Combining Data Independence and Realtime in Media Servers

Klaus Meyer-Wegener
TU Dresden, Fakultät Informatik, Institut für Systemarchitektur
kmw@acm.org

The goal: data independence and realtime

Situation however:

• Object-oriented and object-relational database systems with their extensions (classes, user-defined data types) offer data independence, but not realtime support

• Media servers offer (soft) realtime support, but not data independence
Data independence

- Playout of media objects
  - output format ≠ storage format
- Media-object types, or classes, resp.
  - encapsulation
- Access operations
  - "present image as GIF in black and white, 4 x 3"
- Conversions
- VirtualMedia project
  - expressions to define result, view
- Converter graphs

Converter graphs

- Marder, Univ. of Kaiserslautern, 1998

Diagram:

- Video of speech
- decompose
- stereo to mono
- conversion to WAV format
- client application
- null
- image
- sound
Converter graphs (2)

- Implementation of **access operation** on a media-data object
- For an operation, **several equivalent** filter graphs conceivable
- **Selection** by the server with consideration of:
  - the set of available filters
  - their resource requirements

Realtime

- Placement on secondary storage
  - striping, interleaving, ....
  - RAID
- Scheduling for input/output operations
  - EDF, Scan, Group Sweeping, ....
- Admission control
- Quality of service, reservation
- Open: shared use of resources, non-exclusive reservation, process structures
Jitter-constrained periodic streams

- Hamann, TU Dresden, 1996
- Description:
  - period
  - maximum deviation or lateness (too early, too late)
  - minimum distance

\[ \begin{array}{c}
T \\
D \\
\tau \\
\tau'
\end{array} \]

- allows to compute several values (e.g. max. burst rate)
- use for converter input and output, meta data

Statistical rate-monotonic scheduling

- Azer Bestavros, Boston University, 1998
- Task description:
  - period
  - probability distribution function for resource requirements per period (per job)
  - quality of service (= percentage of jobs that finish in time)
- Scheduler sets *allowance* (budget frame) depending on other tasks
- Admission control
A conflict occurs, if during an access operation the same object is accessed again.


- $P(\text{DBK})$ – DB conflict
- $P(\text{PNK})$ – concurrent-use conflict

Probabilities of conflict (2)

- Database conflict
  \[ P(\text{DBK}) = P(S) \cdot (1 + P(L)) \cdot \sum_{i=1}^{M} P(ZO_i) \cdot (1 - (1 - P(ZO_i))^{ZWZ}) \]
- Concurrent-use conflict
  \[ P(\text{PNK}) = P(L)^2 \cdot \sum_{i=1}^{M} P(ZO_i) \cdot (1 - (1 - P(ZO_i))^{ZWZ}) \]

with

\[ ZWZ = \frac{N \cdot t}{t + tD} \]

\[ t = \sum_{i=I,II,III,IV} \frac{P(L)}{P(S)} \cdot tL + \sum_{i=I,II,III,IV} \frac{P(S)}{P(S)} \cdot tS \]

\[ P(ZO_i) = 1 / \left( \sum_{k=1}^{M} 1/k \right) \]
Project goal

- (Embedded in group of projects on system design automation)
- Specify and implement **building blocks** of a server architecture that can be composed to form particular media servers
- Focus: building blocks with **realtime support**
- Data independence of applications requires **conversions** of data, if necessary in realtime

Project goal (2)

- Running example: video stored without compression (lossless), play out in MPEG-2 or H.263 format
- Utilize **converter** building blocks, if necessary in combination
- Represented as **converter graphs** (see above)
Converter description

• **Functional capabilities:**
  – input format (e.g. MPEG)
  – output format (e.g. WAV)
• **Resource requirements:**
  – jitter-constrained periodic streams
  – probability distribution function
  – each for input and output
• **Data structure** holds all properties of a converter
• **Meta data** for the media objects stored

Converter description (2)

Sketch of inquiry functions (for resource management):

- **conversion_capability**: → { (source_format, (destination_format)) }
- **scale_type** = { resolution, precision, frequency }
- **scaling_capability**: → { (format, scale_type) }
- **composition_capability**: → { {(source_format), destination_format) }
- **decomposition_capability**: → { (source_format, (destination_format)) }
- **compression_capability**: → { encoding }
- **decompression_capability**: → { encoding }
- **internal_buffer_needed**: (source_quality_descriptor, target_quality_descriptor) → buffer_size
Converter descriptions (3)

Sketch of inquiry functions (cont.):
- **change_of_data_volume**: \((\text{source	extunderscore quality	extunderscore descriptor}, \text{target	extunderscore quality	extunderscore descriptor}) \rightarrow \text{scaling	extunderscore factor})\)
- **complexity**: \((\text{source	extunderscore quality	extunderscore descriptor}, \text{target	extunderscore quality	extunderscore descriptor}) \rightarrow "O(f(n))")\)
- **cpu_usage**: \((\text{source	extunderscore quality	extunderscore descriptor}, \text{target	extunderscore quality	extunderscore descriptor}) \rightarrow \text{usage	extunderscore factor})\)

Operating-system interface

- Map converters to **processes or threads**
- Cooperating resource managers
  (concept introduced by OS group)
  - mapping of resources
  - number of processor cycles must be known
- Access data via **storage manager**:
  "virtual-format storage system" (VFSS) of KANGAROO
- cooperation with OS file system
Resource managers

Storage manager

- Reservation for a media data object:
  access_mode = \{create, read, update\}
  book: (OID, external_quality_descriptor, access_mode) \rightarrow reservation

- Accepting a reservation:
  accept: reservation \rightarrow reservation

- Fixing the data in buffer:
  fix: (reservation, start_unit, count) \rightarrow iterator

- Iterator:
  pause: blocking a data stream
  resume: deblocking a data stream
  bulk_copy: for copying on block-level
dereferencing
diverse pointer arithmetic
### Storage manager (2)

- **Cancel a reservation:**
  - `cancel`: reservation →

- **Remove a fixing:**
  - `unfix`: (iterator, only) →
    - only the fixing or the reservation as well

- **Change the reservation:**
  - `rebook`: (reservation, external_quality_descriptor, access_mode) → reservation

### Summary and outlook

- **Goal:** Data independence and realtime
- **Conversions**
- **Converter graphs**
- **Towards mathematical models:**
  - jitter-constrained periodic streams
  - statistical rate-monotonic scheduling
  - probabilities of conflict
- **Prototype systems KANGAROO and Memo.real**
- **Scheduling on the basis of converter descriptions**