# Data Stream Sharing

Richard Kuntschke and Alfons Kemper

Fakultät für Informatik
Technische Universität München
Germany

# Overview

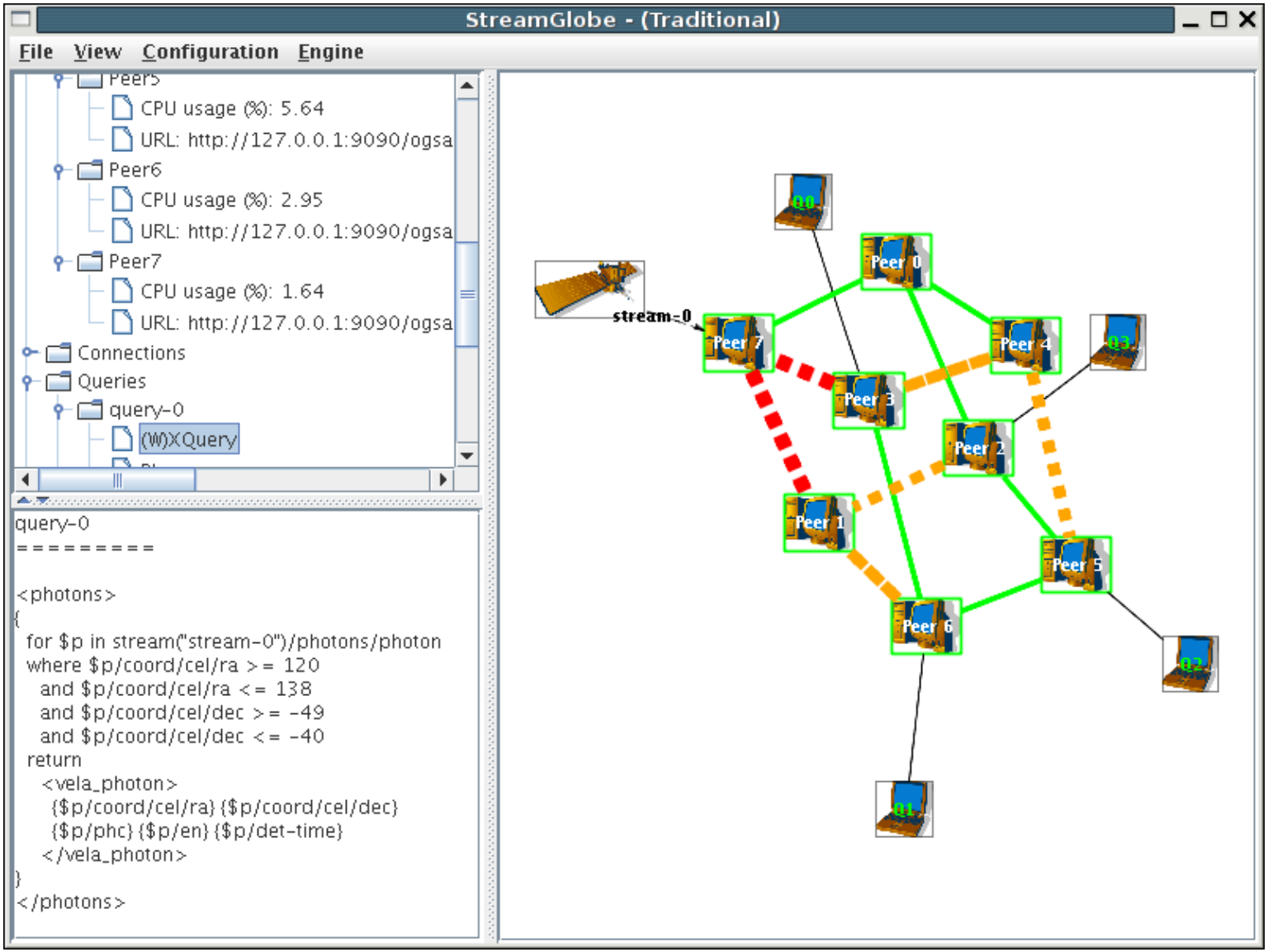
- Introduction and Motivation
- Subscription Language
- Data Stream Sharing
- Conclusion and Outlook

# Overview

- Introduction and Motivation
- Subscription Language
- Data Stream Sharing
- Conclusion and Outlook

# Preliminaries

- The *StreamGlobe* Data Stream Management System (DSMS) (VLDB 2005)
- Grid-based P2P network
  - Super-Peers
  - Thin-Peers
  - Speaker-Peer
- XML Data Streams
- (W)XQuery Subscriptions
- Query processing with *FluX* (VLDB 2004)


StreamGlobe - (Traditional)
File View Configuration Engine
Peer5
CPU usage (%): 5.64
URL: http://127.0.0.1:9090/ogsa
Peer6
CPU usage (%): 2.95
URL: http://127.0.0.1:9090/ogsa
Peer7
CPU usage (%): 1.64
URL: http://127.0.0.1:9090/ogsa
Connections
Queries
query-0
(W)XQuery
query-0
==========
<photons>
{
for $p in stream("stream-0")/photons/photon
where $p/coord/cel/ra >= 120
and $p/coord/cel/ra <= 138
and $p/coord/cel/dec >= -49
and $p/coord/cel/dec <= -40
return
<vela_photon>
{$p/coord/cel/ra}{$p/coord/cel/dec}
{$p/phc}{$p/en}{$p/det-time}
</vela_photon>
}
</photons>
stream-0
Peer 0
Peer 1
Peer 2
Peer 3
Peer 4
Peer 5
Peer 6
Peer 7
Q0
Q1
Q2
Q3

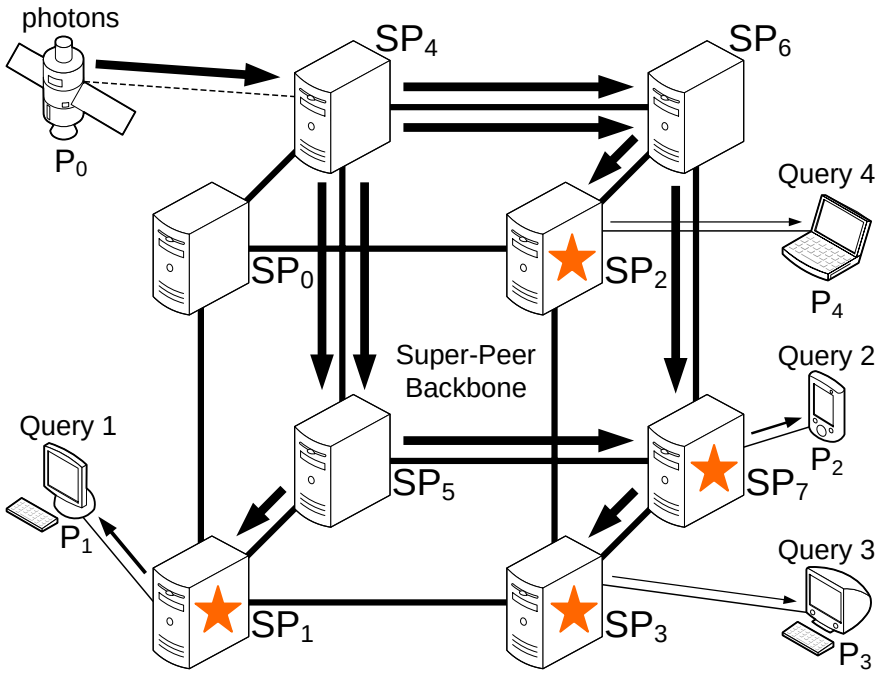
# Goals and Challenges

- Optimize incrementally registered subscriptions
- Find suitable input data in the network
- Reduce network traffic and peer load

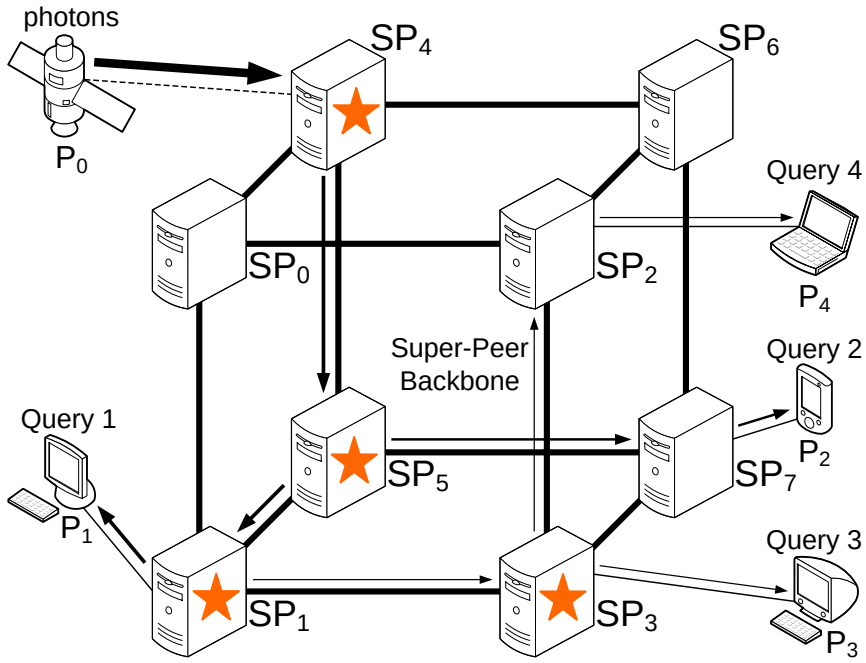
# Optimization Techniques

- Data Stream Sharing
  - In-network query processing
  - Multi-subscription optimization
- Treat queries and data streams symmetrically
- Cost-based optimizer

# Motivation




Without Data Stream Sharing

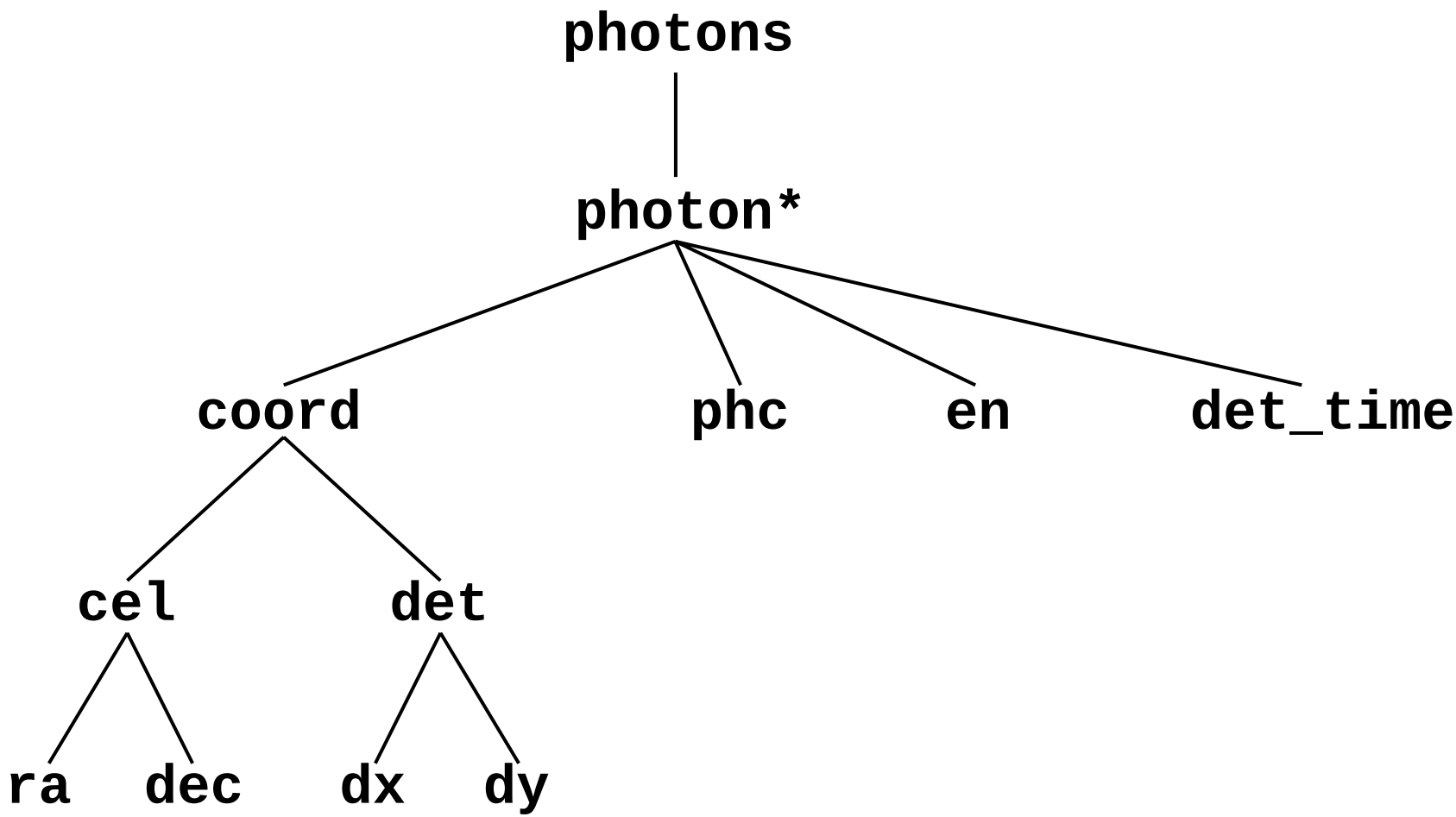



With Data Stream Sharing

# Overview

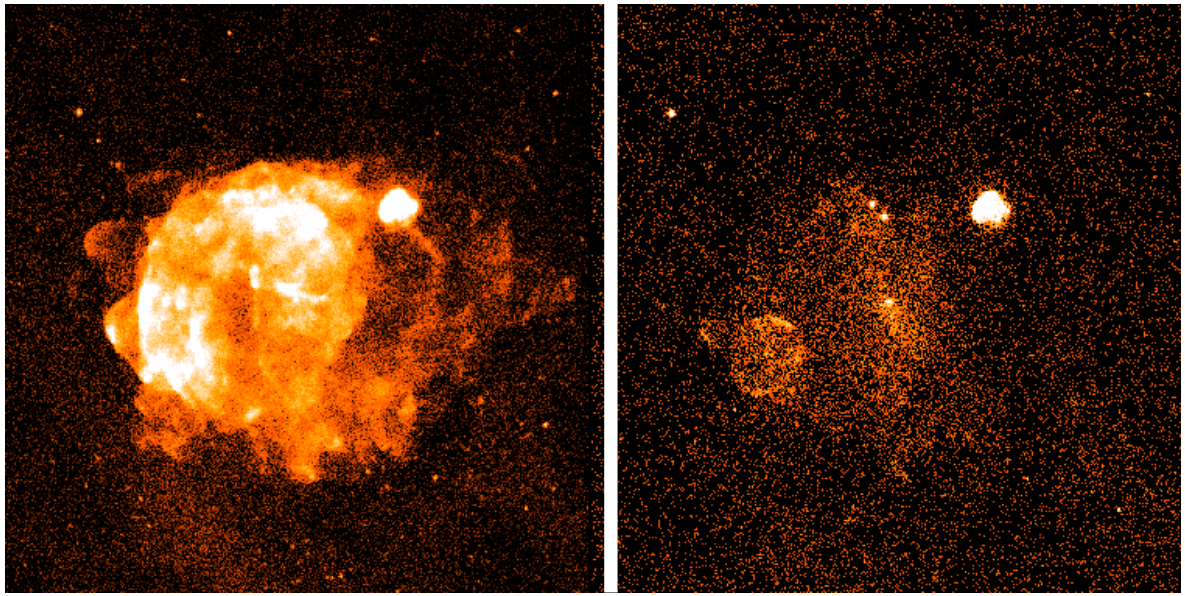
- Introduction and Motivation
- Subscription Language
- Data Stream Sharing
- Conclusion and Outlook

# Example Data Stream



# Example Data

Vela Supernova Remnant



RXJ0852.0-4622 Supernova Remnant

# WXQuery (Windowed XQuery)

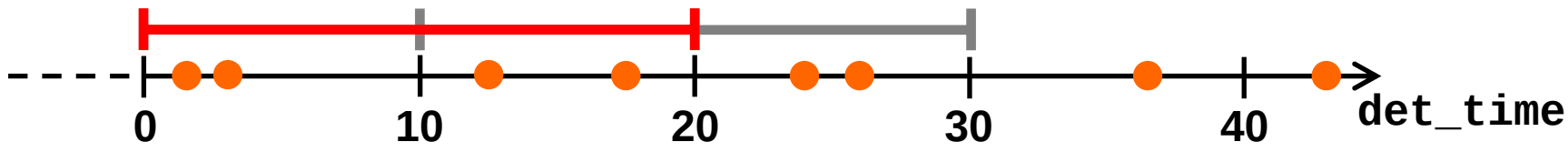
```
<photons>
  for $p in stream("photons")/photons/photon
  where $p/coord/cel/ra >= 120.0
    and $p/coord/cel/ra <= 138.0
    and $p/coord/cel/dec >= -49.0
    and $p/coord/cel/dec <= -40.0
  return
    <vela>
      {$p/coord/cel/ra} {$p/coord/cel/dec}
      {$p/phc} {$p/en} {$p/det_time}
    </vela>
</photons>
```

# WXQuery (Windowed XQuery)

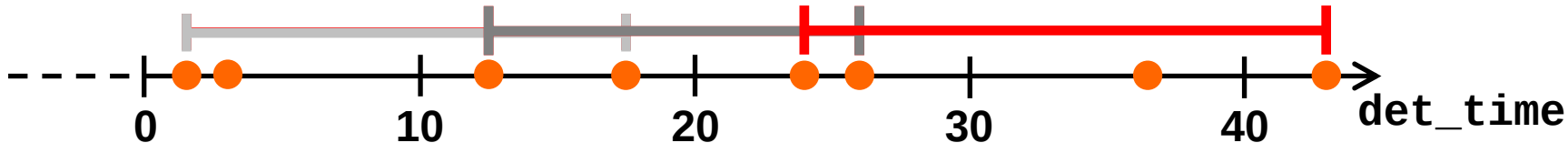
```
<photons>
  for $w in stream("photons")/photons/photon
    [coord/cel/ra >= 120.0 and
     coord/cel/ra <= 138.0 and
     coord/cel/dec >= -49.0 and
     coord/cel/dec <= -40.0]
    |/photon/det_time diff 20 step 10|
  let $a := avg($w/photon/en)
  return
    <avg_en> {$a} </avg_en>
</photons>
```

# Data Windows

**|/photon/det_time diff 20 step 10|**




**|count 4 step 2|**



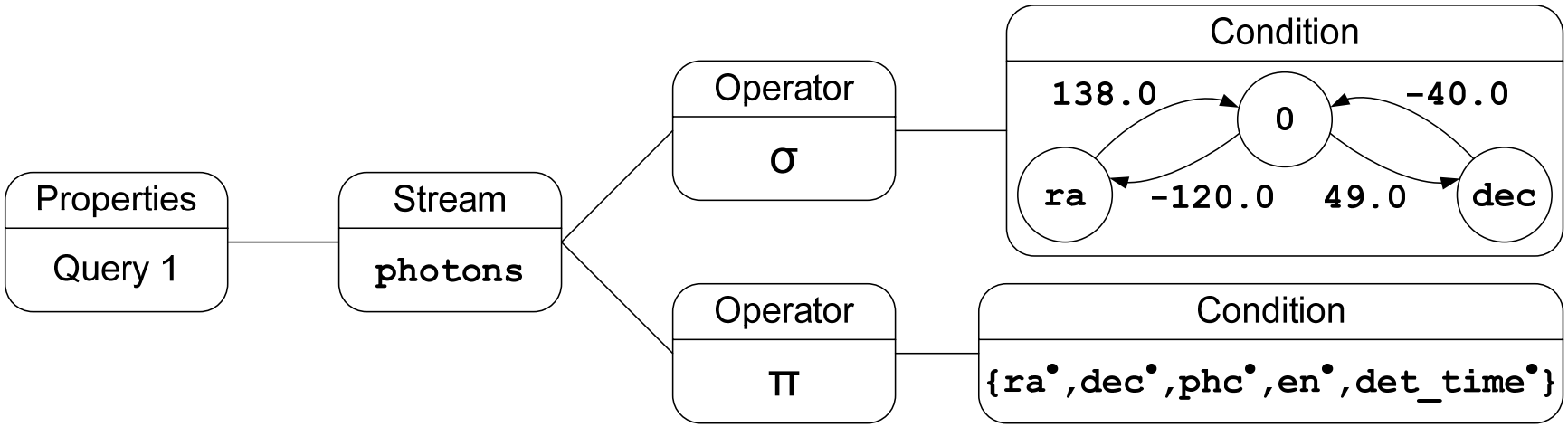
# Overview

- Introduction and Motivation
- Subscription Language
- Data Stream Sharing
- Conclusion and Outlook

# Query 1

```
<photons>
  for $p in stream("photons")/photons/photon
  where $p/coord/cel/ra >= 120.0
    and $p/coord/cel/ra <= 138.0
    and $p/coord/cel/dec >= -49.0
    and $p/coord/cel/dec <= -40.0
  return
    <vela>
      {$p/coord/cel/ra} {$p/coord/cel/dec}
      {$p/phc} {$p/en} {$p/det_time}
    </vela>
</photons>
```

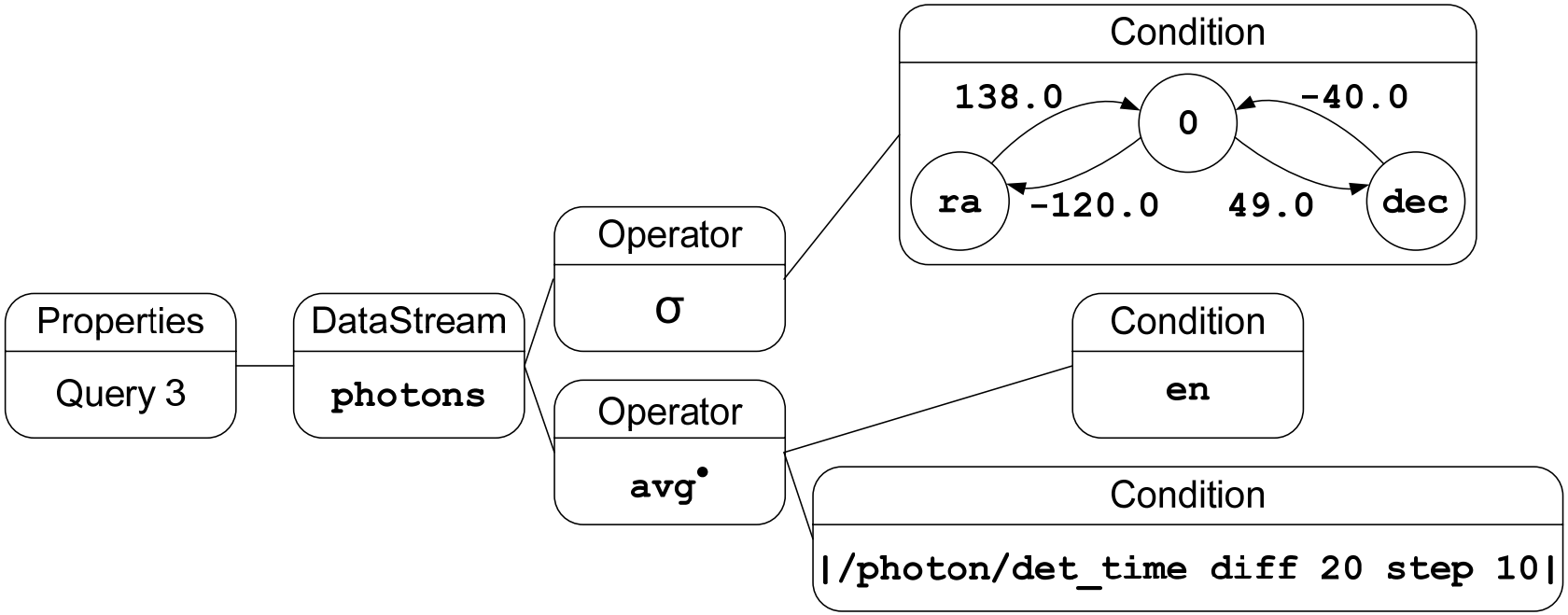
# Abstract Properties of Query 1



# Query 2

```
<photons>
  for $w in stream("photons")/photons/photon
    [coord/cel/ra >= 120.0 and
     coord/cel/ra <= 138.0 and
     coord/cel/dec >= -49.0 and
     coord/cel/dec <= -40.0]
    |/photon/det_time diff 20 step 10|
  let $a := avg($w/photon/en)
  return
    <avg_en> {$a} </avg_en>
</photons>
```

# Abstract Properties of Query 2

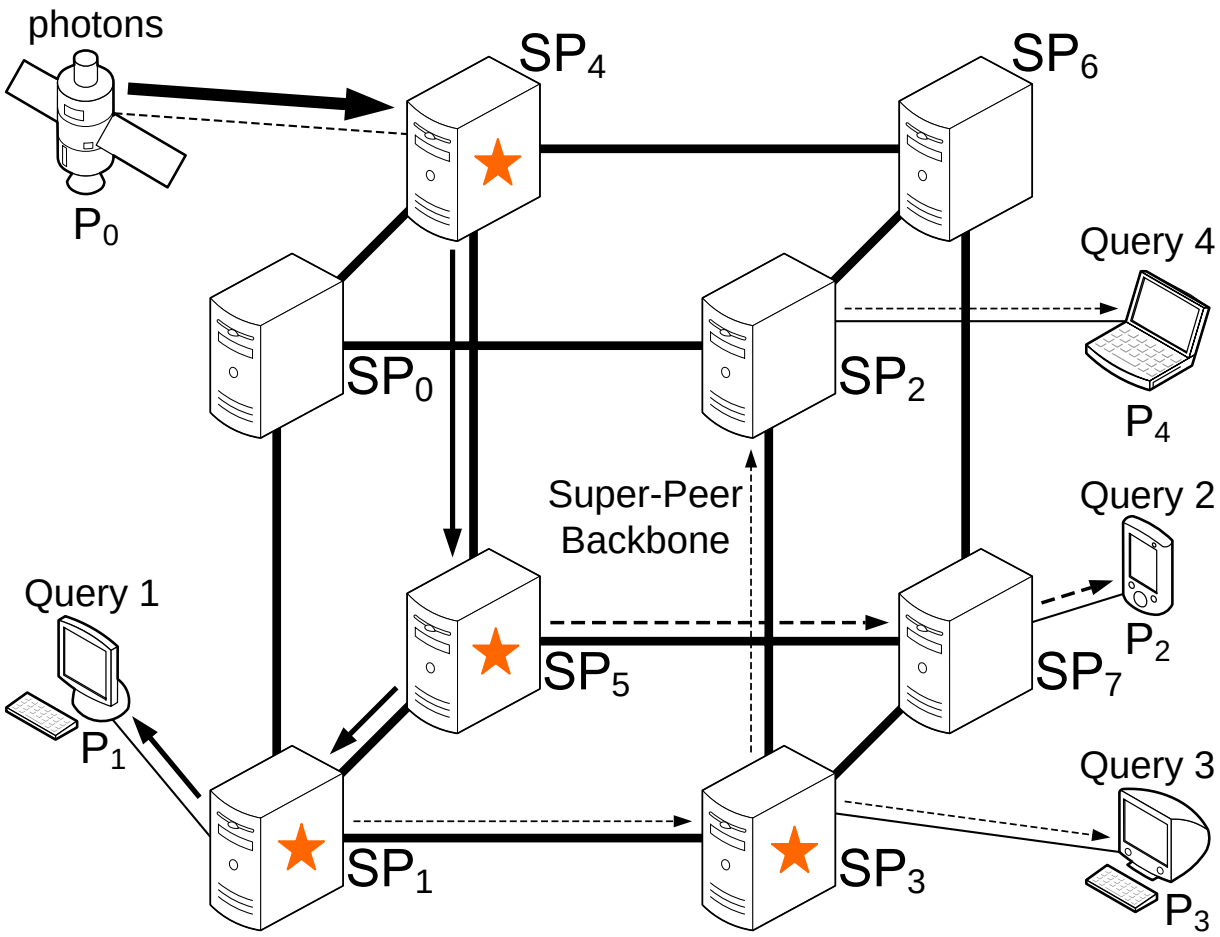


# Data Stream Discovery and Cost Model

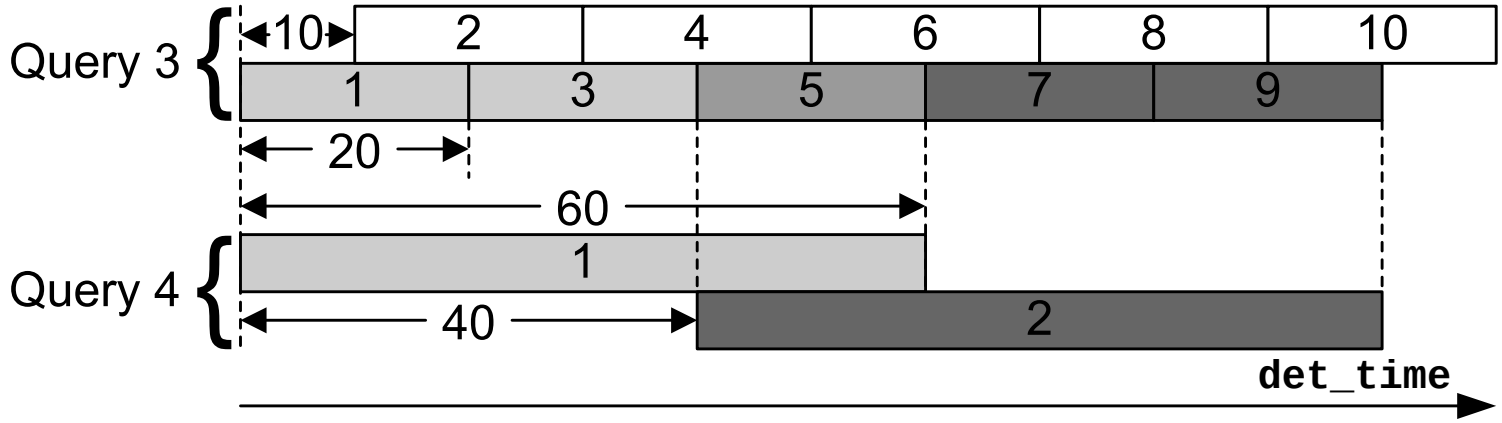
- Data Stream Discovery
  - Start at origin of referenced stream
  - Search forward (BFS or DFS) in the network graph
  - Pruning

- Cost Model
  - Parameters
    - Network traffic
    - Computational load on peers

# Data Stream Discovery Example



# Window-based Aggregation




Query 3: **|/photon/det_time diff 20 step 10|**

Query 4: **|/photon/det_time diff 60 step 40|**

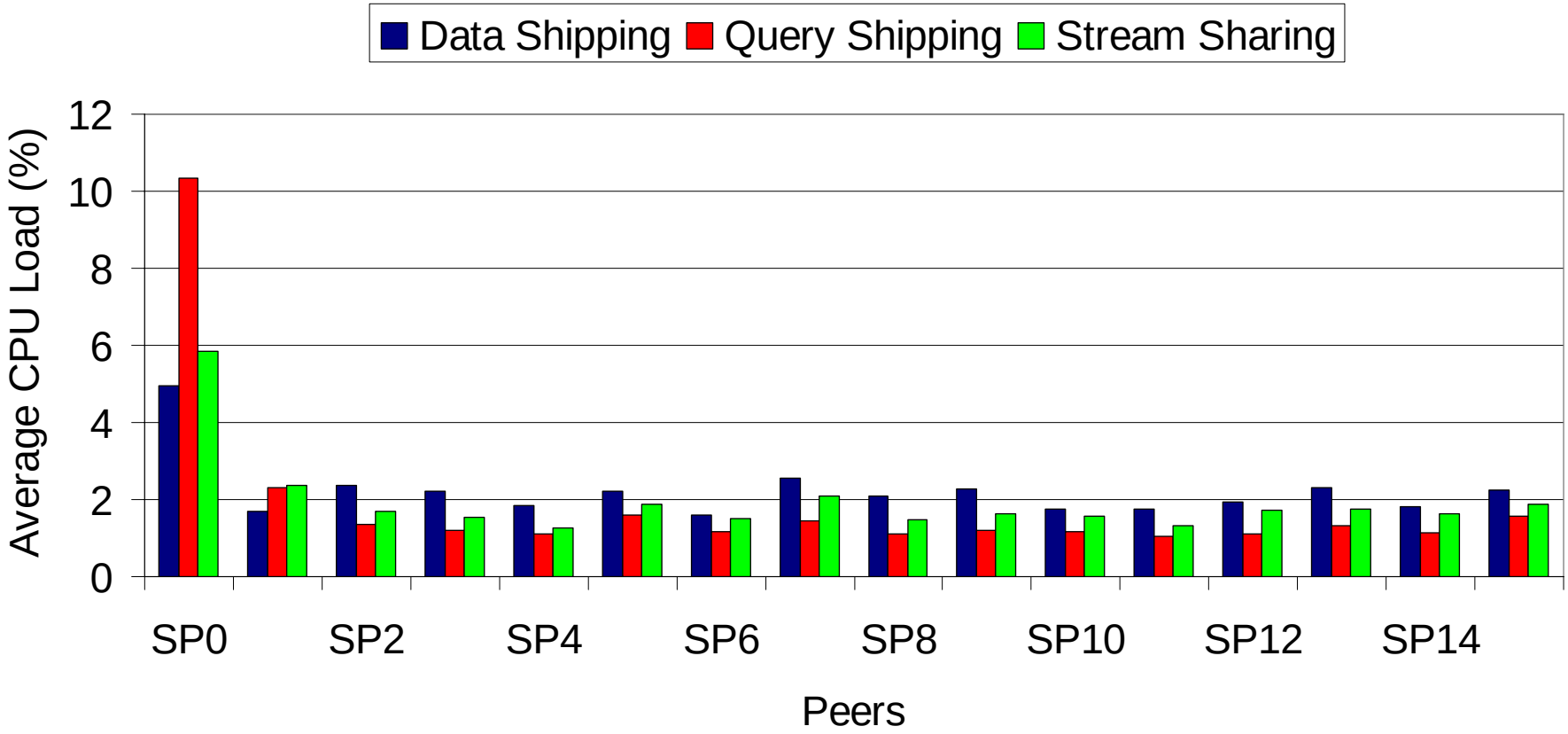
- **60 mod 20 = 0**
- **40 mod 10 = 0**
- **20 mod 10 = 0**

- **60 div 20 = 3**
- **40 div 10 = 4**
- **20 div 10 = 2**

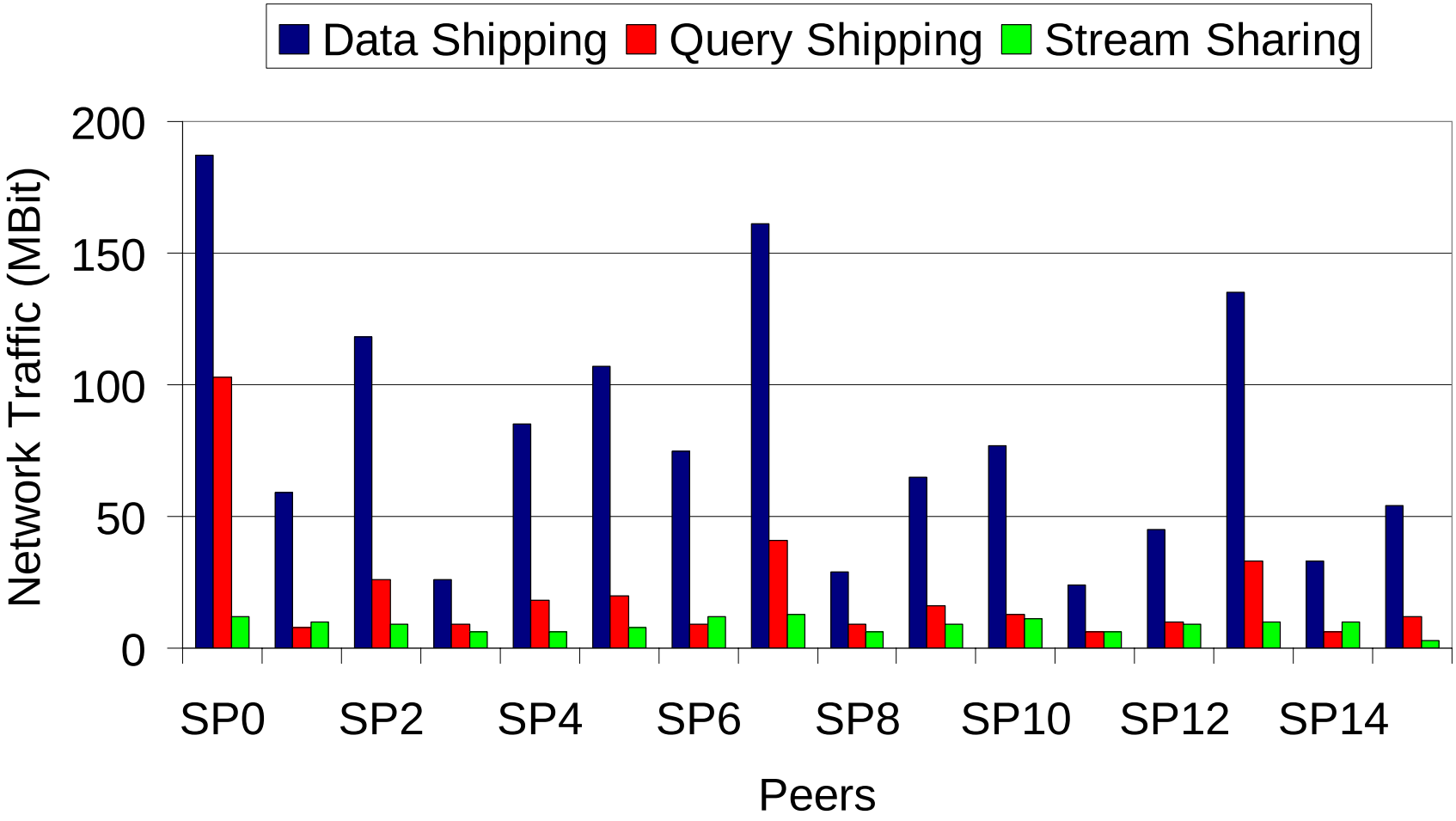
# Performance Evaluation – Preliminaries

- 4 x 4 Grid Topology
- 16 Super-Peers
- 2 Data Streams
  - Real astrophysical data
  - **photons** data streams
- 100 Queries
  - Randomly generated
  - Query Templates for Selection/Projection/Aggregation queries
  - Constant values for selection predicates and data window definitions randomly chosen from predefined set

# Performance Evaluation – Peer Load



# Performance Evaluation – Network Traffic



# Overview

- Introduction and Motivation
- Subscription Language
- Data Stream Sharing
- Conclusion and Outlook

# Conclusion

- What has been presented:
  - Subscription language
  - Properties approach
  - Cost model
  - Algorithms for data stream sharing
- Data Stream Sharing takes three steps:
  - Properties construction
  - Identification of shareable streams through properties matching
  - Plan generation, installation, and execution

# Outlook

- Advanced Data Stream Sharing
  - Improved properties structure
  - Support for nested queries
  - Data stream widening
- Dynamic optimizer
- Scalability
  - Hierarchical network organization
  - Fully distributed network organization