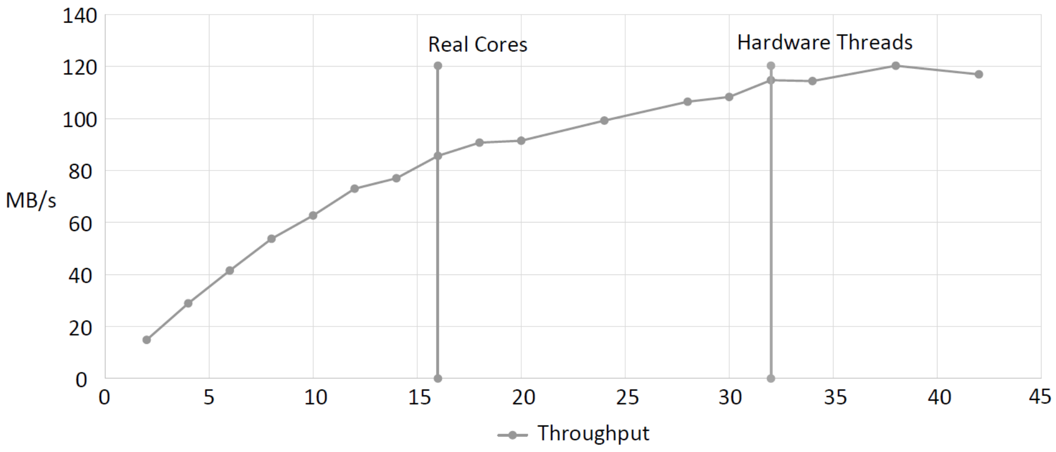
- Realistic
- Scalable
  - Continuous ($SF \in \mathbb{N}$)
  - Based on #customers
    - Historical Load: $SF * 5000$
    - Each Incremental Load: $SF * 5$
  - Domain and tuple scaling
  - Linear scaling of most tables
  - Static scaling of few tables





Data Generation (PDGF)

- Based on Parallel Data Generation Framework (PDGF)
  - Generic – can generate any schema
  - Configurable – XML configuration files for schema and output format
  - Extensible – plug-in mechanism for
    - Distributions
    - Specialized data generation formats
  - Efficient – utilizes all system resources to a maximum degree (if desired)
  - Scalable – parallel generation for modern multi-core SMPs and clustered systems



- Evaluation
  - 2 E5-2450 Intel Sockets
  - 16 cores, 32 hardware threads
  - 1-42 workers (= degree of parallelism)
  - Almost linear scale-up with cores
  - Slow down after 38 workers

18 Trans-formations

DSS Table	Characteristics of Transformations																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
$T_{H,DimAccount}$	✓	✓	✓		✓		✓	✓									
$T_{H,DimBroker}$				✓	✓												
$T_{H,DimCompany}$	✓	✓	✓	✓	✓	✓	✓	✓									
$T_{H,DimCustumer}$	✓	✓	✓	✓	✓	✓		✓		✓			✓				
$T_{H,DimDate}$					✓												
$T_{H,DimSecurity}$		✓			✓	✓	✓				✓	✓	✓		✓	✓	
$T_{H,DimTime}$					✓												
$T_{H,FactTrade}$			✓	✓					✓		✓	✓					✓
$T_{H,FactCashBalances}$				✓							✓						✓
$T_{H,FactHolding}$				✓					✓		✓	✓		✓			✓
$T_{H,FactMarketHistory}$			✓	✓						✓	✓		✓				✓
$T_{H,FactWatches}$				✓		✓		✓			✓			✓			✓
$T_{H,Industry}$				✓													
$T_{H,Financial}$						✓	✓				✓			✓		✓	
$T_{H,FactProspect}$	✓	✓															✓
$T_{H,StatusType}$				✓													
$T_{H,TaxRate}$				✓													
$T_{H,TradeType}$				✓													

1. Transfer XML to relational data
2. Update DIMessage file
3. Convert CSV to relational data
4. Merge multiple input files of the same structure
5. Convert missing values to NULL
6. Standardize entries of the input files
7. Join data from input file to dimension table
8. Perform extensive arithmetic calculations
9. Join data from multiple input files with separate structures
10. Consolidate multiple change records per day and identify most current
11. Read data from files with variable type records
12. Check data for errors or for adherence to business rules
13. Detect changes in fact data, and journaling updates to reflect current state
14. Detect changes in dimension data, and applying appropriate tracking mechanisms for history keeping dimensions
15. Filter input data according to pre-defined conditions
16. Identify new, deleted and updated records in input data
17. Join data of one input file to data from another input file with different structure

Transformations

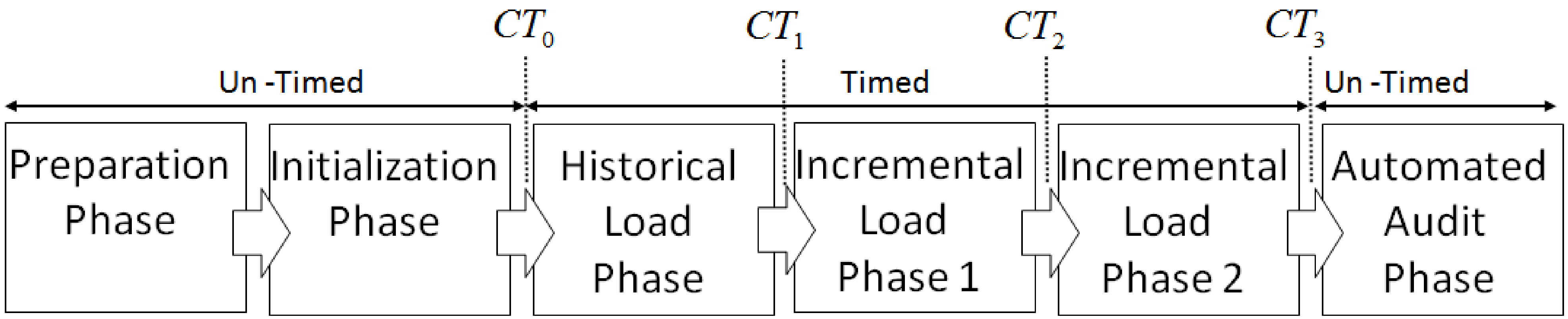
- No standard language to define
- Transformations are specified in English text
- Correctness of transformation implementations is guaranteed by:
  - Independent audit by a certified TPC auditor
  - Correctness queries run during the benchmark run and at benchmark completion
  - Qualification run on small scale factor and comparison of results with reference output

Pseudo code example: DimAccount

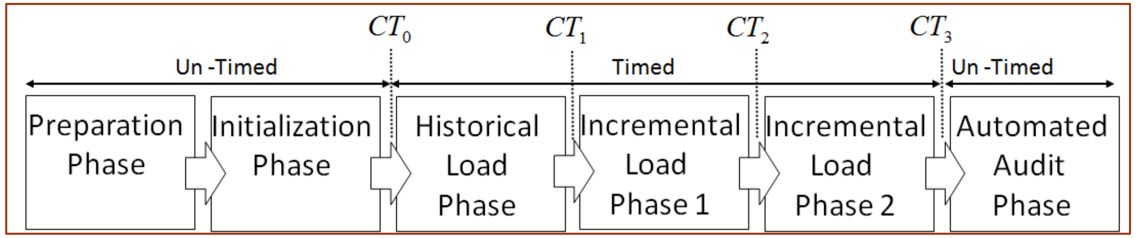
```
if @ActionType=NEW or ADDACCT then
    AccountID ← Customer/Account/@CA_ID;
    AccountDesc ← Customer/Account/@CA_NAME;
    TaxStatus ← Customer/Account/@CA_TAX_ST;
    SK_BROKER_ID ←
        SELECT BrokerID
        FROM DimAccount,DimBroker
        WHERE Customer/Account/@CA_D_ID =
        DimBroker.BrokerID ;
    SK_CUSTOMER_ID ←
        SELECT C_ID
        FROM DimAccount,DimCustomer
        WHERE Customer/Account/@C_ID=
        DimCustomer.CustomerID
        AND @ActionTS BETWEEN EffectiveDate AND
        EndDate ;
    Status ← ACTIVE;
else if @ActionType=UPDACCT then
    foreach source field with data do
        the same as for NEW and ADDACCT;
    end
    foreach source field without data do
        retain current values;
    end
else if @ActionType=CLOSEACCT then
    Status ← INACTIVE;
else if @ActionType=UPDCUST then
    foreach account held by customer do
        SK_CustomerID ← updated customer record;
    end
else if @ActionType=INACT then
    SK_CustomerID ← updated customer record;
    Status ← INACTIVE;
    foreach account held by customer do
        SK_CustomerID ← updated customer record;
        Status ← INACTIVE;
        IsCurrent, EffectiveDate, and EndDate ← according to
        algorithm to history keeping dimensions
    end
```

Algorithm 1: DimAccount Historical Load Transformation

Execution Rules



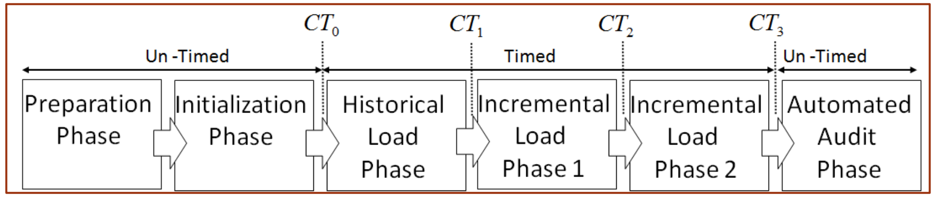
- Un-timed part prepares the system
- Timed part measures the data integration performance:
  - **Historical Load**: Initial load of decision support system from historical records or due to restructuring of decision support system
  - **Incremental Loads**: Periodic incremental updates of daily feeds
  - Two incremental loads to measure the affect of data structure maintenance
- No phase may overlap



Metric

- Before executing the benchmark a scale factor must be chosen
- Scale factor determines the amount of data inserted during each phase
- Elapsed times are defined as difference between completion times, i.e. $E_H = CT_1 - CT_0; E_{Ii} = CT_{(i+1)} - CT_i; i \in \{1,2\}$
- Elapsed time is dependent on system performance

Metric Characteristics



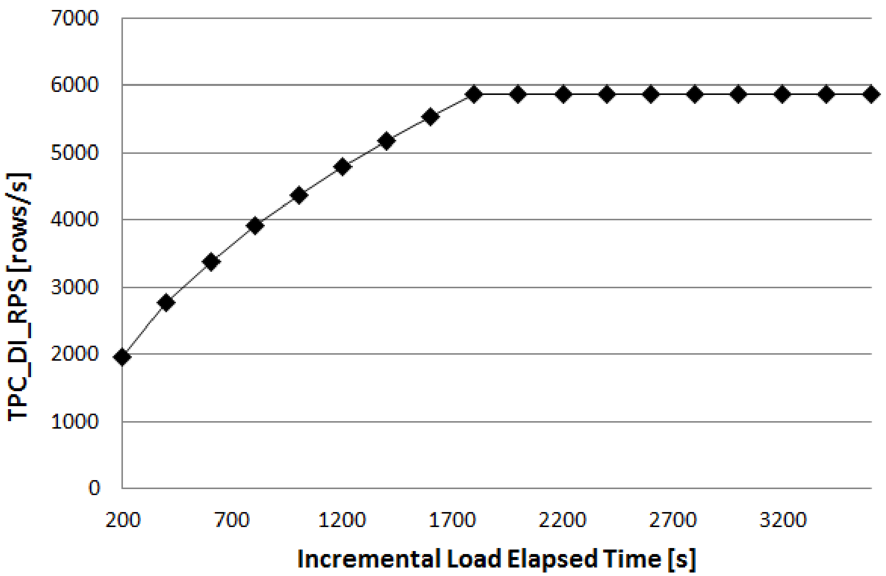
$$T_H = \frac{R_H}{E_H}; T_{I1} = \frac{R_{I1}}{\max(T_{Ei}, 1800)}; i \in \{1,2\}$$

$$TPC\_DI\_RPS = \left\lfloor \sqrt{T_H^2 \times \min(T_{I1}, T_{I2})^2} \right\rfloor$$

- One performance metric → Makes ranking of results easy
- Throughput metric: rows processed per second
- Geometric mean of the throughputs during historical and incremental load phases → entices performance improvements in all phases
  - E.g. reducing the elapsed time of the historical load from 100s to 90s has the same impact on metric as reducing the incremental load with the smaller elapsed time from 10s to 9s.

$$T_H = \frac{R_H}{E_H}; T_{I1} = \frac{R_{I1}}{\max(T_{Ei}, 1800)}; i \in \{1,2\}$$

$$TPC\_DI\_RPS = \left\lfloor \sqrt{T_H^2 \times \min(T_{I1}, T_{I2})^2} \right\rfloor$$



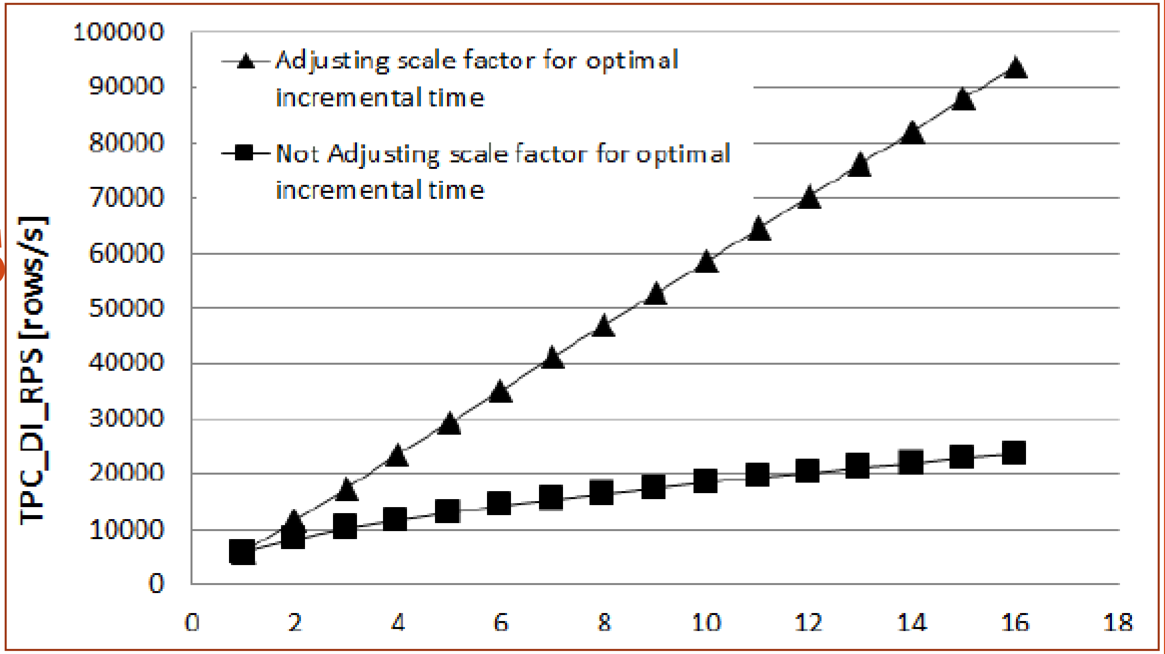
Metric Analysis

- Metric encourages the processing of a sufficiently large amount of data
- Actual amount of data processed depends on the system performance
- The higher the performance of a system, the more data it needs to process
- While the benchmark rules allow elapsed times of less than 1800s, there is a negative performance impact due to the max function in the denominator of the incremental throughput functions (T_1 and T_2)
- The above graph shows the performance of a system with load performance linear to data size.

Metric Analysis

$$T_H = \frac{R_H}{E_H}; T_{I1} = \frac{R_{I1}}{\max(T_{Ei}, 1800)}; i \in \{1,2\}$$

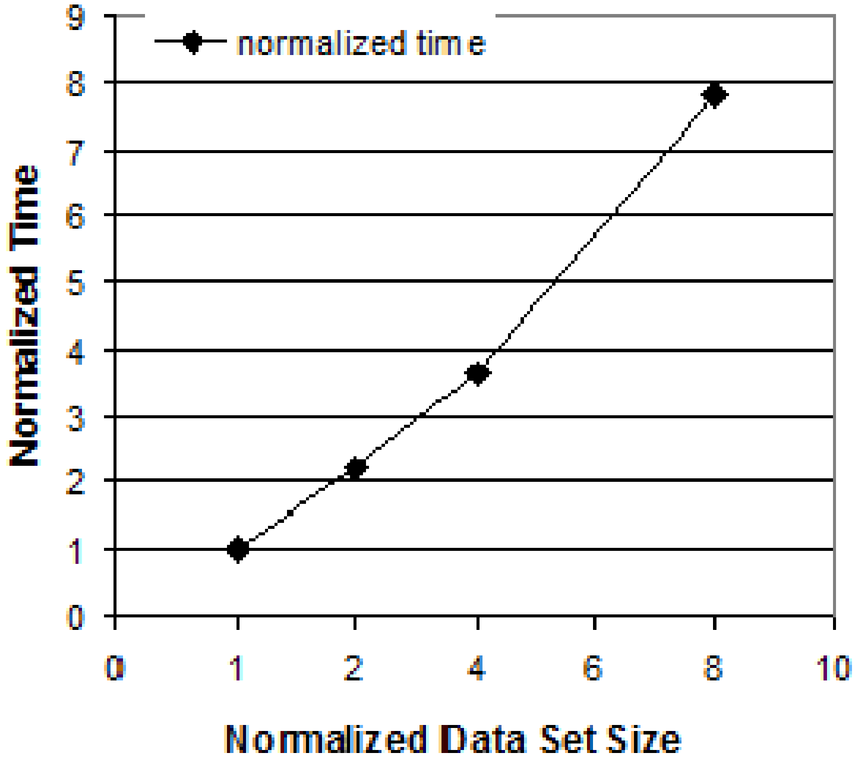
$$TPC\_DI\_RPS = \left\lfloor \sqrt{T_H^2 \times \min(T_{I1}, T_{I2})^2} \right\rfloor$$



- Metric encourages constant elapsed times of consecutive incremental loads due to min(T_1,T_2)
- Metric scales linearly with scale factor
  - Important for measuring scale-out and scale-up solutions
  - System with double the resources, e.g. CPU, memory, should show double the performance
  - This is only true if a scale factor is chosen that results in an elapsed time of 1800s.

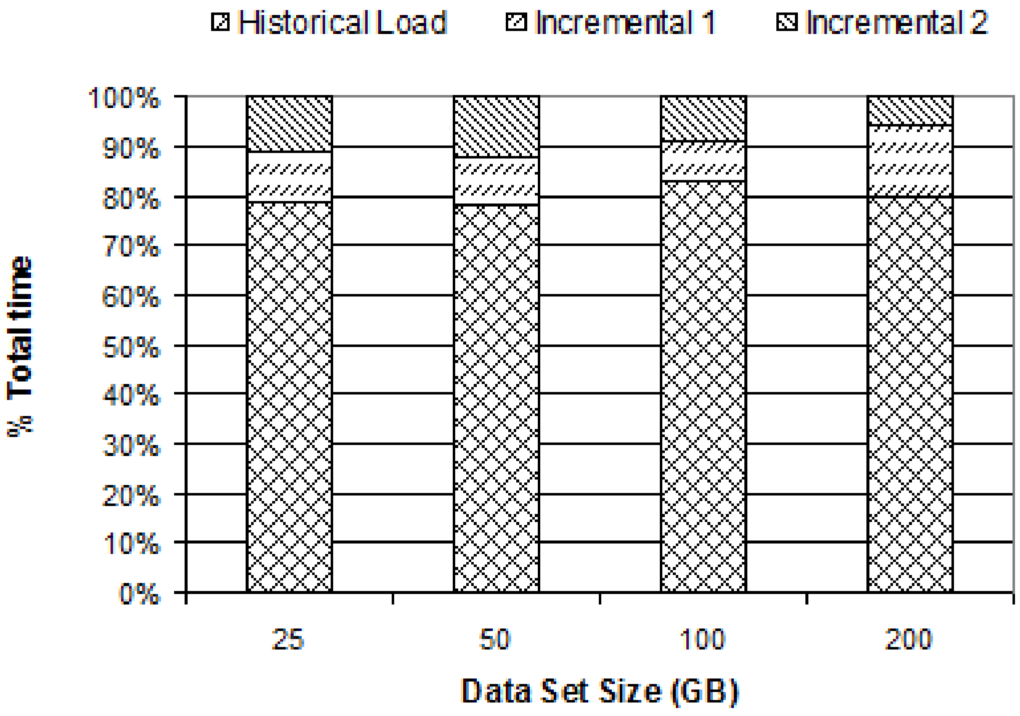
Experimental Results

- TPC-DI was run with 5 scale factors
- Figure shows normalized data
  - X-axis shows normalized data size
  - Y-axis shows normalized elapsed time
- Results show linear scalability
- In another experiment it was shown that the elapsed time remained constant when the hardware resources were scaled to match the data size, i.e. double the data size with double the number of CPU's, IO and memory



Experimental Results

- Figure shows:
  - X-axis: three different data sizes in Gigabytes
  - Y-axis: percentage of elapsed time spent in historical load, incremental load 1 and incremental load 2
- Across data size the time spent in the historical load is 80%. 20% is spent in the incremental update phases.
- This can be used to extrapolate the total elapsed time of benchmark runs.



Conclusion

- New TPC Standard Benchmark for Data Integration
  - Accepted in January 2014
- Brokerage firm business model
  - OLTP system, HR, CRM, external data
  - 18 transformations into integrated data warehouse
- Covers
  - Multiple formats
  - Historical load and updates
  - Complex interdependencies

Questions?

- Thank You!

- TPC-DI website
  - http://www.tpc.org/tpcdi/default.asp