Mimer SQL on OpenVMS
Present and Future

Bengt Gunne, CTO
Agenda

- Background
- Mimer customers
- Platforms
- Technical features
- Mimer SQL on OpenVMS
- Ongoing development
- Q&A
Bengt Gunne

- Started working with Mimer in 1981
  - Have worked with many parts of the system such as:
    - Database kernel
    - Transaction handling
    - Client/server communications
    - Clients such as ODBC and ADO.NET
    - Overall design
    - Multiuser systems for PDP, VMS, HP-UX, Windows etc.
    - Most recently written a new SQL optimizer
  - Head of development since 1991
  - Chief Technical Officer
Who was Mimer?

Mimer was a giant in the Norse Mythology who guarded the well of wisdom.

The gods came to Mimer for advice. When they looked into the well they could see everything that happened in the world.

The god Oden even took out one of his eyes and put it in the well to be able to see everything at all times.

Today, ordinary people come to Mimer for advice…
Mimer Information Technology AB

- HQ in Uppsala, Sweden
  - Office in Stockholm and Beijing
  - Partners in China, Japan, Korea, Central Europe, and USA
- World class experts in relational database technology
- Developer of the Mimer SQL product family
  - Enterprise Solutions
  - Industrial/Automotive/Embedded Solutions
  - Mobile Solutions
- Mimer SQL used in mission critical systems worldwide since the 1970s
What is Mimer SQL?

- Relational database management system
- Standard SQL
- Runs on many platforms
  - Tight integration with OpenVMS
Company focus

- The company focuses solely on Mimer SQL
  - Tools are through third party integrations
- How is this possible/facilitated?
  - Standard SQL
  - Standard programming interfaces

- Requires a truly open system
Mimer SQL background

- Roots at Uppsala University, Sweden
- 35+ years on VMS! First port 1980 (VAX780)
- Close co-operation with VSI
- OpenVMS main development platform since 1982
  - VAX
  - Alpha
  - Integrity
  - and now…
Mimer SQL port to OpenVMS on x86

- Will be done using prerelease from VSI
- Close in time with OS release to customers
Agenda

✓ Background
  - Mimer customers
  - Platforms
  - Technical features
  - Mimer SQL on OpenVMS
  - Ongoing development
  - Q&A
Some Customers
**Mimer SQL in Life-Critical System**

- World’s largest centralized blood supply management system
- NHS Blood and Transplant service, England & Wales
  - Responsible for **all** donated blood and tissues
  - System controls 2.5 million donations annually
  - 3,500+ active users
  - Database of 1 billion records
    +8 million / week
- “Our experience is that Mimer SQL rarely gives us a problem and has a very low maintenance overhead, so there’s no need for any expensive database administrators to run the system.”
  Ian Henderson, MD at Savant Enterprises, the developer of PULSE
NHS Blood and Transplant
An Integrated System

“NHS Blood & Transplant provides a life-saving service. It cannot fail, especially in the face of major incidents. NHSBT has achieved an extremely high level of disaster tolerance through the use of dual data centers and split-site OpenVMS clusters.

... In summary, this system demonstrates considerable in-depth strength to deliver extremely high availability blood-product services to NHSBT by using the PULSE software, the Mimer SQL database, and the OpenVMS clusters running on Integrity Server systems and EVA storage subsystems.”

Availability Digest, Oct 2008
Mimer SQL at NHS

- **Database Server**
  - HP Integrity servers at 2 locations
  - OpenVMS
  - Mimer SQL database
  - Serving data through ODBC

- **COM+ Windows server**
  - Providing “middleware” data services to clients

- **Clients**
  - Windows PC (thin client through Citrix)
  - HTML web pages
  - Web services to third party clients
Mimer SQL at NHS BT
Vital Statistics (July 21 2011)

- **Live System**
  - 1,251,033,111 records in 715 tables
  - Largest table (donation audit) is 130+ million records

- **Archive System**
  - For older, time expired records
  - 2,065,863,274 records
• Maintained 99.9% uptime
• Reduced data centre footprint
• Maintained exceptional reliability
• Built disaster-tolerant solution
Mimer SQL in Production

- Controls flow of parts to the assembly lines at the Volvo car plant in Gothenburg
- Mission critical (unplanned database downtime very expensive)
- 450 database servers for different purposes
- 250,000 cars annually
- In production since 30+ years
Mimer SQL in Life-Critical System

- National Blood Service, England & Wales
- Nation-wide system responsible for all donated blood and tissues
  - System controls 2.5 million donations annually
  - 3,500+ active users
  - Database of 1 billion records
    +8 million / week in a single table
- Runs in OpenVMS cluster
- 99.9% uptime
Mimer Installations of Mission critical systems within Telecom industry

- System type: Managing SMS for operators
- Transaction volume total: Two digit billions of SMS every month
- Number of operators: 29
- Number of servers: 178
- Number of processors: 700 (more than 2100 cores)
- Support incidents since 2014: 1

- All systems migrated from Oracle/Rdb to Mimer SQL
- Runs in cluster environment to achieve redundancy
Agenda

- Background
- Mimer customers
  - Products
  - Technical features
  - Mimer SQL on OpenVMS
  - Ongoing development
  - Q&A
Mimer SQL in different environments

- Enterprise
- Mobile
- Embedded
- Real-time

Single source code
Mimer SQL positioning

- Mimer SQL
- Home-made solutions
- "Light" DBMSes
- Oracle
- MS SQL Server
Mimer SQL platforms

- Same database kernel on all platforms
  - Enterprise server on a mobile phone
  - Small footprint and zero maintenance enterprise server
Mimer SQL in different environments

- Enterprise
- Mobile
- Real-time
- Embedded
Rich functionality

- Even the smallest editions of Mimer SQL contains functionality only found in enterprise class databases
  - Stored procedures
  - Views
  - Statement triggers
  - Instead of triggers
  - Foreign keys
  - Constraints
  - Union/Intersect/Except/Unique
  - Distinct
  - Sub-queries
  - Left/Right/Full Outer join
  - Precompiled SQL
  - Multi-user
  - Client/Server based
  - ...

- Optional:
  - Collations
  - Sequences
  - Large objects
  - Auto-upgrade
Phones with Operating systems from Enea OSE, Symbian, Windows, and Android with Mimer Inside

Included in more than 125.000.000 devices.
Mimer SQL on Android: Dynamic Database Switch

- Different applications/providers may use different database engines
  - Application/Provider does not know which database is used
  - Runtime switch
  - XML configuration
Mimer SQL on Android

- **Same Apps**
  - No changes
  - Switch in Android framework

- Database Switch
- Content Provider
- Sqlite
- Mimer SQL Client
- Database Switch
- Mimer SQL Client
- Sqlite

- Contacts
- Email
- Messaging
- Calendar
- Media
Performance on Android

Mimer SQL vs SQLite

<table>
<thead>
<tr>
<th>Activity</th>
<th>Sqlite</th>
<th>Mimer SQL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contacts</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Calendar</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>SMS</td>
<td>26</td>
<td>10</td>
</tr>
<tr>
<td>Email</td>
<td>15</td>
<td>10</td>
</tr>
</tbody>
</table>
Mimer SQL in different environments

- Enterprise
- Mobile
- Embedded
- Real-time
Real-time databases

- Customer problems
  - Amount of data constantly growing
  - Many different mechanisms used
  - Need to make new real-time analysis for each implementation
  - Hard to perform analysis and aggregation of data
Mimer SQL Real-Time Edition

- A "true"-real-time edition of Mimer SQL
  - Real-Time database operations has...
    - ... **predictable** behavior
    - ... **guaranteed worst-case response-time**
    - ... **full control** of I/O and blocking

- Safe sharing of real-time and non-real-time data
  - **All** system data modeled in one database
Mimer SQL Real-time Architecture

Ordinary SQL applications

Database Server

Real-time applications

Guaranteed response times

Database Cache
Mimer SQL Real-Time – Performance

Mimer SQL real-time introduces as little as 3% overhead compared to using shared variables!
Aggregating real-time information

- Real-time sensor code unaware of trigger
- Trigger calls SQL stored procedure to process data
- Procedure customizable by user
Aggregating real-time information

- Procedure writes to log table
  - SQL access to history data
User controlled aggregation

- User written code gets the aggregated information
  - Frequency can be varied
- Can, for example, write aggregated data to network
Agenda

✓ Background
✓ Mimer customers
✓ Platforms
  - Technical features
  - Mimer SQL on OpenVMS
  - Ongoing development
  - Q&A
Mimer SQL foundation

- Rich SQL support
- Performance
  - Scalability
- Availability
  - Reorganization free
  - Online backup
- Ease of use
  - Self-maintaining
  - Few system parameters
Mimer SQL Standard APIs

- Open architecture and APIs

- .NET
  ADO.NET, ADO.NET CF

- Python, Perl, PHP, Ruby, ...

- C/C++
  ODBC, Micro API, Embedded SQL

- Java
  JDBC, J2ME

Database

Mimer SQL Database Server
Visual Studio integration
Visual Studio documentation
DbVisualizer
DbVisualizer – SQL explain
Mimer SQL: Zero maintenance

- Automatic database reorganization
- Non-locking concurrency control
- Very few tuning parameters
Mimer SQL: Availability (i)

- Database shadowing
  - One or more copies of the data
- Transparent fail-over to shadow on error
  - Applications unaffected
Mimer SQL: Availability (ii)

- Zero down-time for database reorganizations
- Online backup
  - With Mimer SQL commands
  - or OS backup utilities
Mimer SQL: Availability (iii)

- **Immediate restart**
  - Allows the database to be accessible directly after a system crash even when the database size is several 100 GB
  - Can be used with dual ported disks or cluster with shared disks to start the server from another node after a fail-over
Logical independence

Original configuration: table XYZ

Modified table structure:

Instead of trigger
Distributed transaction handling

- Allows operations to be performed over several database systems as one single unit (transaction)
  - Can be several Mimer SQL databases
  - Mixed vendor transactions also possible

Standard: X/Open Distributed Transaction Processing Model
Spatial data

- Developed specifically for small footprint environments
- Data types:
  - Latitude and Longitude
  - Location
  - Coordinate
- Location index
  - completely maintenance free
  - not sensitive to order of operations
  - predictable with regards to response times
  - very space efficient
Linguistic Sorting and Searching

- Built-in and user defined collations
  - 140+ different built-in languages
- Text search and sorting:
  - Independent of case
  - With national characters in correct order
  - With or without regard for accents
- Index according to collation
- Word search and indexing
- Pinyin search and indexing
- User defined collations
Some built-in collations

- AFRIKAANS
- ALBANIAN
- ARABIC
- ARUMANIAN
- ASTURIAN
- BASQUE
- BELARUSIAN
- BOSNIAN
- BRETON
- BULGARIAN
- CATALAN
- AZERBAIJANI
- CORSICAN
- CROATIAN
- CZECH
- DANISH
- DUTCH
- ENGLISH
- EOR
- ESPERANTO
- ESTONIAN
- FAROESE
- FILIPINO
- FINNISH
- FRENCH
- FRISIAN
- FRIULIAN
- GALICIAN
- GERMAN
- GREEK
- GREENLANDIC
- HEBREW
- HUNGARIAN
- ICELANDIC
- IRISH_GAELIC
- ITALIAN
- KAZAKH
- KIRGHIZ
- KURDISH
- LAT
- LATVIAN
- LITHUANIAN
- LUXEMBOURGISH
- MACEDONIAN
- MALTESE
- MOLDAVIAN
- NORWEGIAN
- OCCITAN
- POLISH
- PORTUGUESE
- ROMANIAN
- ROMANsch
- RUSSIAN
- SAMI
- SCOTS
- SCOTTISH_GAELIC
- SERBIAN
- SLOVAK
- SLOVENIAN
- SORBIAN
- SPANISH
- SWEDISH
- TATAR
- THAI
- TURKISH
- TURKMEN
- UKRAINIAN
- UNICODE
- UZBEK
- VIETNAMESE
- WELSH
Some additional collations

- Indian languages
  - Assamese
  - Bengali
  - Gujarati
  - Hindi
  - Kannada
  - Konkani
  - Malayalam
  - Manipuri
  - Marathi
  - Nepali
  - Oriya
  - Punjabi
  - Sanskrit
  - Sinhala
  - Tamil
  - Telugu

- African Languages
  - Hausa
  - Igbo
  - Yoruba

- Middle East languages
  - Dari
  - Pashto
  - Persian

- Chinese
  - KangXi (康熙)
  - Pinyin (拼音)
  - ZhuYin (注音)
  - WuBiHua (五笔画)

- Korean
  - Hangul and Chinese together

- Japanese
Multilingual Support

SQL> select word from t order by word collate japanese_3;

WORD
====
ていねい - teinei - polite
テープ - teepu - tape
でぐち - deguchi - exit
テスト - tesuto - test
では - dewa - well, then
デパート - depaato - dep. store

SQL> select * from t where name like 'AA%' collate danish_1;

NAME
====
Aalborg
Århus
Advanced search using collations

- T9 sorting for numeric keypads
- Can also be used with Pinyin

```
select * from state where name like '83%' collate t9
```

<table>
<thead>
<tr>
<th>NAME</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Texas</td>
<td>Vermont</td>
</tr>
</tbody>
</table>

2 rows found

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>∞</td>
<td>ABC</td>
<td>DEF</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>GHI</td>
<td>JKL</td>
<td>MNO</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>PQRS</td>
<td>TUV</td>
<td>WXYZ</td>
</tr>
</tbody>
</table>

8 matches T, U, and V
3 matches D, E, and F
Agenda

✓ Background
✓ Mimer customers
✓ Platforms
✓ Technical features
  ▪ Mimer SQL on OpenVMS
  ▪ Ongoing development
  ▪ Q&A
Mimer SQL in different environments

Enterprise

Mobile

Mimer SQL

Real-time

Embedded
Mimer SQL on OpenVMS

- Developed on OpenVMS for OpenVMS
- Not a Unix or Windows product ported to OpenVMS!
- Tightly integrated on OpenVMS to get best possible performance and scalability
Mimer SQL Server communication

Remote client
- TCP/IP

Local clients
- Executive mode User-Written System Service
- Shared memory based
- Avoids unnecessary copying of data
- $HIBER/$WAKE synchronization

Mimer SQL Database Server
Mimer SQL: Database cache

- Stores large parts of database in main memory
- 64-bit size
- Possible to use Reserved Memory Registry in OpenVMS
  - No paging
  - No working set quota taxation
  - Reserved at boot time
Mimer SQL in a cluster (i)

- Application configured to run local communication on cluster node 2 and TCP/IP from node 1
- Done outside application
Mimer SQL in a cluster (ii)

- Database server can be automatically restarted on node 1 if node 2 fails.
- Server gives instantaneous access to data after database server start (called immediate restart!)
Mimer SQL in a cluster (iii)

- Two database servers handle separate data
- Each database operation towards a single database
- Transactions can span both databases
Mimer SQL Version 11.0

- New storage engine
  - Database cache from 16 GB -> any size
  - File size from 8 TB -> no limit
  - New efficient storage formats and block sizes

- New SQL functionality
  - Full outer join
  - Unique predicate
  - Etc.

- Beta version available for OpenVMS on Itanium at: http://developer.mimer.com/
Agenda

✓ Background
✓ Mimer customers
✓ Platforms
✓ Technical features
✓ Mimer SQL on OpenVMS
  ▪ Ongoing development
  ▪ Q&A
Ongoing - Hot standby

- Automatic failover
  - Actual triggering logic platform dependent
- Can be combined with in-memory option
- Next
  - Read/write access in all databases server instances
Ongoing - Rdb migration tools

- Support for Module SQL
- Tool that translates from SQL dialect used by Oracle/Rdb to Mimer SQL
  - Translate SQL scripts
  - Translate Embedded SQL source
  - Translate entire Module SQL files
  - Perform ad-hoc translation of a SQL command
- Customer driven
Ongoing – In memory solutions

- Migrating from small footprint versions
  - Database schema “linked” into database server
- Schema and data in database files
  - Possible to memory map database files
  - Optionally save at shutdown
- Combine with hot standby
Ongoing - Parallel execution

- Will allow a single query to be executed by several server threads
- Optimizer controlled concurrency and plan
- Parallel joins
  - Inner join
  - Merge join
- Parallel sorts
- Straightforward to extend current architecture
Ongoing - Machine learning

1. Facilitate access to data in Mimer SQL
   - New Python database client

2. Allow deployment of trained neural networks etc. within database server
   - Deployed as functions within the database server
   - Accessible from SQL
   - Advanced indexing techniques

3. Allow training within database server
   - Allows incremental refinement as more data is added
Agenda

✓ Background
✓ Mimer customers
✓ Platforms
✓ Technical features
✓ Mimer SQL on OpenVMS
✓ Ongoing development

▪ Q&A