Robust Real-time Query Processing with QStream

Sven Schmidt, Thomas Legler, Sebastian Schär, Wolfgang Lehner
Dresden University of Technology
Database Technology Group

VLDB 2005
Trondheim, Norway
August 29–September 3, 2005
QoS–based DSMS Query Processing

• **Focus**
  - QoS requirements on a per query basis
  - Guaranteed result quality of DSMS queries

• **(Real-)time dependent QoS requirements**
  - throughput
  - output delay

• **Data–dependent QoS requirements**
  - precision
  - sampling rate

• **Concept**
  - set of basic operators form operator network (queries)
  - description of resource consumption / quality influence of each operator
  - calculation of overall resource requirements
QoS Management / Resource Reservation

Data Stream(s)

Standing Queries

statistics (data)
- data rate
- jitter

statistics (time)
- processing time
- jitter

QoS
- output delay
- scheduling strategy

Min Delay
Max Throughput

Max Data Rate
Avg Data Rate

period length P

initial buffer size b

CPU utilization c

final buffer size b*

b* \leq \text{Mem AND } c < \text{CPU}

RTOS Resource Manager

Mem
CPU
Adaptation

Data Stream(s)

Standing Queries

- statistics (data)
  - data rate
  - jitter

- statistics (time)
  - processing time
  - jitter

update

- recalculation of resource requirements
- deviation between original and current resource requirements
- indication of misconfiguration by error signals (operator starvation, buffer overrun, ...)

threshold ...

... for DSMS adaptation

- changing the operator network speed
- adjusting jitter tolerance by
  - reallocation of data exchange queues
  - recalculation of operator period lengths

admission control

\[ b^* \leq \text{Mem AND } c < \text{CPU} \]

<table>
<thead>
<tr>
<th>Mem</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTOS Resource Manager</td>
<td></td>
</tr>
</tbody>
</table>
Adaptation Framework

- Controller
  - queryStats
  - returnStats

- Statmon
  - queryStats
  - returnStats

- DSMS Engine
  - adaptation

- Initial resource reservation
- Operator scheduling

Trondheim
August 29 – September 3, 2005
Robust Real-time Query Processing with QStream
Adaptation Framework (cont.)

- **Controlling the operators**

  - initial resource reservation
  - operator scheduling

  ![Diagram]

  - adaptation

  - stop periodic operator work
  - set new period length
  - set new priority
  - start periodic operator work

  - create
  - init
  - start

  - real-time task
    - periodic operator work

  - operator process
Monitoring: two-layered Statistics Collection

- Monitoring: two-layered Statistics Collection

**DCS Repository**
- separate (real-time) application
- stores window-based statistics
- network and stream analyses

**DCS Collector**
- integrated into operator network
- gathers and preprocesses stats
- sends statistics periodically to DSC Repository

**Statmon**
- periodic transfer to DSC Repository
  - monitor task
    - run time / data rate measurement
    - collection + aggregation
  - real-time task
    - operator process
Robustness

- **Concept**
  - running network \(\rightarrow\) pause/adapt/resume \(\rightarrow\) running network
  - **configuration 1**
    (jitter tolerance, resource consumption, ...)
  - **configuration 2**

- **Robustness**
  - abstract measure of system steadiness
  - **general assumption**
    - the more (initial) resources the user spends, the higher the robustness
    - the less the data stream fluctuations are, the higher the robustness

...may be mapped to the number of adaptations per time unit

\[\text{number of adaptations} \uparrow \quad \text{granted resources} \downarrow\]
Data Rate Scheduling

- **Avg Data Rate (ADR)**
  - Period length $P$ determined by avg data rate ($P$ “large“)
  - Leads to lower CPU utilization
  - Jitter tolerance given as a cumulated value; must be large enough for covering all jitter
  -> compensated by buffers (large buffers)
  - Continuous, uninterrupted data flow (no buffer access errors)

- **Max Data Rate (MDR)**
  - Period length $P$ determined by max data rate ($P$ “small“)
  - Leads to higher CPU utilization
  - No jitter to cope with
  - Intermediate buffers only for adjusting batch sizes (small buffers)
  - Data flow permanently interrupted by empty buffers
  - As soon as data becomes available it will be processed with QoS guarantees
QStream Demonstration Outline

- Resource calculation strategies for CQs
- Statistics collection
- Adaptation of our real-time DSMS
- Trading between robustness and granted resources (QoS measure)

- Wednesday, 14:00–15:30
- Friday, 11:00–12:30