**StreamGlobe: P2P Stream Sharing**

Data Stream Sharing in a Grid-based P2P Data Stream Management System

Richard Kunstchke
Lerstuhl Informatik III: Datenbanksysteme
Technische Universität München – Fakultät für Informatik

---

**Network Basics:**
- Grid-based P2P network
- Super-Peer Backbone
- Super-Peers: Powerful stationary servers
- Thin-Peers: Less powerful, possibly mobile peers, sensor devices, etc.

**Deficiencies of traditional approach:**
- Redundant transmission of data streams
- Redundant execution of stream transforming operators
- Transmission of unnecessary data
  - **Increased network traffic**
  - **Increased peer load**

---

**WXQuery Subscription Language**

**Example Stream DTD:**

```xml
<photons>
  <photon>
    <phc/>
  </photons>
</photons>
```

**Example Query 1:**

```xml
<photons>
  <photon>
    <phc/>
  </photons>
</photons>
```

**Example Aggregation Query:**

```xml
<photons>
  <photon>
    <phc/>
  </photons>
</photons>
```

---

**Predicate Matching and Evaluation**

**Example Predicates:**

- Stream Predicate p1:
  - \((a \geq 3) \land (a \leq 12) \land (b \geq 5) \land (b \leq 1)\)
  - \((a \geq 9) \land (a \leq 14) \land (b \geq 2) \land (b \leq 5)\)
  - \((a \geq 5) \land (a \leq 5) \land (b \geq 3) \land (b \leq 6)\)

- Query Predicate p2:
  - \((a \geq 1) \land (a \leq 8) \land (b \geq 2) \land (b \leq 4)\)

**Predicate Matching Algorithms:**
- Quick Check (QC)
- Heuristics with Simple Relaxation (HSR)
- Heuristics with Complex Relaxation (HCR)
- Exact Matching (EM)

**Predicate Evaluation Algorithms:**
- Standard Evaluation (SE)
- Index-based Evaluation (IE)

**Optimization:**
- Multi-dimensional index support (I)
- Improve performance of evaluation index through short-circuiting (SC)

---

**Representing and Merging Query and Stream Properties**

**Properties of Query 1:**
- Annotated tree
- Tree reflects structural properties
- Annotations reflect content-based properties

**Properties of Stream:**
- Spatial data structures

**Merged Properties of Queries 1 and 2:**
- Spatial data structures

**Abstract Properties Approach:**
- Treat queries and streams symmetrically
- Represent both internally using the same properties data structure
- Properties reflect relevant aspects for query optimization using in-network query processing and multi-subscription optimization

**Merging Properties:**
- Necessary for generating data streams that can be used to satisfy multiple queries
- Requires predicate matching strategy

---

**Results / Ongoing and Future Work**

Data Stream Sharing proves to be beneficial:

**Predicate Matching and Evaluation Results:**

**Ongoing Work:**
- Implementation and evaluation of an advanced Data Stream Sharing technique including Data Stream Widening/Narrowing through merging properties structures
- Hierarchical or fully distributed network architecture to address scalability
- Dynamic optimization
- Specialized multi-dimensional index structure for predicate matching and evaluation

---

**StreamGlobe Basics:**
- StreamGlobe: Distributed Data Stream Management System (DSMS)
- Super-Peer process and route data streams
- Thin-Peers publish and subscribe to data streams

**Benefits of StreamGlobe approach:**
- Stream sharing avoids redundant stream transmission
- Sharing computational results avoids redundant computation
- Early filtering and aggregation avoid unnecessary data transmission
  - **Reduced network traffic**
  - **Reduced peer load**

---

**StreamGlobe Basics:**
- StreamGlobe: Distributed Data Stream Management System (DSMS)
- Super-Peer process and route data streams
- Thin-Peers publish and subscribe to data streams

**Benefits of StreamGlobe approach:**
- Stream sharing avoids redundant stream transmission
- Sharing computational results avoids redundant computation
- Early filtering and aggregation avoid unnecessary data transmission
  - **Reduced network traffic**
  - **Reduced peer load**

---

**TXQuery Subscription Language**

**Example Stream DTD:**

```xml
<photons>
  <photon>
    <phc/>
  </photons>
</photons>
```

**Example Query 1:**

```xml
<photons>
  <photon>
    <phc/>
  </photons>
</photons>
```

**Example Aggregation Query:**

```xml
<photons>
  <photon>
    <phc/>
  </photons>
</photons>
```

---

**Predicate Matching and Evaluation**

**Example Predicates:**

- Stream Predicate p1:
  - \((a \geq 3) \land (a \leq 12) \land (b \geq 5) \land (b \leq 1)\)
  - \((a \geq 9) \land (a \leq 14) \land (b \geq 2) \land (b \leq 5)\)
  - \((a \geq 5) \land (a \leq 5) \land (b \geq 3) \land (b \leq 6)\)

- Query Predicate p2:
  - \((a \geq 1) \land (a \leq 8) \land (b \geq 2) \land (b \leq 4)\)

**Predicate Matching Algorithms:**
- Quick Check (QC)
- Heuristics with Simple Relaxation (HSR)
- Heuristics with Complex Relaxation (HCR)
- Exact Matching (EM)

**Predicate Evaluation Algorithms:**
- Standard Evaluation (SE)
- Index-based Evaluation (IE)

**Optimization:**
- Multi-dimensional index support (I)
- Improve performance of evaluation index through short-circuiting (SC)