1. Motivation

Information most valuable when **new**.
- Batch is not enough.
- Need to handle big **streaming** data
- Need **low-latency** access to **fresh** results, with **low resource utilization**

Motivating example – CDN:
- 1000s of edge servers
- Emit log data
- Interested in **aggregates** (e.g., count, sum, min, max)

**Data Cube**
- Tuples: \((d_1, d_2, ..., d_p, m_1, m_2, ..., m_n)\)
- \(k \) **dimensions**: e.g., content provider, country
- \(n \) **measures**: e.g., # of bytes downloaded
- \(2^k \) **cuboids**: aggregates—e.g., sum, max—over measures, grouped by subset dimensions

![Diagram of a data cube](Image)

2. Problem

Pre-computing cuboids reduces query latency, **but** computing all cuboids is infeasible
- Not all \(2^k\) will be queried
- Pre-computed cuboids aid other queries

Stream sources & compute nodes geographically distributed
- Wide-area network (WAN) bandwidth limited, costly
- Must **reduce WAN utilization**

**Core questions:**
1) Which cuboids should be **eagerly** pre-computed? Which **lazily** at query-time?
2) Where should cuboids be computed & stored?

**Metrics:**
- **Utilization**: load on platform; e.g. WAN traffic
- **Latency**: time from submitting query to retrieving results
- **Staleness**: age of query results once they are received

3. Tradeoffs

**Staleness-Utilization:**
- Less staleness of results \(\rightarrow\) less reuse \(\rightarrow\) greater utilization

**Staleness-Latency:**
- Less staleness \(\rightarrow\) less reuse \(\rightarrow\) more queries for not-yet-computed results

![Diagram of staleness-utilization and staleness-latency tradeoffs](Image)

4. Approach

**Modeling**
- Model **utilization**, **latency**, and **staleness** as functions of
  - cuboid selection and placement
  - platform, data, and workload
- Develop **heuristics** for selection and placement
- Cost/benefit trade-off: pre-computing speeds queries, but increases utilization
- **New here: Consider cost of WAN links, placement**

**System Implementation**
- Global optimization plus local dynamic optimization
- Storm as communication substrate

**Experiments**
- Inputs: CDN log data
- Deployment: globally distributed (e.g., PlanetLab)

**Approach:**
- Use modeling to devise optimization heuristics that inform the design of a real system. Demonstrate system performance on real-world inputs on a globally distributed testbed.