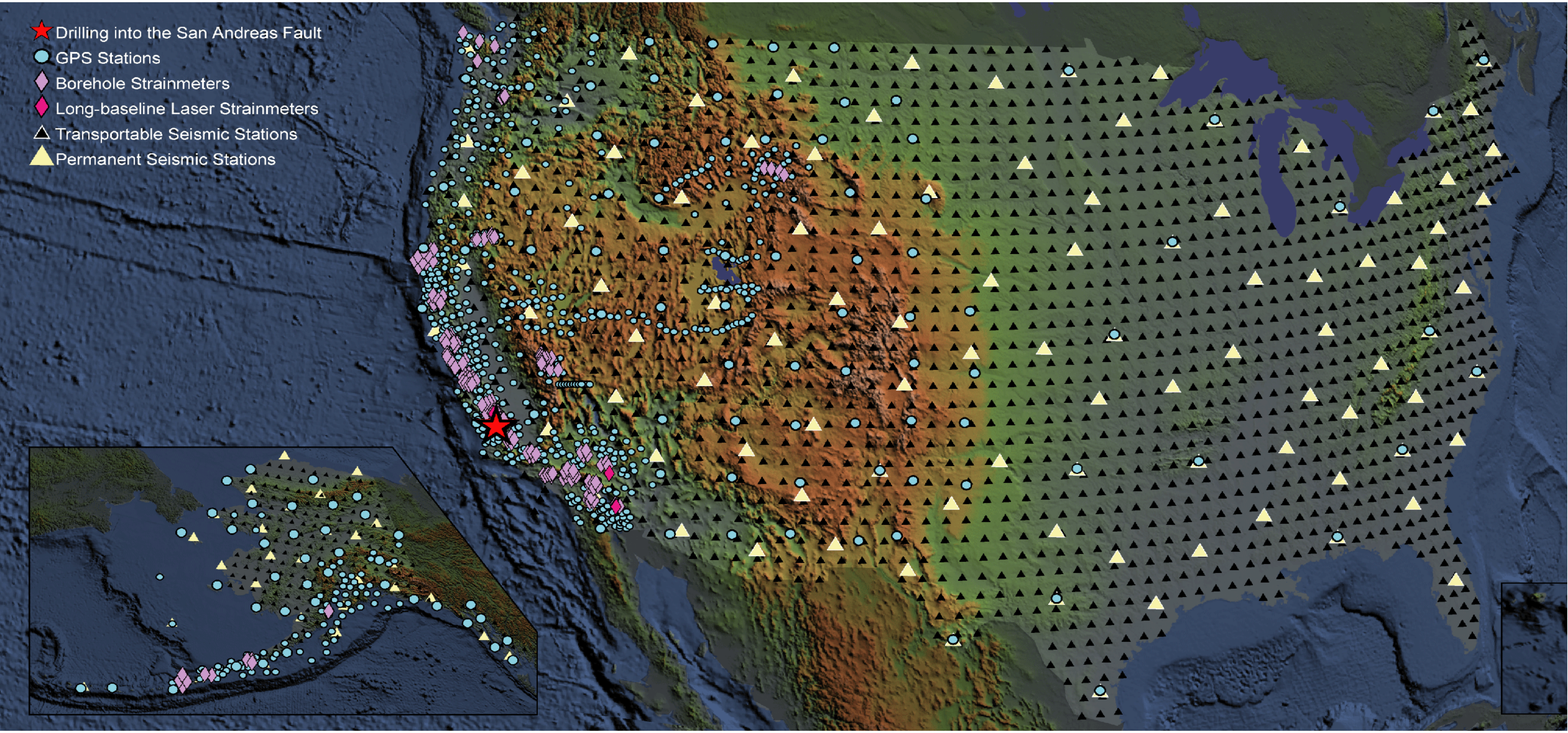
A Large-Scale Overlay Infrastructure for Streaming Real-Time Data

Peter Pietzuch
prp@eecs.harvard.edu

Systems Research Group – Harvard University
Division of Engineering and Applied Sciences

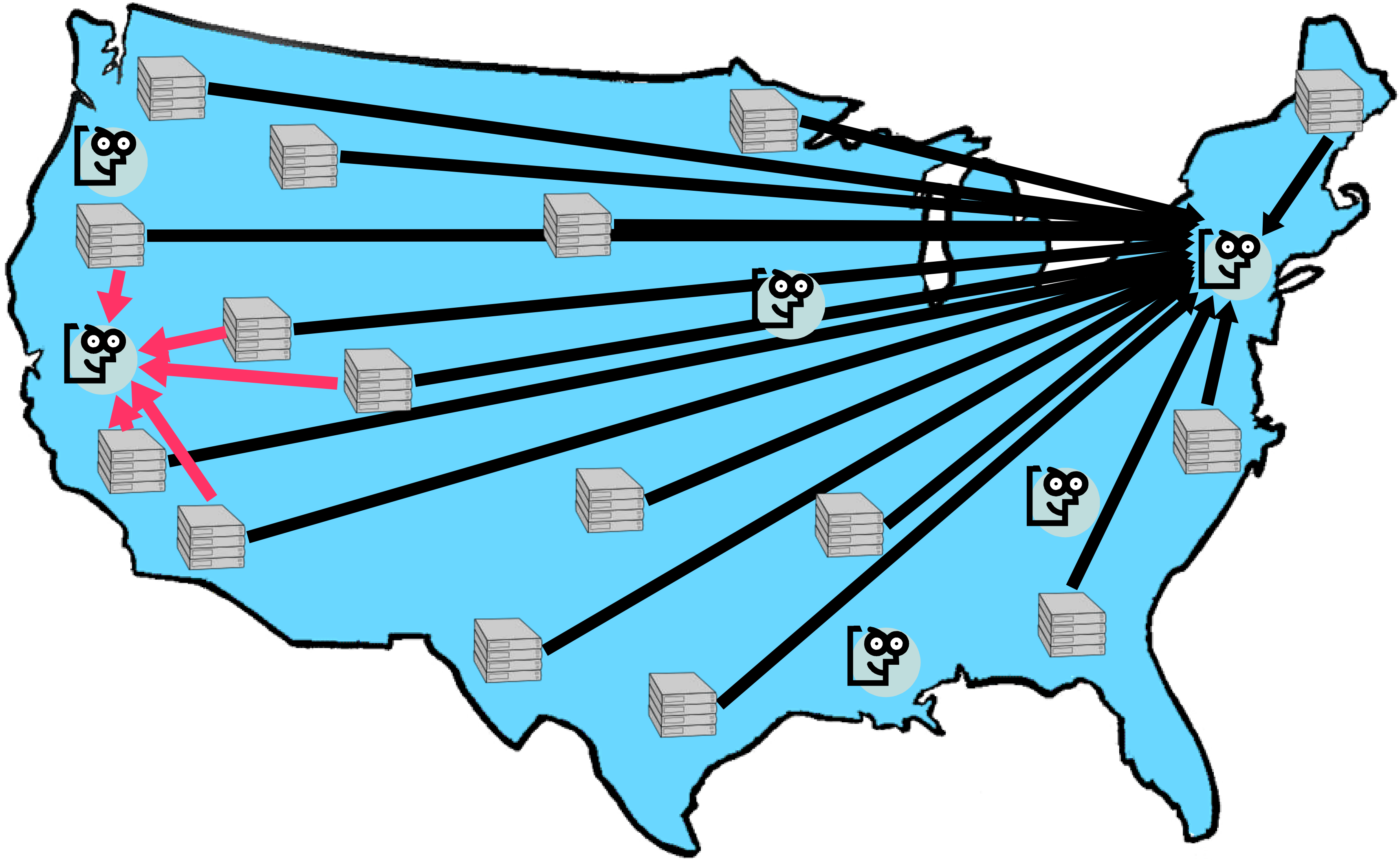
Cambridge Systems Colloquium - October 2005

Motivation: Internet-Scale SensorNets



- EarthScope: Instrument the continent to understand geological evolution
  - 400 seismometers, 1000 GPS stations, 180 strainmeters
  - **How are we going to harness this real-time data?**

Motivation: Network Monitoring



- Instrument routers to receive flow information
  - Many different queries by researchers, network admins, ...
  - **How can we support many different applications?**

Research Challenges

- Scalability
  - Large number of data sources and consumers
  - Large volume of data (sensors, RFID tags, telescopes, ...)

In-network processing

- Performance
  - Real-time stream data
  - Network and node resources are limited
  - Network and node conditions change over time

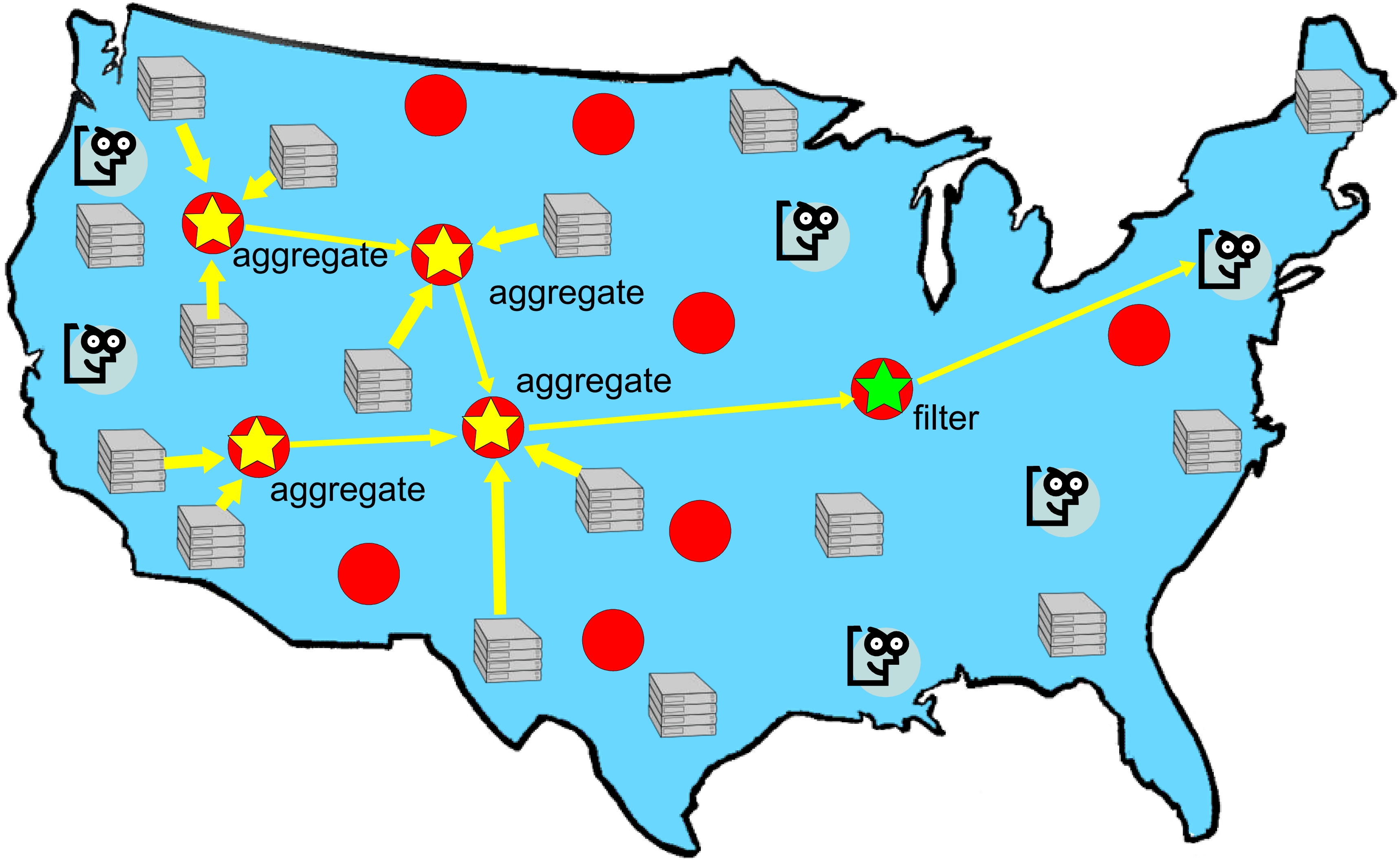
Optimization and adaptation

- Heterogeneity
  - Wide range of different applications
  - No single data model (relational, XML, VOTable, ...)
  - No fixed set of processing operators

Application independence

☞ **New Infrastructure for Building Large-Scale Stream-Processing Applications**

Stream-Based Overlay Network (SBON)

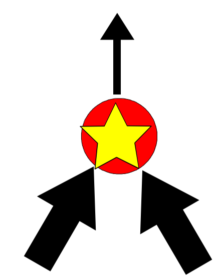


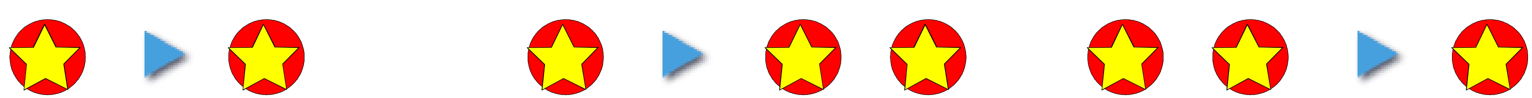
- Overlay network that processes streams on behalf of clients
  - Massive number of data sources and queries
  - Where do we locate the operators?

SBON Model

- Stream and Node Management
  - Instantiation of stream data paths and operators
  - Management of resources for in-network processing
  - Stream Optimization

- Operator Model
  - SBON is data and operator model agnostic
  - Processing operators are application-defined
    - e.g. *aggregate, join, filter-XML, match-face, adjust-parallax, ...*
  - Describe abstract operator properties
    - Measure incoming/outgoing data rates to estimate **selectivity**
    - Functions to
      - **migrate** - **decompose** - **reuse**





Distributed Stream Optimization

- Classic DB query optimization doesn't work in this context
  - Assume knowledge of operator semantics
  - Smaller scale: 100s of processing nodes and 1000s of streams
  - Global stable view of the entire system
  - Network properties not taken into account
    - latency, bandwidth, packet loss, ...
- Need novel approach for distributed stream optimization

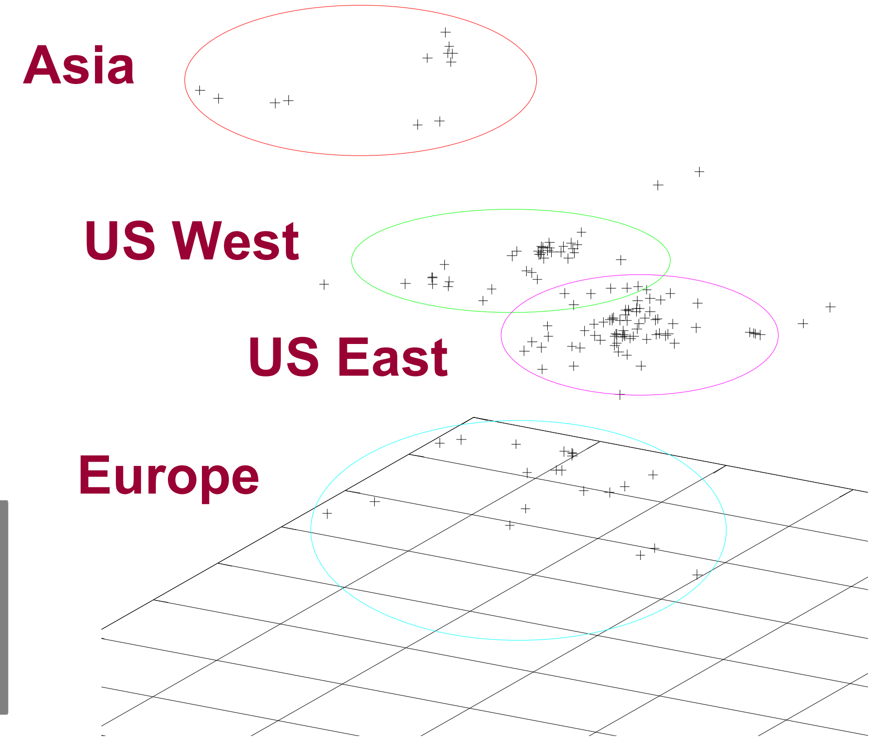
  ☞ Our approach: Perform stream optimization decisions in a virtual metric space

- Optimization metric
  - Reduce *latency* and minimize *network* effect on others
    - Push aggregation operators close to data sources
  - Minimize the amount of *in-flight data*
    - Product of ***latency*** and ***datarate***

Cost Space

- Encodes the cost of stream routing using *network coordinates*
  - Euclidean **distance ≈ latency**
    - Latency is proportional to cost
  - Distributed implementation
    - *Vivaldi*, *Lighthouses*, ...

1. Compute optimal query in cost space
2. Map to physical overlay nodes



Network coordinates on PlanetLab

  - Nearest neighbor lookup
    - e.g. *geometric routing*, *DHT*, ...

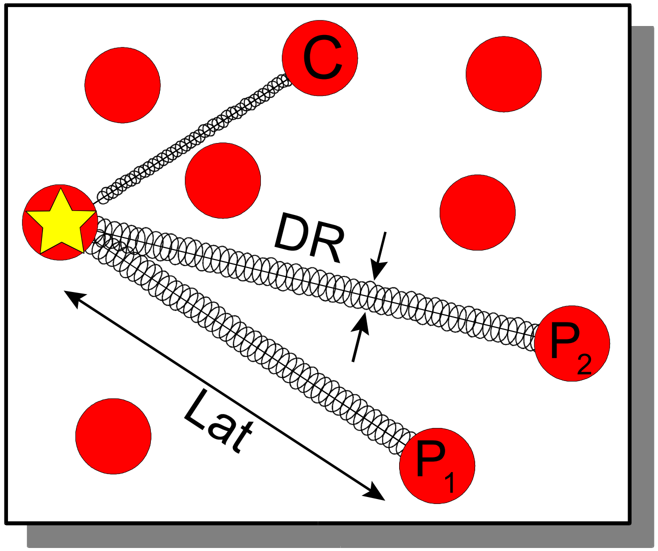
- Advantages
  - Decentralized and scalable implemention
  - Adapts to changing network conditions
  - Geometric algorithms applicable for optimization decisions

Operator Placement

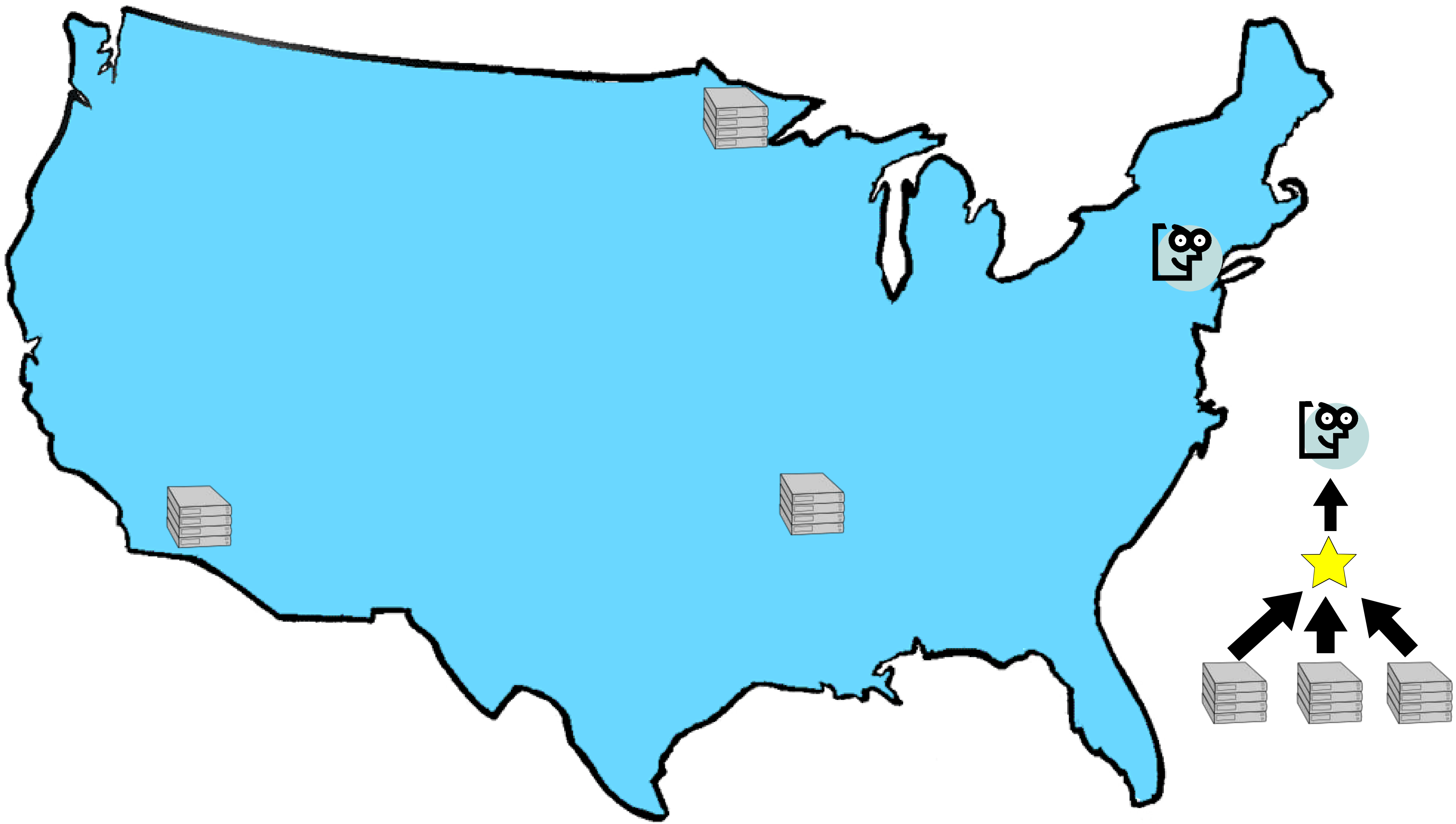
- Placement Problem
  - Different operator placements have different costs
  - Approximate optimization problem in cost space
  - Map solution back to physical node to host operator

$\sum$ Lat • DR

- Relaxation Placement
  - Physical simulation: model streams in cost space as a *network of springs*
    - Spring ***extension*** = ***latency***
    - Spring ***constant*** = ***datarate***
      - Springs "pull" according to datarate

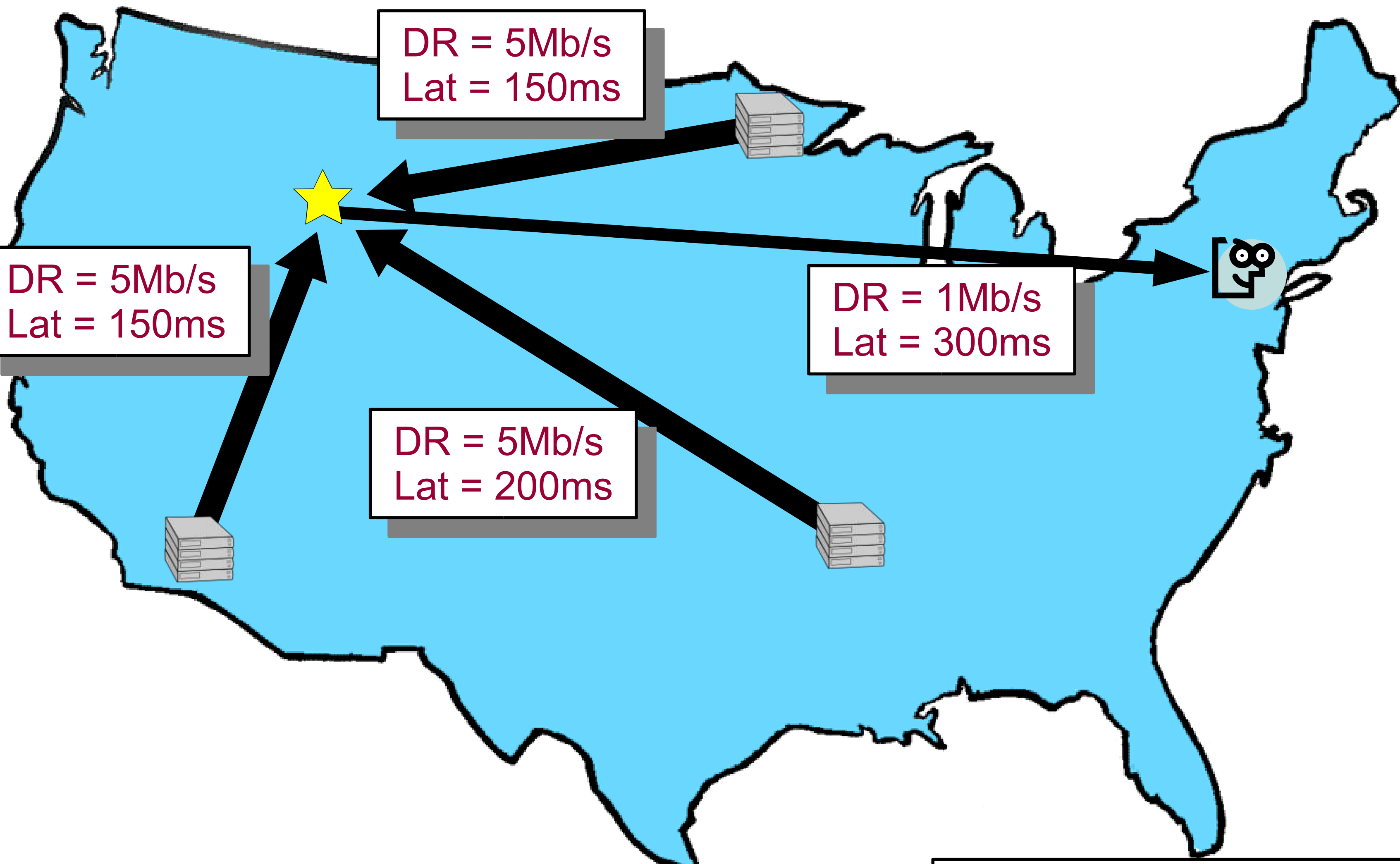


Relaxation Placement



- Minimize ***latency-datarate*** product
  - Decentralized and adaptive computation

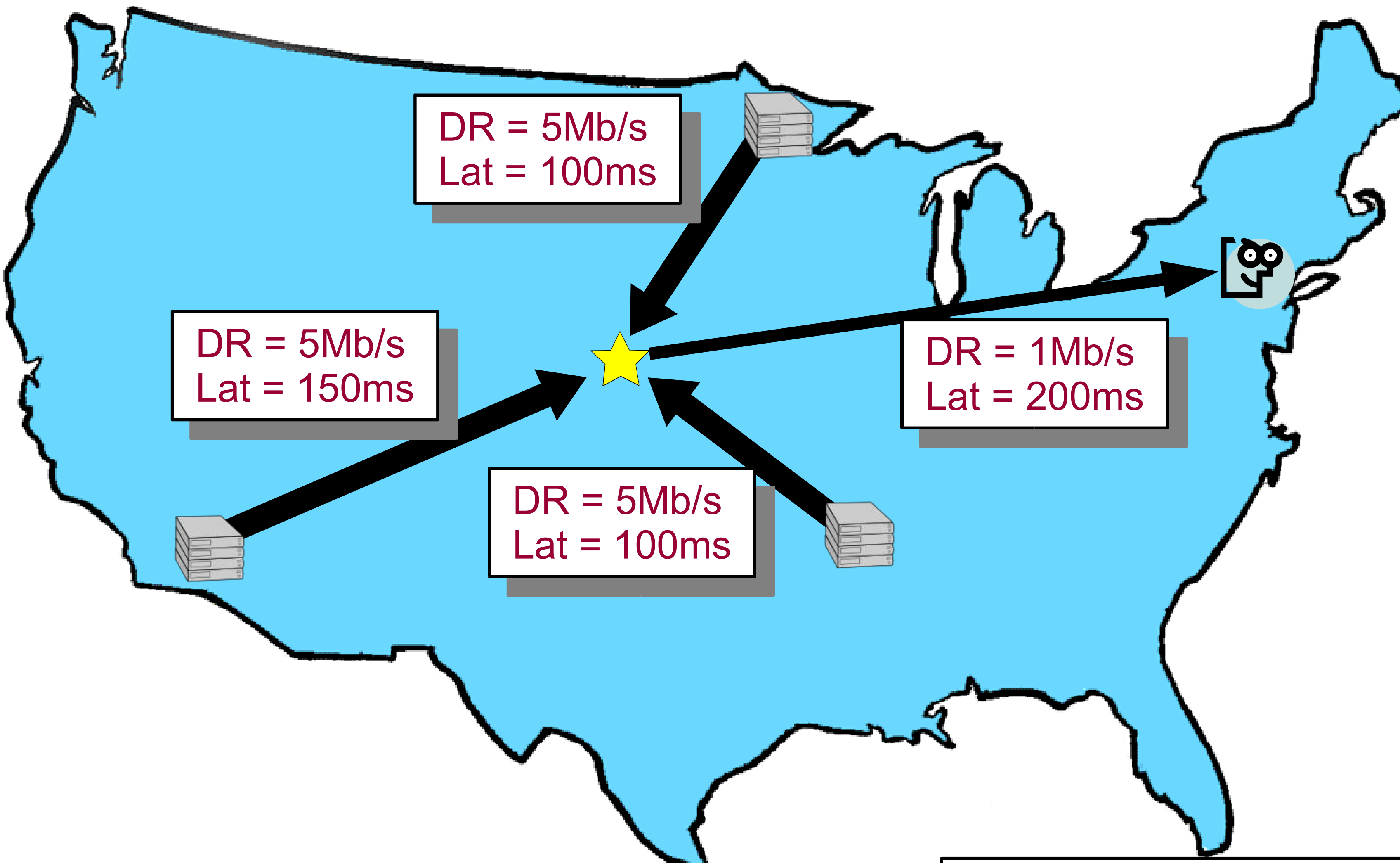
Relaxation Placement



$\sum$ Lat • DR = 2800Mb

- Minimize ***latency-datarate*** product
  - Decentralized and adaptive computation

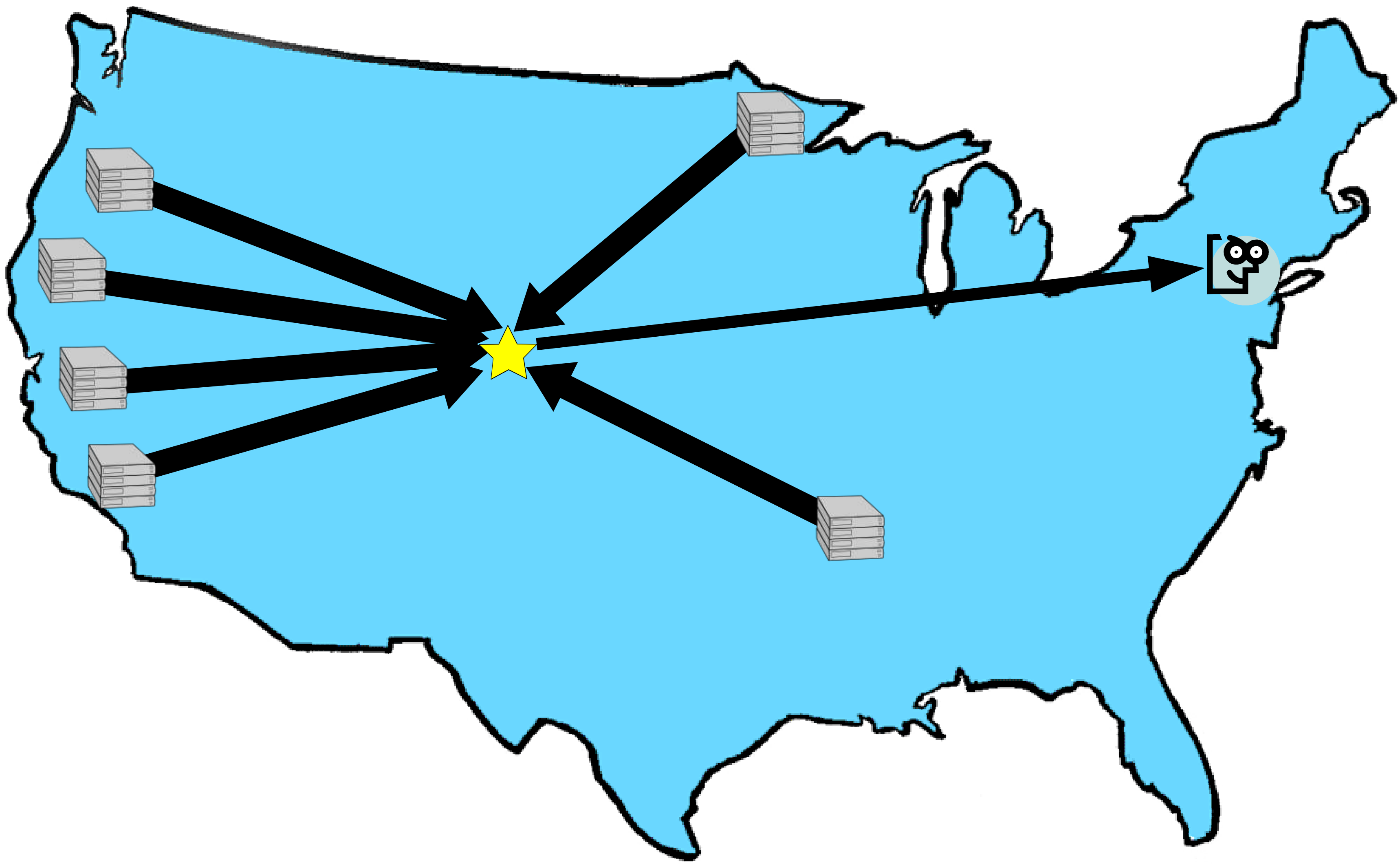
Relaxation Placement



$\sum$ Lat • DR = 1950Mb

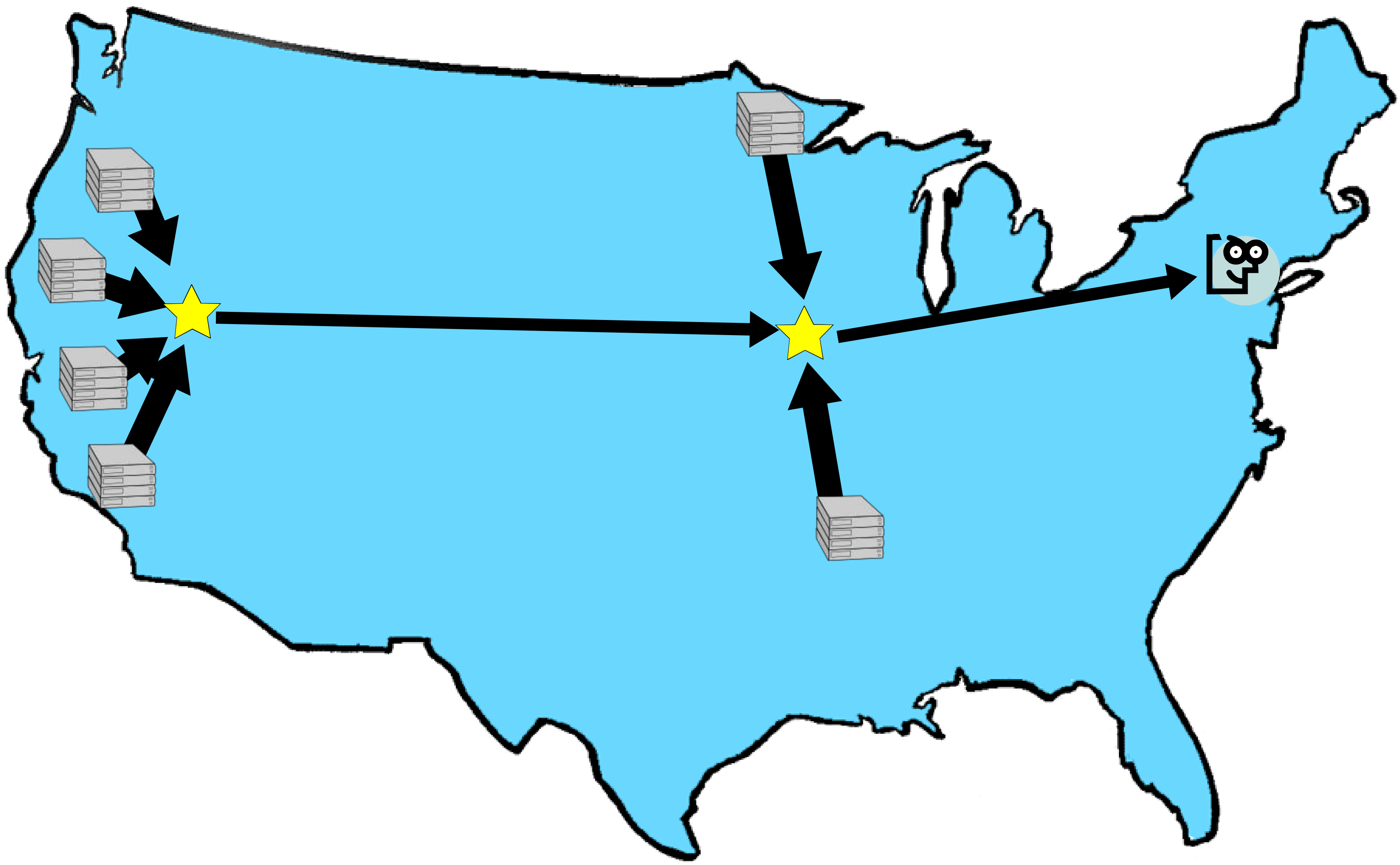
- Minimize ***latency-datarate*** product
  - Decentralized and adaptive computation

Operator Decomposition



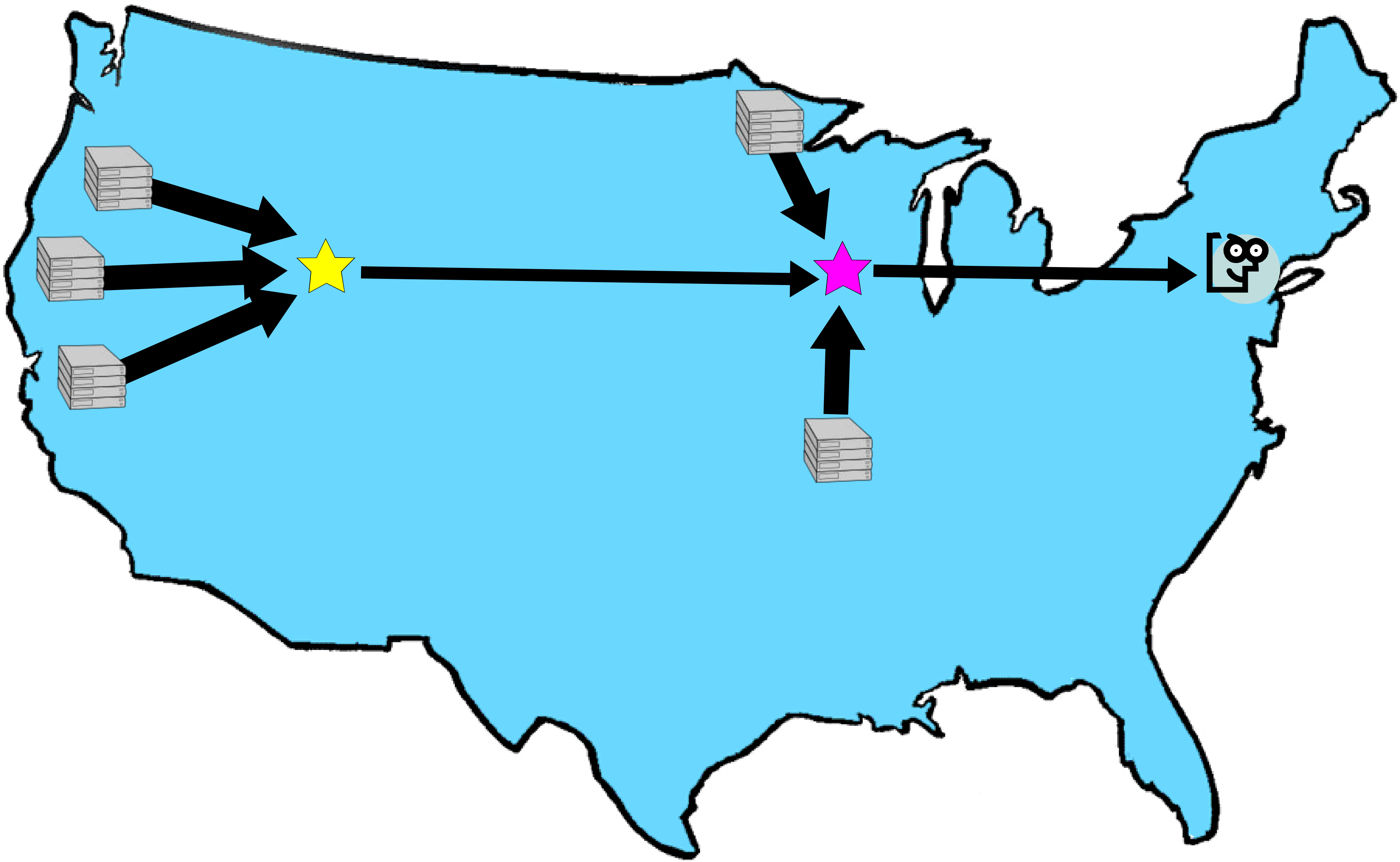
- Decompose operators due to network and CPU load
  - Consider springs pulling in given direction

Operator Decomposition



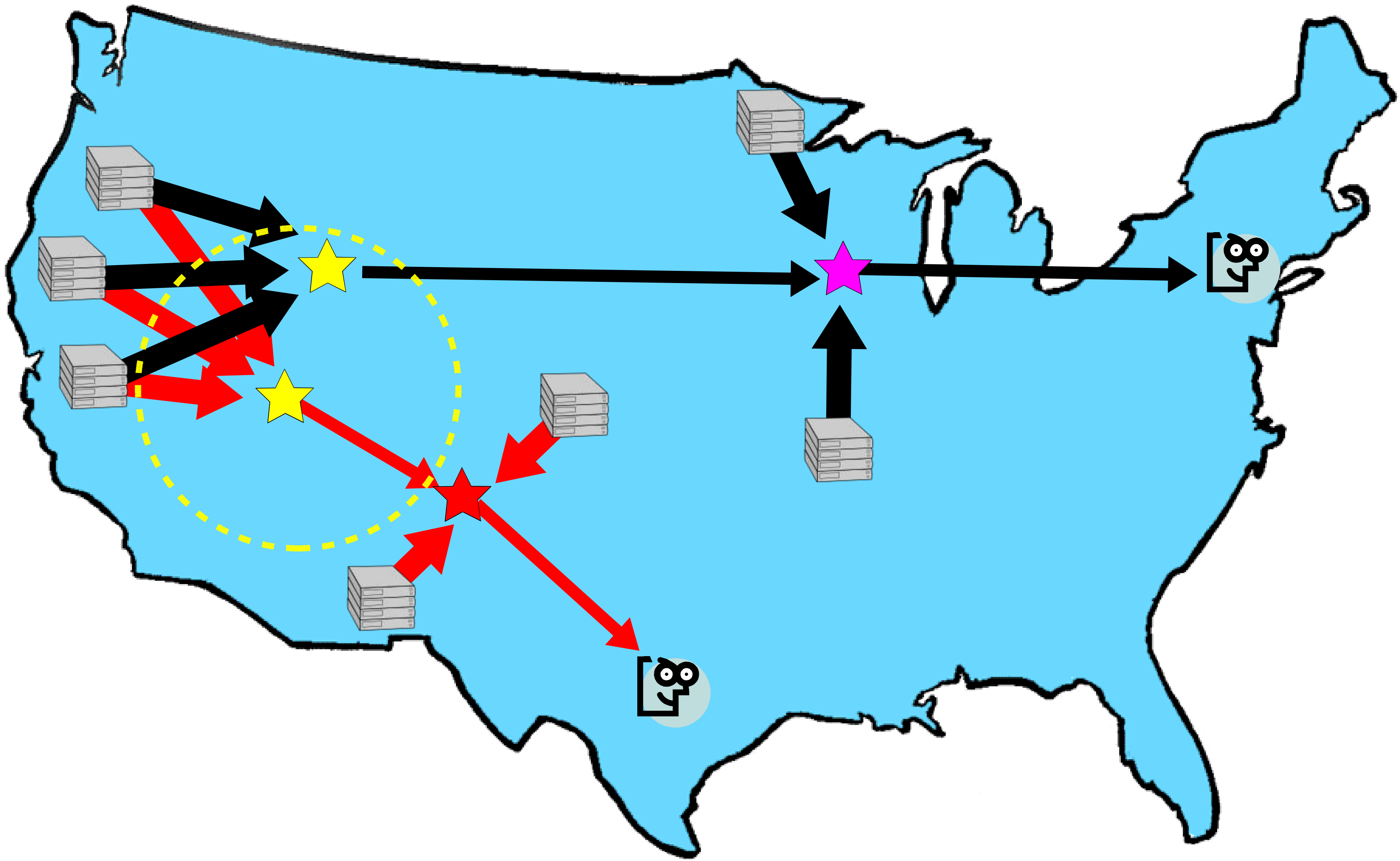
- Decompose operators due to high network and CPU load
  - Consider springs pulling in given direction

Operator Reuse



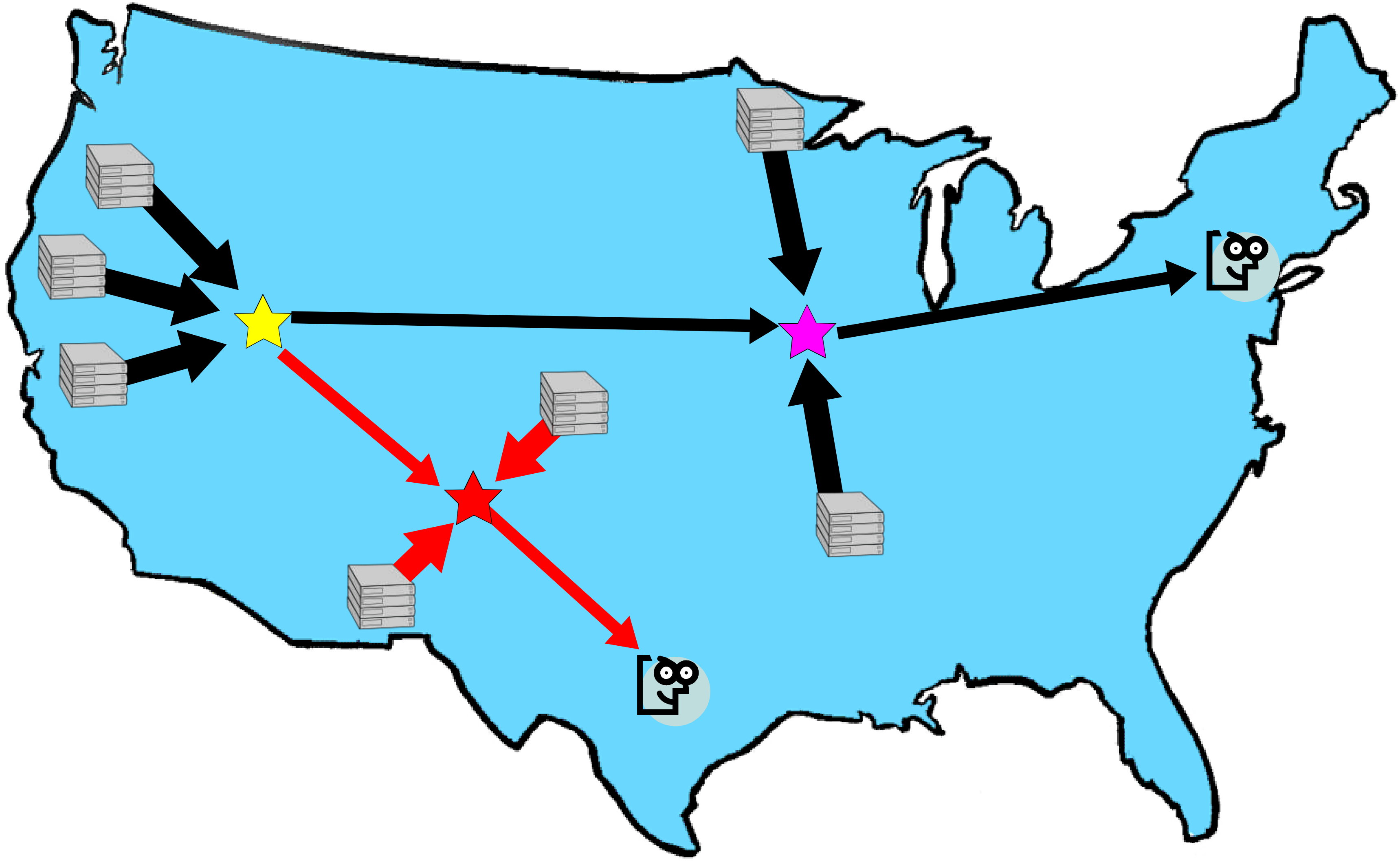
- Exploit commonality between queries
  - Use cost space to restrict search for reusable operators

Cross-Query Optimization



- Exploit commonality between queries
  - Use cost space to restrict search for reusable operators

Cross-Query Optimization



- Exploit commonality between queries
  - Use cost space to restrict search for reusable operators

Research Agenda

- Distributed Stream Optimization
  - Right set of optimization primitives
  - Take advantage of semantic knowledge
- Query Interface
  - Rich expressive query language
  - Implementation language for operators
- Resource Discovery
  - Efficient nearest neighbour search in cost space
  - Discover sensor networks
- Build and deploy real applications
  - Analysis of political weblogs
  - Detection of network attacks with PlanetFlow traffic data
  - Exploring collaborations with domain scientists

Summary

- Large-scale stream applications need new infrastructures
  - Support for in-network stream processing
  - Adaptation to network and node dynamics
- Stream-Based Overlay Network
  - Overlay infrastructure for multiple stream-processing applications
  - Data and operator model agnostic
  - Efficient placement of in-network processing operators
- Distributed Stream Optimization
  - Need new query optimization techniques for this space
  - **Cost Space** encodes network state efficiently
  - Algorithms for **placement**, **decomposition**, and **reuse**
  - SBON nodes periodically re-optimize hosted operators

Thanks!

Peter Pietzuch

http://www.eecs.harvard.edu/~prp

prp@eecs.harvard.edu

The Hourglass Project Team

Jonathan Ledlie, Jeff Shneidman, Rohan Murty,
Matt Welsh, Mema Roussopoulos, Margo Seltzer