Near Real-Time Data Integration

Active Data Warehouse Workshop
Continuous TPump in a W2K/WinXP Environment

Robert M. Bruckner
bruckner@ifs.tuwien.ac.at

Institut für Softwaretechnik & Interaktive Systeme
Technische Universität Wien

Outline

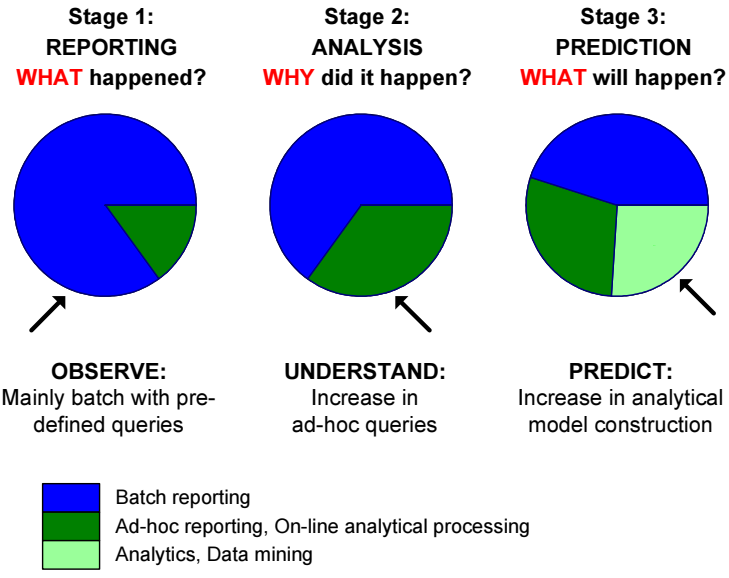
- Active Warehouse
- Continuous Data Integration
- Need for Messaging Infrastructure
- TPump in a Continuous Environment
- Message Queuing Infrastructure
  - Setting up MSMQ
    - MSMQ 2.0 (W2K)
    - MSMQ 3.0 (WinXP)
  - Setting up TPump & database
  - Setting up QTool
- QTool
  - Queue Administration
  - Data Feeding
  - TPump Job-Scheduling
- Results & Comparison
  - ADW CoE Sample, QTool
  - MSMQ, MQSeries
- Conclusion

Traditional Data Warehousing

- Collection of data to support management needs (complex analysis for strategic decisions) [Inmon 1992]:
  - subject-oriented
  - integrated
  - time-variant
  - nonvolatile
- High volumes of data
- Integration of external data
- Batch load ("update window")

- Examples of data warehouse usage:
  - Analyze product sales, stock inventory, customer behavior, etc.
  - Analyze process performance, etc.

Information Evolution in a traditional Data Warehouse Environment

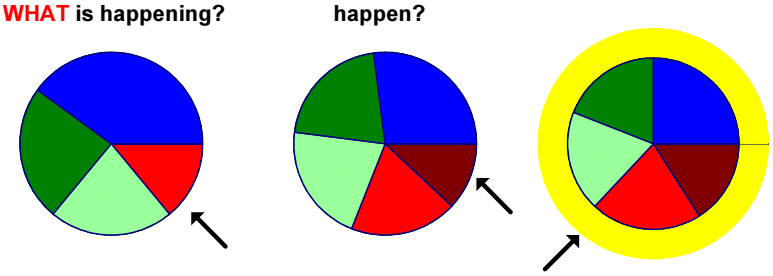


Information Evolution in an Advanced Data Warehouse Environment

Stage 4:
OPERATIONAL DATA WAREHOUSE
WHAT is happening?

Stage 5:
ACTIVE WAREHOUSE
What do I WANT to happen?

Stage 6:
MINIMIZED LATENCY
What's the FEEDBACK?



REACT:
Continuous updates and time sensitive queries gain importance

RE-ORGANIZE:
Event-based triggering

AUTOMATE & CONTROL:
Closed loop on information integration and automated information delivery

Batch reporting
Ad-hoc reporting, OLAP
Analytics, data mining

Continuous update, tactical queries
Event-based triggering, automated reactions
Notifications

What is an Active Data Warehouse?

- Traditional Data Warehousing
  - Focus on "ivory tower" decision makers
  - Long-term decision making
  - Strategic focus
- Active Data Warehousing
  - Expand scope to include "in the field" decision makers
  - Day-to-day (minute-to-minute) decision making
  - Tactical focus with strategic implications

Business needs both strategic and tactical decision support capabilities.

What is an Active Data Warehouse?

Traditional	Active
Strategic decisions only	Strategic + tactical decisions
Results sometimes hard to measure	Results measured with operations
Daily, weekly, monthly data currency acceptable; summaries often appropriate	Only comprehensive detail data available within minutes is acceptable
Limited number of users accessing the system concurrently	High number (1000+) users accessing and querying the system at the same time
Highly restrictive reporting used to confirm/check existing processes and patterns. Often using pre-built summary tables or data marts.	Flexible, ad hoc reporting as well as machine assisted modeling such as data mining to discover new hypotheses
Power users, knowledge workers, internal users	Operational staffs, call centers, external users

Business Drivers for Minimized Latency

- **Decrease the time** it takes to make the business decisions.
- Minimize latency between the **cause and effect** of a business decision.
- Notify the business of actionable **recommendations**.
- Effectively **close the gap** between business intelligence systems and business processes.

Analytical decisions integrated into operational processes combined with closed loop analytics.

Options for Data Integration 1/2

- Important requirement for Teradata Active Data Warehouse: **Ability to provide data that is close to up-to-date**

Teradata Options:

- FastLoad
  - Short, frequent FastLoad executions
  - Loading data into an empty table for later select/insert/update processing into the final target table

  You cannot use the FastLoad utility to:
  - Insert additional data rows into existing tables
  - Update individual rows of existing tables
  - Delete individual rows from existing tables
  - Load data into multiple tables

Options for Data Integration 2/2

Teradata Options (continued):

- MultiLoad
  - Frequent MultiLoad executions directly into target table
  - Issues: Efficiency, concurrency, resource consumption

- TPump
  - Data feed from message queues in a continuously executing mode:
    - Message queuing infrastructure
    - Feeding tool
    - Scheduler for continuously executing TPump jobs

Why TPump?

- Economies of scales
  - MultiLoad is not necessarily efficient when operating on large tables (→ fact table) when there are not many rows to insert or update.
  - For MultiLoad to be efficient, it must touch more than one row per data block in the Teradata database.
- Concurrency
  - No limit for TPump instances running concurrently (MultiLoad: 15 instances).
  - TPump uses row-hash locks, making concurrent updates on the same table possible (MultiLoad: table-level locks).
- Resource consumption
  - TPump allows the operator to modify the statement rate, while the job continuously runs.
  - MultiLoad is designed for the highest possible throughput, and uses any available database and host resources.

Motivations using TPump for NRT Data Integration

- TPump is suitable, when some of the data needs to be updated closer to the time the event or the transaction took place
- avoids table-level locks (row-hash locks only)
- flexibility in when and how it is executed
- queries can access a table concurrently with TPump
- several TPump jobs can run against the same table at the same time
- sources as varied as MVS, NT or UNIX concurrently

TPump as Continuous ETL Tool?

Limitations:

- Concatenation of data files is not supported. File size limit: 2 GB.
  → We will solve these by using message queues!
- Data retrieval capability from the Teradata RDBMS via SELECT statements is not allowed.
- Arithmetic functions are not supported.
- Access Logging can cause a severe performance penalty in TPump, because TPump uses normal SQL operations.

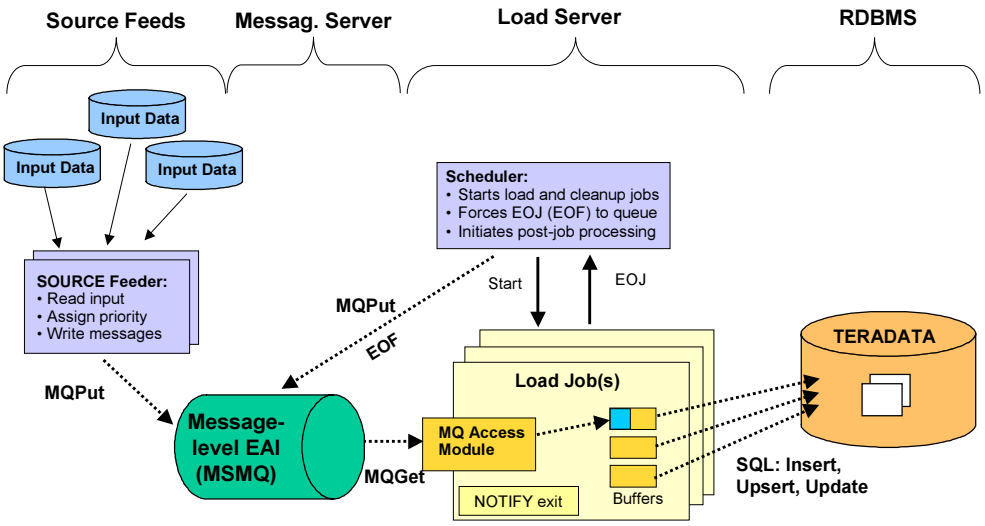
Need for a Messaging Infrastructure

Continuous data integration
→ message-level Enterprise Application Integration (EAI):

- Asynchronous, connectionless service
  (vs. RPC or socket-based communication)
- Reliable delivery, transaction support if required
- Priority-based messaging
- Decoupling from transactional systems
  → supporting dynamic data rates
  → source systems: information push
  → data integration: information pull
- Common options: IBM MQSeries®, Microsoft MQ (MSMQ)

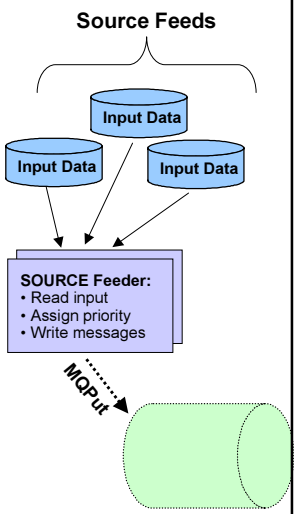
Rotating TPump Architecture

Continuous Data Integration based on Teradata:



Role of Source Feeder

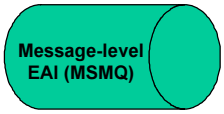
- Provide application data feeds:
- Read transaction messages from a file
- Add a timestamp to the message
- Put message into the queue
- Arrival rate is adjustable
- Discussion: state vs. event data



Role of Messaging Infrastructure

- Provide low-latency data feeds:

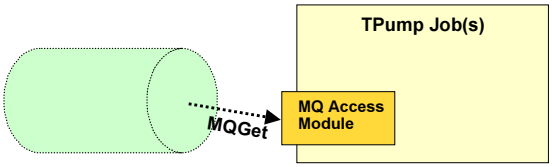
- Multiple sources can feed the queue
- Reliable delivery, transaction support if required
- Asynchronous delivery:
  - Source systems can write in (near) real-time
  - Load utility may process the message immediately or some time later
- In general: FIFO (first-in-first-out) processing of messages



Role of Access Modules

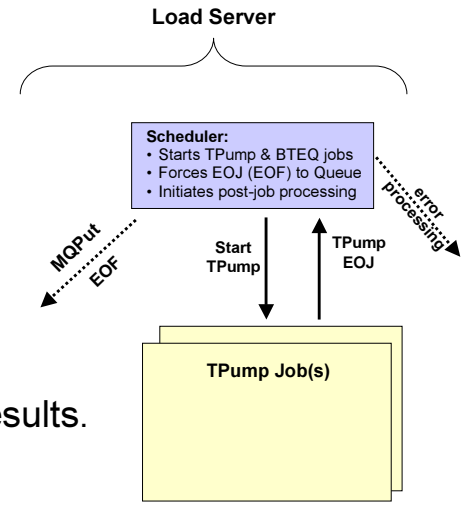
- Connect to MSMQ / MQSeries:

- Local or remote connection
- Get messages from MSMQ / MQSeries
- Add timestamp to the messages
- Stream messages to TPump
- Guaranteed reliability:
  - No lost inserts
  - No duplicate inserts

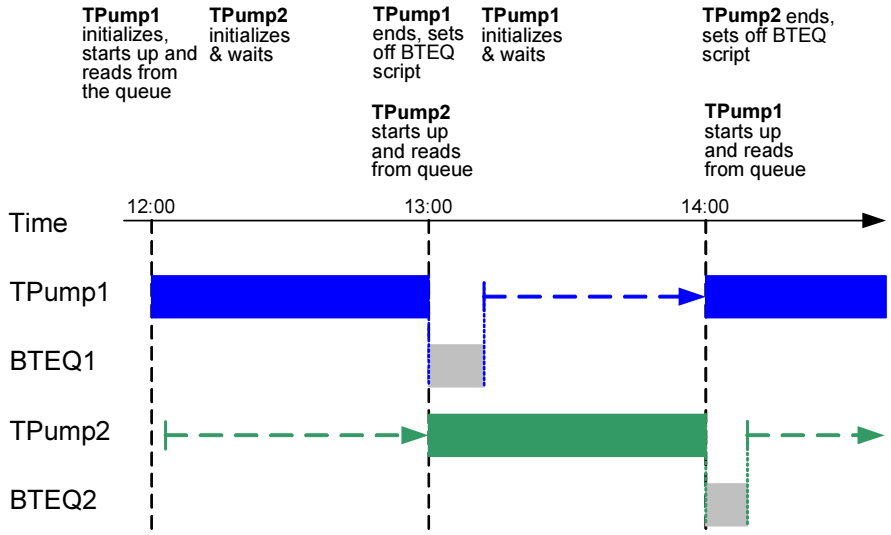


Role of Scheduler

- Coordinate two parallel TPump instances:
- Start TPump job
- End TPump job by placing an EOF message to the Queue
- Launch post-job processing: BTEQ script to consolidate error rows inside Teradata.
- Monitor load process status/results.



QTool / TPump Job-Scheduling

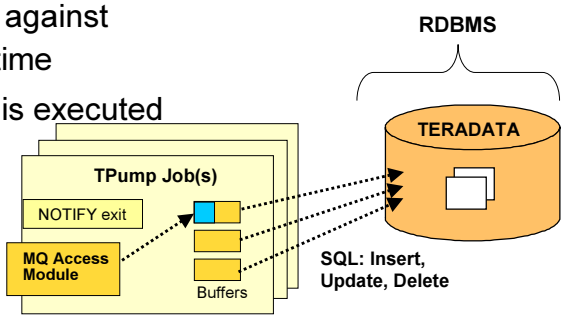


Advantages of „rotating“ TPump instances

- Quicker confirmation of success or failure of a particular load portion.
- Quicker access and reprocessing of error tables.
- Job validation & performance analysis based on end-of-job statistics (e.g. audit trail).
- Opportunity to change parameters on subsequent iterations of TPump (in order to react on changes in the environment).

Role of TPump

- Provide flexibility in continuous data loading:
- TPump is suitable, when some of the data needs to be updated closer to the time the event or the transaction took place
- avoids table-level locks (row-hash locks only)
- queries can access a table concurrently with TPump
- several TPump jobs can run against the same table at the same time
- flexibility in when and how it is executed



Outline

- Active Warehouse
- Continuous Data Integration
- Need for Messaging Infrastructure
- TPump in a Continuous Environment
- Message Queuing Infrastructure
  - Setting up MSMQ
    - MSMQ 2.0 (W2K)
    - MSMQ 3.0 (WinXP)
  - Setting up TPump & database
  - Setting up QTool
- QTool
  - Queue Administration
  - Data Feeding
  - TPump Job-Scheduling
- Results & Comparison
  - ADW CoE Sample, QTool
  - MSMQ, MQSeries
- Conclusion

MSMQ Features

- Integrated with
  - WinNT4: MSMQ 1.0
  - Win2K: MSMQ 2.0
  - WinXP: MSMQ 3.0
- Security, message persistence, transaction support
- **Public Queues** published through directory service
  - Win NT 4 / MSMQ 1.0: SQL Server 6.5
  - Win2K / MSMQ 2.0: Active Directory at domain controller
  - WinXP: Public queues without directory service possible
- **Private Queues** are not published
  - no directory service overhead
- More Details: http://www.microsoft.com/msmq

MSMQ 2.0

- Win2K Professional, Win2K Servers
- Win2K security integration (Kerberos)
- Encryption: 40 bit, 128 bit
- Active Directory integration
  (Workgroup mode is possible but tricky to setup)
- Windows Cluster (active/active) support
- 2GB storage limit per machine

- MSMQ - MQSeries bridge is available
- Cross plattform support:
  MQC (Message Queuing Connectors) for Unix, CICS/MVS, VMS, AS/400

MSMQ 2.0 Setup

- Recommended:
  Re-install MS DTC (Distributed Transaction Controller)
  → run: \winnt\system32\dtcsetup.exe

- Install MSMQ 2.0 (Win2K)
  →Settings
  → Control Panel
  → Add/Remove Programs
  → Add/Remove Windows Components
  → Install Message Queuing Services

- Create message queues using the management console snap-in
  (run: compmgmt.msc /s)

MSMQ 2.0 Workgroup Mode

- Optional: Configure MSMQ for **Workgroup Mode**

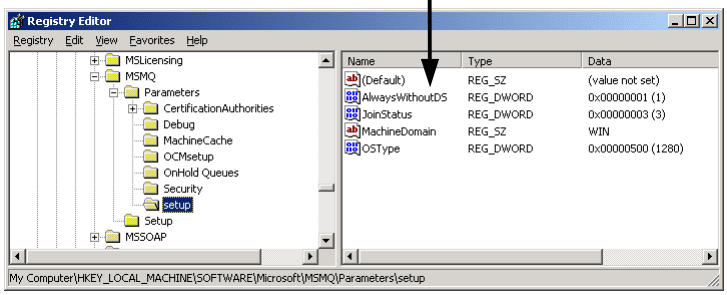
Workgroup Mode:

- Running MSMQ on Win2K within a Windows domain but **not** on the domain controller
- Enables only private queues on the local machine
- No directory service overhead

MSMQ 2.0 Workgroup Mode Setup

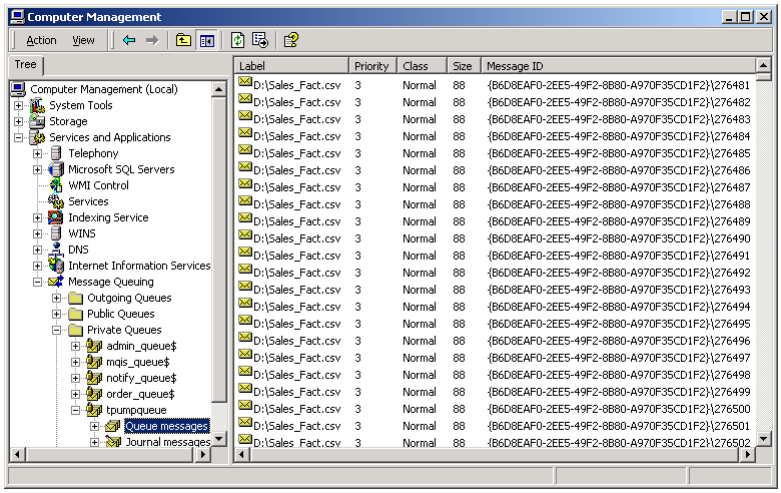
Enable Workgroup Mode:

- Modify Win2K registry:
  - HKLM \ Software \ Microsoft \ MSMQ \ Parameters \ setup
  - **Add DWord** "AlwaysWithoutDS" = 1
- Restart Win2K



MSMQ 2.0 Management

- Management Console Snap-In, run: `compmgmt.msc /s`



MSMQ 2.0 Troubleshooting

- Event ID 2124: Message Queuing was unable to join the local Windows 2000 domain.
- Event ID 2121: Unable to complete Message Queuing Setup.
- Hresult: c00e0075h

→ Re-install MS DTC

→ Configure MSMQ in Workgroup Mode

MSMQ 3.0

- WinXP Professional (available)
- WinXP Servers (current state: RC1)

- New Features
  - Messaging based on HTTP / HTTPS
  - SOAP extensions for reliable messaging (based on HTTP)
  - Network load balancing / web-farm support
  - Multicast messaging
  - Message trigger concept (based on ECA rules)

- 1TB storage limit per machine
- Easier administration & deployment

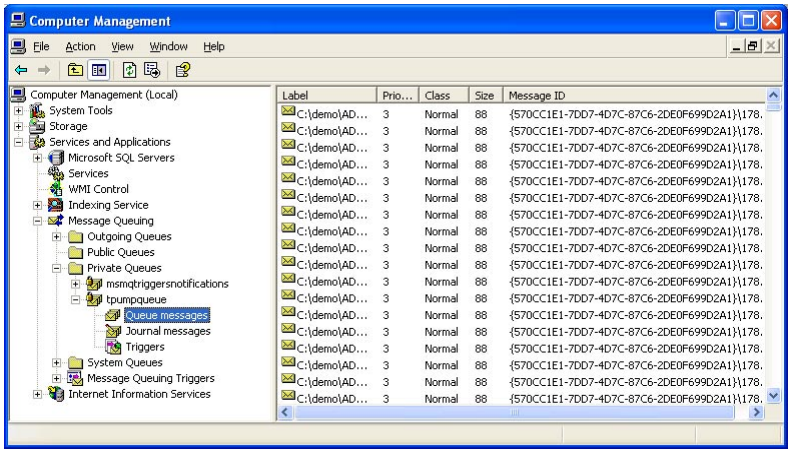
MSMQ 3.0 Setup

- Install MSMQ 3.0 (WinXP)
  → Control Panel
  → Add/Remove Programs
  → Add/Remove Windows Components
  → Install Message Queuing Services

- Create message queues using the management console snap-in (run: compmgmt.msc /s)

MSMQ 3.0 Management

- Management Console Snap-In, run: `compmgmt.msc /s`



Run Samples

- Setup messaging infrastructure
- Setup sources & TPump scripts
- Setup database (tables)
- Setup QTool

Setup TPump scripts

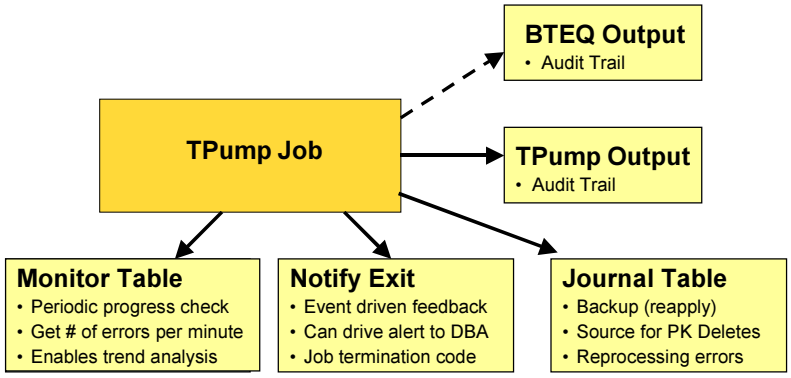
- Pack Factor **10**
  The Pack Factor is the number of statements that will be packed together into a TPump buffer and sent to the database as one multi-statement request.
- Number of sessions **20**
  (Recommendation for Teradata Demo 4.x: sessions = **1**)
- Checkpoint **30**
  Frequency (minutes) between occurencies of checkpointing
- ROBUST **ON**
  avoids re-applying rows that have already been processed in the event of a restart (data integrity).
- SERIALIZE **OFF**
  SERIALIZE ON removes deadlock potential between buffers within the same TPump job, when rows with NUPI values are being processed.

TPump: Rules of the Thumb

- Reduce pack factor in order to reduce data load latency and improve real-time availability of single rows.
- However, high pack factor and more sessions can increase throughput.

- Longer runtime (+40%) when data with errors (1%)
- Longer runtime with a NUSI (+50%), worse with 2 NUSIs (+100%); NUSI = non unique secondary index
- Longer runtime (+45%) with fallback
- SERIALIZE adds 30%, ROBUST adds 15%

Real-time information about TPump



Run Samples

- Setup messaging infrastructure
- Setup sources & TPump scripts
- Setup database (tables)
- Setup QTool

Using Teradata V2R4.x

- Verify connection to database server (TPump uses CLI):
  - Check \winnt\system32\drivers\etc\hosts
- Check for access module in \winnt\system32 (MSMQ_AXSMOD.DLL)

Using Teradata Demo 4.x

- Teradata Demo 4.0: Win2K, Win NT4
- Teradata Demo 4.1: Win2K, WinXP (Patch required!)
  Download WinXP Patch:
  http://www.teradata.com/solutions/Files4XP.zip
- Check TPump (**version 1.4.0** or later required)
  \Program Files\NCR\Teradata Client\bin\tpump.exe
- TPump parameter: **sessions = 1**
- Setup of messaging infrastructure:
  - MSMQ must be configured in **Workgroup mode** on a standalone **Win2K** machine

Outline

- Active Warehouse
- Continuous Data Integration
- Need for Messaging Infrastructure
- TPump in a Continuous Environment
- Message Queuing Infrastructure
  - Setting up MSMQ
    - MSMQ 2.0 (W2K)
    - MSMQ 3.0 (WinXP)
  - Setting up TPump & database
  - Setting up QTool

- QTool
  - Queue Administration
  - Data Feeding
  - TPump Job-Scheduling

- Results & Comparison
  - ADW CoE Sample, QTool
  - MSMQ, MQSeries

- Conclusion

QTool Overview

QTool is:

- A tool designed to enable continuously loading a data warehouse
- MSMQ management utility
- Basic job-scheduler

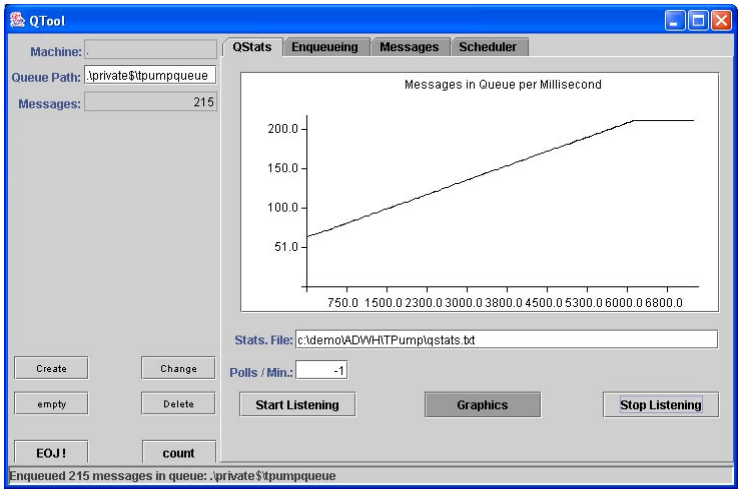
QTool is NOT:

- A messaging facility
- A DWH loading tool
- A complete, standalone solution to continuous loading

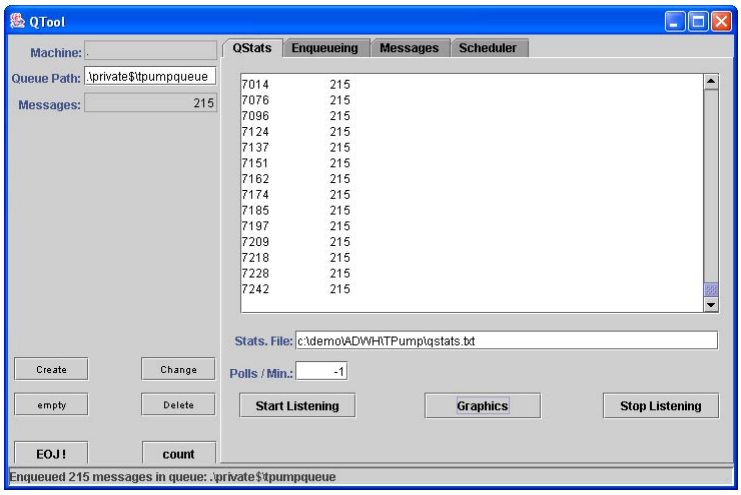
Basic Features of QTool

- Queue monitoring & statistics
- Queue creation, emptying and deletion
- Enqueuing a flat file
- Schedule a „Continuous TPump Job“
- Path configuration at runtime

Queue Monitoring



Queue Monitoring



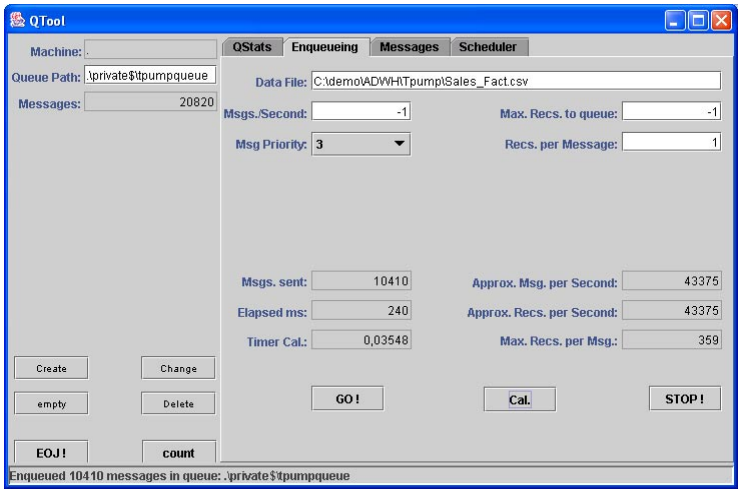
Queue Management

- Create queues
- Empty queues
- Delete queues
- Count messages in queue
  - Show graph
  - Show raw data
- Send job-rotating message (EOJ)

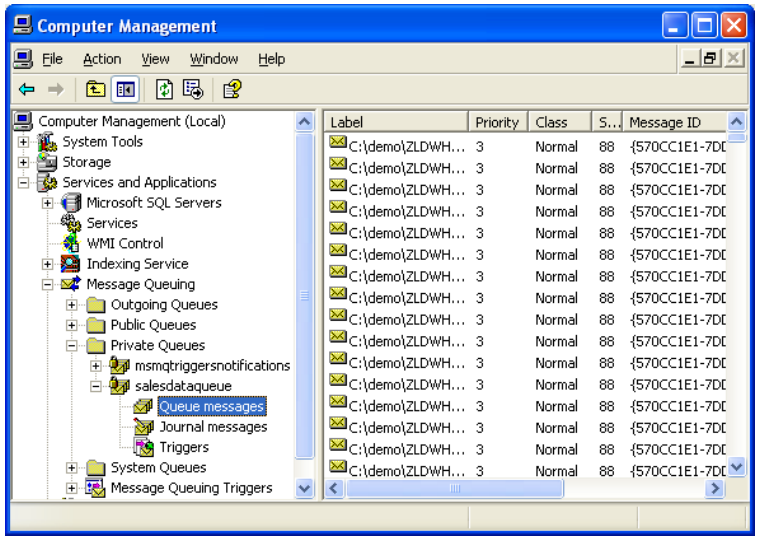
Basic Features of QTool

- Queue monitoring & statistics
- Queue creation, emptying and deletion
- Enqueuing a flat file
- Schedule a „Continuous TPump Job“
- Path configuration at runtime

Enqueuing a Data File



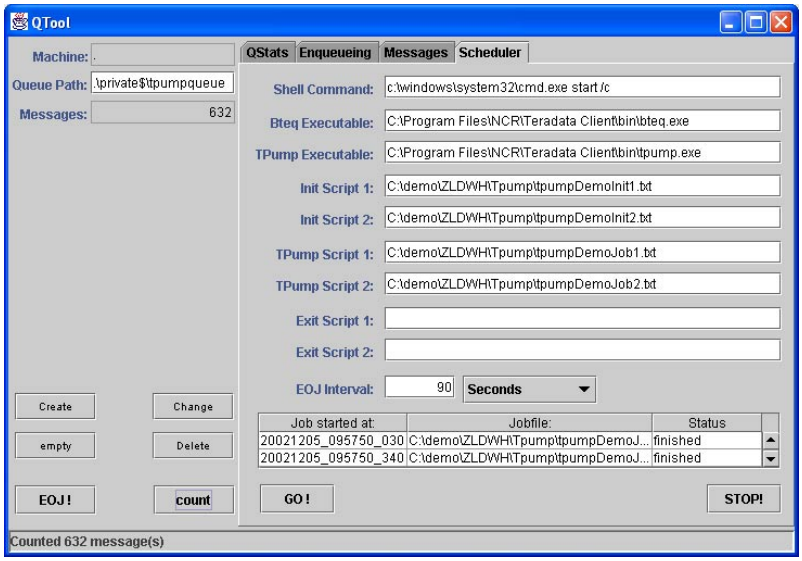
Queued Messages



Enqueuing Data

- Specify data file
- Specify queuing rate
- Specify total number of records
- Specify number of records per message
- Specify message priority
- Allows to calibrate
- Show statistics

Qtool Scheduler

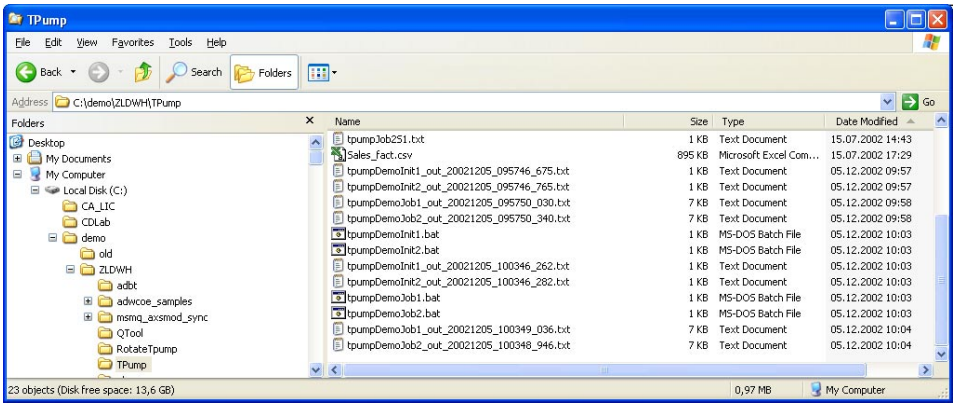


QTool TPump Scheduling

- Specify paths to executables
- Specify paths to script-files
- Specify job-rotation interval
- Shows continuously updated job info
- Check output files by double clicking into results table

Output files

- Every script generates output file to specified directory



QTool Technology

- Frontend written in Java2SE (1.3.1, 1.4.0)
- Uses Java Native Interface (JNI)
- Backend written in C/C++
  - Accesses Standard MSMQ 2.0 API through C++ COM Interface
  - Accesses MSMQ Management 2.0 API through C COM Interface

Run QTool

- Components:
  - **QTool_2_4.jar**
  - **QToolNative.dll**
    Copy DLL into \winnt\system32\
  - **MSMQ_AXSMOD.dll**
    Copy DLL of access module into \winnt\system32\
  - Installed **J2SE** runtime environment necessary
- Setup messaging infrastructure
- Setup sources
- Setup TPump scripts
- Setup database (tables)

- Run QTool:
  java –jar QTool_2_4.jar

QTool Demo

Loading additional POS data into a fact table

- Environment: Win2K / WinXP
- Teradata Demo 4.x

- QTool Version 2.4

Workshop Agenda

- Active Warehouse
- Continuous Data Integration
- Need for Messaging Infrastructure
- TPump in a Continuous Environment
- ADW CoE Sample
  - Content
  - Customization
  - Setting up MSMQ
    - MSMQ 2.0 (W2K)
    - MSMQ 3.0 (WinXP)
  - Setting up TPump & database
- Break / Discussion
- Data Feeds for MSMQ
- TPump Job-Scheduling
- QTool
  - Queue Administration
  - Data Feeding
  - TPump Job-Scheduling
- Results & Comparison
  - ADW CoE Sample, QTool
  - MSMQ, MQSeries
- Conclusion / Discussion

Comparison & Results

- QTool:
  - is up to 3 times faster than Source Feeder of ADW CoE sample.
  - one integrated tool, including queue monitoring.
  - has the ability to pack several datasets into one message – TPump automatically copes with that.
  - can assign various priorities to messages.
  - has dynamic environment configuration (instead of hard-coded paths).
- ADW CoE sample has some additional features not implemented in QTool: FDL output, Timestamps.
- MSMQ access module of the ADW CoE sample is not as sophisticated as MQSeries access module.

Results for Single Load Jobs

- High performance data integration is the real challenge:
  - **Teradata Demo 4.1**, Win XP
    Notebook: Pentium IV, 512 MB RAM
    MSMQ 3.0: ~ 260 msg./sec.
  - **Teradata V2R4.1**, 100 MBit/s Network, NCR S28 Server
    MSMQ 2.0: ~ 47 msg./sec.
    MSMQ 3.0: ~ 69 msg./sec.
  - Teradata V2R4.1, **ADSL Connection**, NCR S28 Server
    MSMQ 2.0: peaks with 20 msg./sec.

Summary

- Motivation for continuous data integration:
  having up-to-date data for better and faster decisions
- Detailed investigation of Continuous TPump Environment
- Using TPump in a MSMQ environment
- Setting up a Proof of Concept (with QTool)
- Explaining the issues & pitfalls
- Comparisons and results

Conclusion

Near real-time data integration works!

- Carefully analyze your requirements.
- Teradata provides powerful integrated load utilities (fastload, multiload, tpump).

Thank You!