# Global Scientific Data Infrastructures: The Big Data Challenges

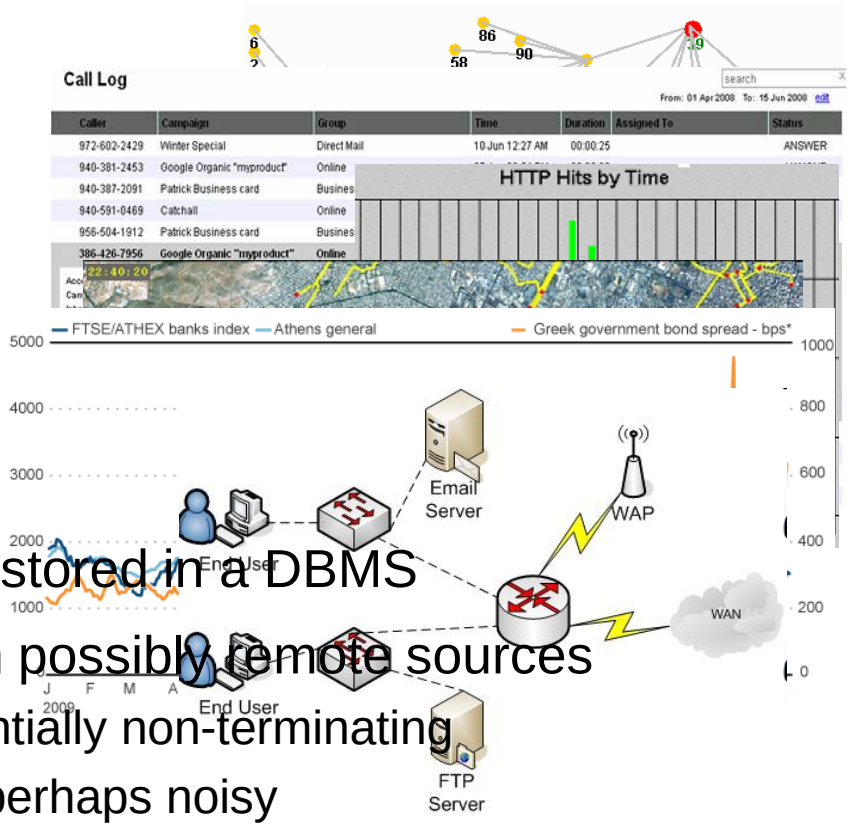
## Managing Streaming Spatial Data

Timos Sellis
timos@imis.athena-innovation.gr

Institute for the Management of Information Systems
Research Center “Athena”

# Streaming Information

- Data streams are almost ubiquitous
  - Giga- *or* Terabytes collected daily for many modern applications:
    - sensor networks
    - phone call logs
    - web logs and clickstreams
    - traffic surveillance
    - financial tickers
    - network security …
- Distinctive features
  - *not* a finite dataset persistently stored in a DBMS
  - but *unbounded* data items from possibly remote sources
    - continuously arriving and potentially non-terminating
    - rapid, transient, time-varying, perhaps noisy
    - distributed, pervasive, transmitted through networks



# Continuous Queries

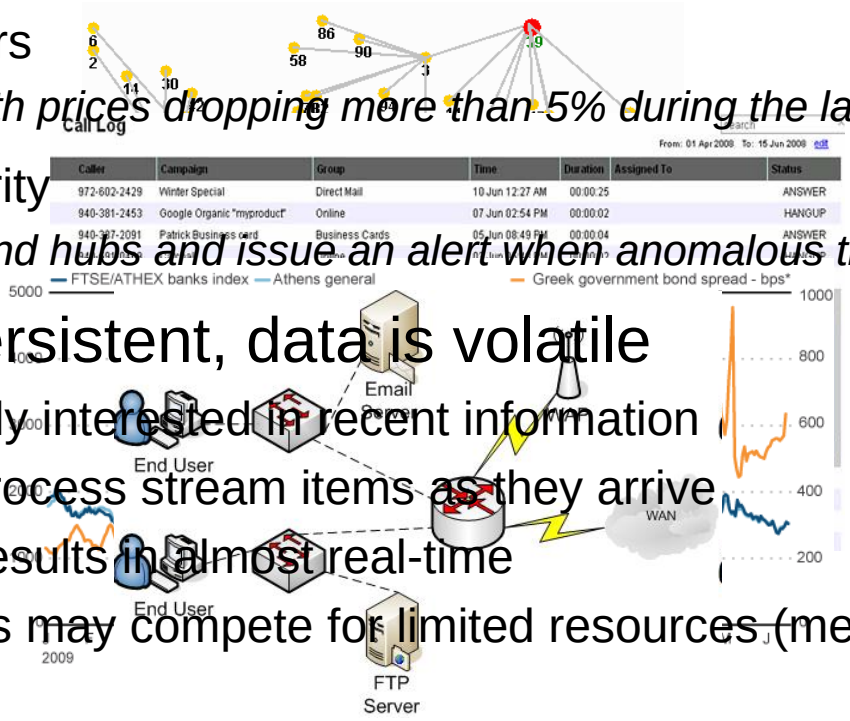
- In a streaming context, user requests remain active for long
  - Example CQs:
    - sensor networks

      *"Every 5 min report average temperature from readings over past hour"*
    - phone call logs

      *"What are the 10 most frequent pairs <caller, callee> over the past week?"*
    - financial tickers

      *"Identify stocks with prices dropping more than 5% during the last 10 minutes"*
    - network security

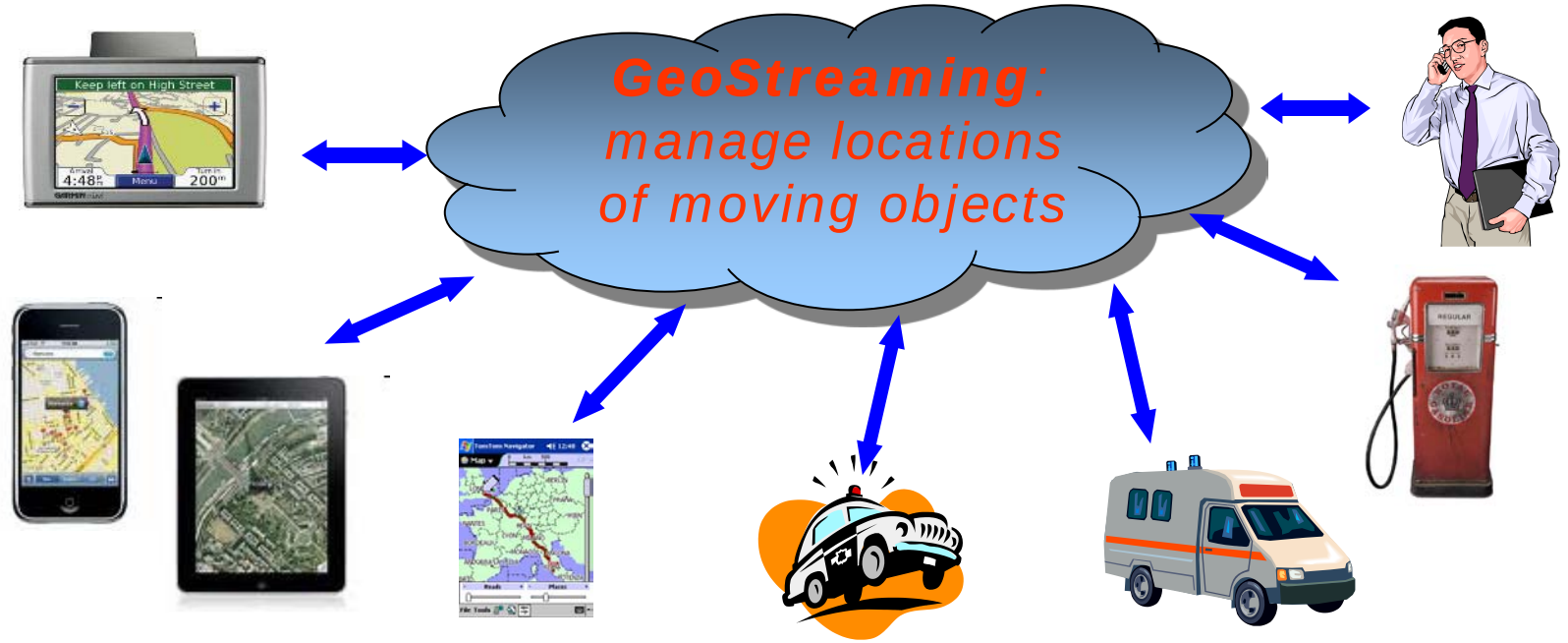
      *"Monitor routers and hubs and issue an alert when anomalous traffic is detected"*




- Queries are persistent, data is volatile
  - users are mostly interested in recent information
  - system must process stream items as they arrive
  - provide fresh results in almost real-time
  - multiple queries may compete for limited resources (memory, CPU)

# Monitoring Applications

- *Complex Event Processing* (*CEP*):
  - rapid event processing, in-depth impact analysis, pattern matching etc. for:
    - business process management • financial trading • network security ...
- Event processing is vital for *location-based services* (*LBS*):
  - navigation
  - emergency calls
  - environmental protection
  - traffic telematics
  - tourist guides
  - advertising ...*and more!*



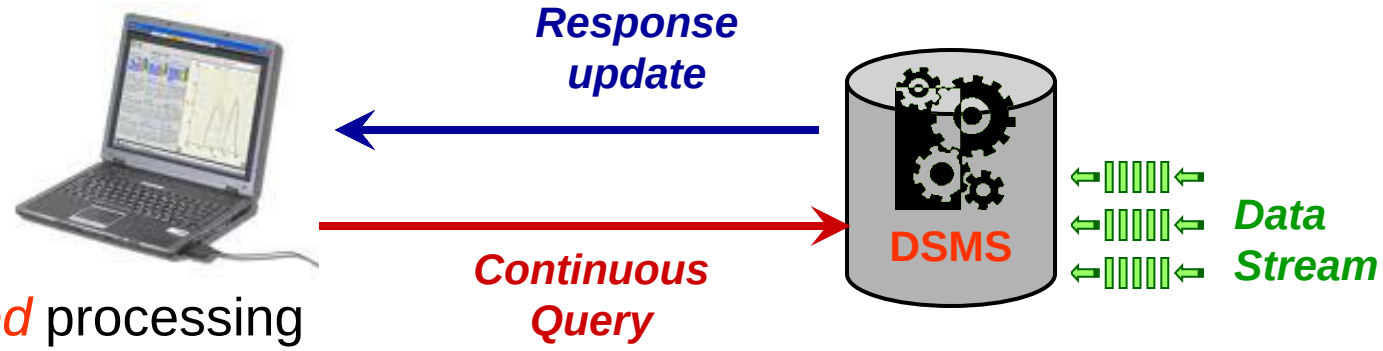
# Keyword Cloud

# Outline of the talk

- Introduction
  - Modern data-intensive monitoring applications
  - The case of location-aware processing
- Issues in Stream Processing
  - A novel processing paradigm
  - Semantics, Evaluation & Approximation
  - Scalability & Optimization
- GeoStreaming: Management of Streaming Locations
  - Analyzing continuously moving objects
  - Evaluating continuous spatiotemporal queries
  - Indexing & summarization requirements
- Perspectives
  - *Stream Engines*: from academic prototypes to industry platforms
  - Challenges & Research directions

# A Novel Processing Paradigm

- Towards Data Stream Management Systems (DSMS)
  - – typical *one-time* queries are the exception, not the rule
  - + concurrent evaluation of multiple long-running *continuous queries*
    - incremental results with *online* processing of incoming data feeds
  - – *pull-based* model of traditional DBMS is not affordable
    - cannot store massive updates on hard disk → slow, costly, *offline*
  - + *push-based* paradigm for processing such volatile data
    - newly arriving items trigger response updates → *data ordering* matters!
    - *in-memory* processing ideal for low latency




*Push-based* processing

# Stream Semantics & Query Language

- A *relational* interpretation of streams:
  - sequence of tuples with a common schema of attributes
    + a *timestamp* from a discrete domain ($T$, $\leq$)
  - Timestamping for each incoming tuple:
    - *time-based* : items have time indications → *simultaneity*
    - *tuple-based* : rank items by their arrival → *ordering*
  - For real-time computation, must restrict the set of inspected tuples
    - *Punctuations*: embedded annotations
    - *Synopses* : data summaries
    - *Windows* convert the unbounded stream into a *temporary finite relation*
      - repeatedly refreshed *sliding* windows: e.g., *items received in past 3 min*
- Query Language: an extension of SQL
  - Continuous Query Language [*STREAM*]
  - StreQuel [*TelegraphCQ*]
  - SQuAl [*Aurora*]
  - GSQL [*Gigascope*]
  - recent efforts towards a common StreamSQL standard
    - bridging the gap between simultaneity and ordering

# Real-time Evaluation

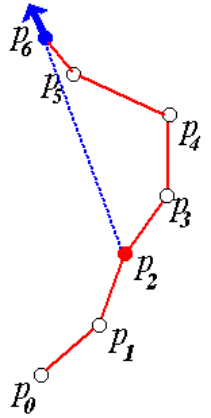
- Continuous Query Execution
  - adaptive to varying query workloads & scalable data volumes
  - *shared evaluation* of multiple user requests via composite query plans
- Approximate Answers
  - Maintain dynamically updateable *synopses*:
    - ➢ *sketches* ➢ *wavelets* ➢ *sampling* ➢ *quantiles* ➢ *histograms* ...
    - mostly for analyzing evolving trends, heavy hitters, outliers, similarities, …
  - Algorithms for stream summarization *trade off accuracy for cost*:
    - ➢ *One-pass computation*, i.e., no backtracking over past items
    - ➢ *Very small memory footprint*, much less than the original stream
    - ➢ *Low processing time* per item to keep up with the stream rate
  - Fast, succinct, but approximate response with *error guarantees*
    - "*At most 3% off the exact answer with high probability*"
  - ➢ Proposals for *load shedding* without processing a portion of data
    - *Semantic / Random*: when exceeding system capacity, evict items of less utility

# Scalable Stream Processing

- Query optimization strategies abound:
  - *rate-based*: maximize query throughput depending on actual arrival rate
  - *multi-query*: share select, join, aggregate, window… expressions
  - *scheduling*: prioritize operators to minimize memory consumption
  - *Quality-of-Service* (QoS): schedule operators and tuples in batches
  - *Eddies*: continuously adapt evaluation order as items arrive
- Centralized processing could become a bottleneck…
  - ➢ *Distributed* computation may offer certain advantages:
    - Load balancing
    - High availability
    - Fault tolerance
  - Minimize communication overhead & maximize sensor lifetime with:
    - in-network processing
    - multi-level communication trees
    - randomized approximation
    - local filters at data sources …
- *XML streams* : sequence of tokens
  - Another line of work for both structured and unstructured data
    - ➢ *appilcations*: personalized content, retail transactions, distributed monitoring, …

# GeoStreaming

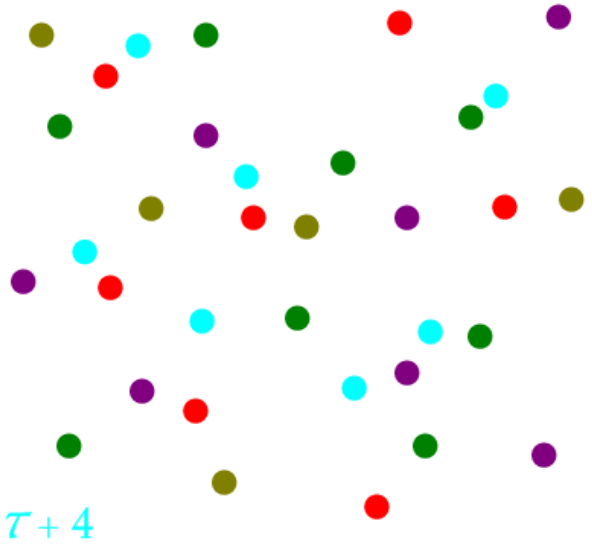
- *Geospatial streams* derived from real-time data acquisition
  - geosensors ~ *vector data* • imagery/satellite ~ *raster data* (mostly)
- Much interest on monitoring *location-aware* moving objects:
  - *numerous* people, merchandise, devices, animals,...
  - *PRESENT* → record their current *location*
  - *PAST* → maintain historical *trajectory*
  - *FUTURE* → predict route / estimate trend




- Streaming locations captured with GPS/RFID
  - *timestamped, georeferenced points* posing challenges:
    - ➢ consume fluctuating, intermittent, voluminous positional updates
    - ➢ provide timely response to spatiotemporal continuous requests
    - ➢ overcome lack of suitable operators in traditional databases
- Algorithmic issues for efficient geostreaming
  - query evaluation – in-memory indexing – data reduction/approximation

# Positional Streams

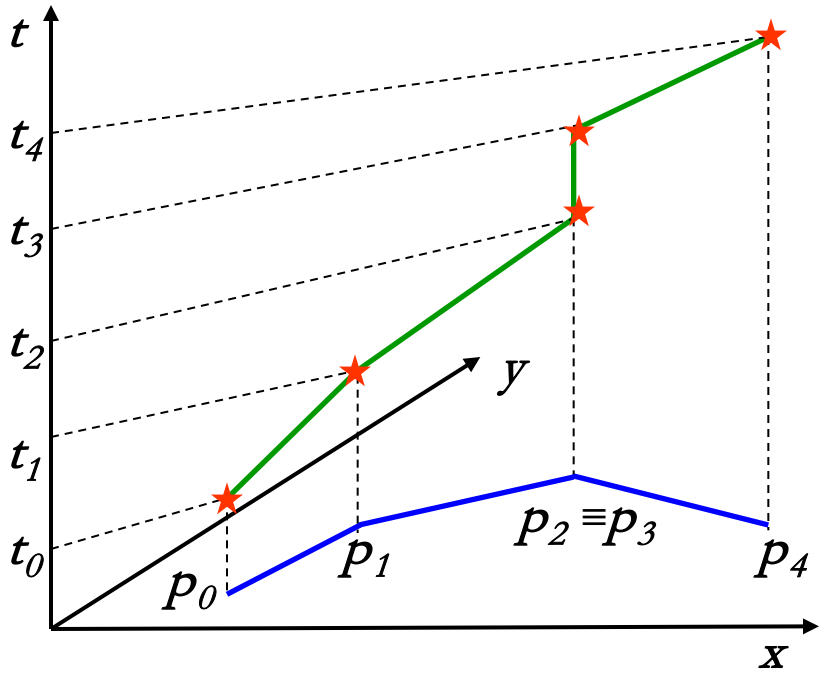
- In *space* domain
  - *locations* : point coordinates of objects
  - usually in 2-D Euclidean space
- In *time* domain
  - *timestamps* at every incoming item
  - varying reporting frequency per object




- Managing *streaming locations*
  - accept incoming flux of object statuses with *space-timestamps*
    - deduce whether objects are actually moving or remain stationary
  - collect unbounded sequences from multiple objects
    - assume that *finite* data feeds arrive per timestamp
  - manipulate *missing* or *noisy* data
    - exploit *correlations* typical in geostreaming data (e.g., traffic patterns)
    - smooth outliers according to archived historical traces

# Trajectory Streams

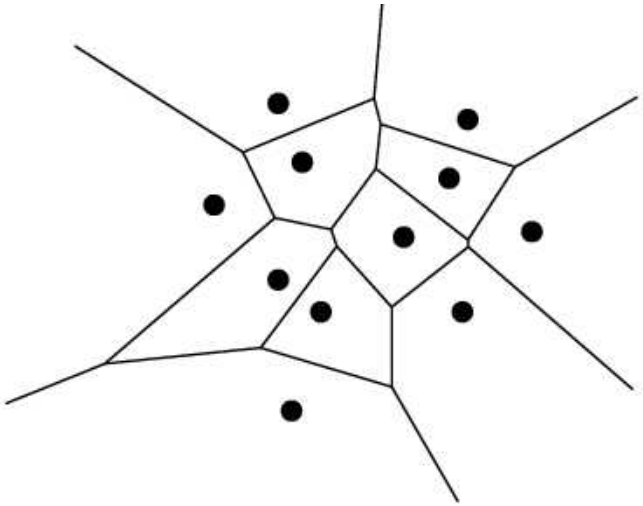
- *Trajectory* of a moving object
  - *in theory*, continuously evolving
    - in both space and time domain
  - *in practice*, a sequence of positions
    - discrete timestamped locations
- Trajectory stream
  - dynamic time series of positions
  - compiled from multiple objects
    - object identity (*oid*) at each tuple

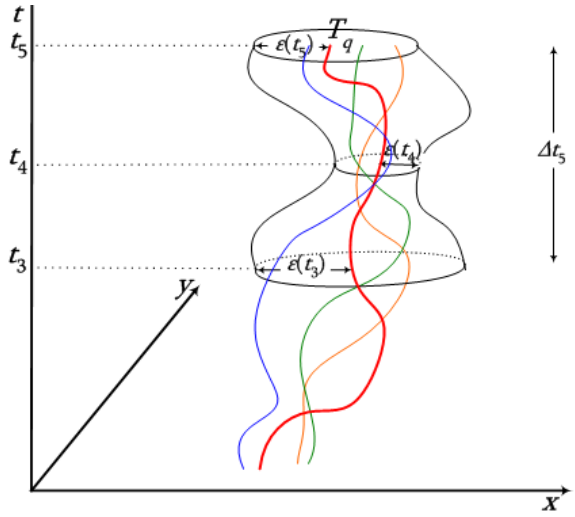



- ➢ *temporal monotonicity* → ordering of incoming locations
- ➢ *spatial locality* in each object's movement → coherent motion
- in-memory *online evaluation* → only segments of trajectories can be retained
  - *object-side*: relay position upon significant deviation from known course
  - *server-side*: abstract recent movement of objects with *windowing*

# Spatiotemporal Continuous Queries

- *Coordinate-based*
  - ➢ Spatial processing
    - range (*with a region predicate*)
    - proximity (*k*-NN, reverse *k*-NN)
    - aggregates (*distinct count*)
    - density areas ...
  - ➢ Geometric computation
    - convex hull
    - Voronoi cell ...
- *Trajectory-based*
  - similarity (*synchronous or time-relaxed*)
  - clustering (*convoys, flocks*)
  - orientation
  - *k*-nearest neighbors (*k*-NN) …





# Online GeoSpatial Processing

- Data summarization
  - Real-time, single-pass compression of *positions*
    - synthesize similarly moving objects into a cluster, discarding its constituents
    - acts like an occasional *load shedder*
  - Dynamic synopses over *trajectories* at varying levels of abstraction
    - *amnesic*, *aging-aware*, *time-decaying*, *multi-resolution*… trajectory simplification
    - progressively coarser representation for older features
  - Other methods:
    - spatiotemporal histograms
    - sketches
    - sampling …
- Indexing transient locations
  - Accelerate *NOW*-related continuous requests, like *range* or *k-NN* search
    - must handle consecutive waves of numerous positional updates
    - build a common index for objects and queries
  - *Data-driven* methods (like *R*-trees) *cannot* easily sustain rapid updates
  - A flair for in-memory *space-driven indexing*
    - uniform *grid partitioning* or quadtrees are mainly employed

# Stream Processing Engines

- Academic prototypes
  - Aurora + Borealis (Brown/MIT/Brandeis)
  - Gigascope (AT&T/Carnegie Mellon)
  - NiagaraST (Wisconsin/Portland State)
  - STREAM (Stanford)
  - TelegraphCQ (UC Berkeley)
- CEP
  - Cayuga [Cornell]
  - Esper and NEsper [EsperTech]
- Spatiotemporal systems
  - SECONDO [Hagen Univ.]
  - PLACE [Purdue]
  - Microsoft StreamInsight Spatial
- Commercial platforms
  - StreamBase
  - Coral8 → Sybase CEP
  - Oracle CEP
  - Microsoft StreamInsight
  - Truviso
  - IBM System S
  - SQLStream
  - …
- Benchmarks
  - Linear Road [Aurora, STREAM]
  - NEXMark [NiagaraST]
  - BerlinMOD [Hagen Univ.]

# Next-Generation Stream Management

- Offer advanced functionality
  - Richer class of queries
    - set-valued results, extensible windows, joins with relational tables, …
  - Dynamic revision of results
    - deal with inherent stream *imperfections* like disorder or noise
  - Multi-level optimizers at varying granules, e.g.:
    - sensor nodes
    - servers
    - server clusters …
- Tackle scalability and load balancing
  - Stream processing *in the cloud*
  - Flexible, highly-distributed resource allocation
    - data emanates from multi-modal devices & flows through heterogeneous networks
- Software enhancements
  - *GUI* for visualization + *API* for fine-grain control over complex events
  - *Application development*: design, build, test, and deploy customized modules
  - *Platform performance*: microsecond latency even for huge workloads

# Infrastructure for GeoStreaming

- Address advanced spatiotemporal requests
  - Modeling and analysis over *positional streams* for special cases:
    - *uncertainty* • multiple dimensions • movement in networks • *indoor awareness*
  - Novel approaches to *trajectory streams* :
    - *navigation*: delineate routes according to actual traffic patterns
    - *personalization*: integrate preferences from user profiles or context
    - explore dynamic *motion patterns* (*flocks*, *convoys*, ...) across time
- Adapt spatial operators to geostreaming mode
  - *Beyond* typical range or *k*-NN search on point locations: *skylines, top-k, …*
  - Handle operands representing *evolving linear* and *polygon* features
  - Weigh real-time events against historical patterns to avoid false alarms
- Trailblazing research opportunities
  - Geostreaming in the cloud
  - Geo-social networks
  - Probabilistic spatial streams
  - Privacy preservation, authentication
  - Real-time spatial data visualization
  - Interoperability & standards …

# References

- Data Streams


**[ACC+03]** D.J. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, and S. Zdonik. Aurora: a New Model and Architecture for Data Stream Management. ***VLDB Journal***, 2003.

**[AAB+05]** D.J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A.S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik. The Design of the Borealis Stream Processing Engine. ***CIDR***, January 2005.

**[AHWY03]** C. Aggarwal, J. Han, J. Wang, and P.S. Yu. A Framework for Clustering Evolving Data Streams. ***VLDB***, September 2003.

**[ABW06]** A. Arasu, S. Babu, and J. Widom. The CQL Continuous Query Language: Semantic Foundations and Query Execution. ***VLDB Journal***, 2006.

**[ACG+04]** A. Arasu, M. Cherniack, E. Galvez, D. Maier, A. Maskey, E. Ryvkina, M. Stonebraker, and R. Tibbetts. Linear Road: A Stream Data Management Benchmark. ***VLDB***, September 2004.

**[AW04]** A. Arasu and J. Widom. Resource Sharing in Continuous Sliding-Window Aggregates. ***VLDB***, September 2004.

**[BBD+02]** B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and Issues in Data Stream Systems. ***PODS***, May 2002.

**[BAF+09]** I. Botan, G. Alonso, P.M. Fischer, D. Kossmann, and N. Tatbul. Flexible and Scalable Storage Management for Data-intensive Stream Processing. ***EDBT***, March 2009.

**[BDD+10]** I. Botan, R. Derakhshan, N. Dindar, L. Haas, R. Miller, and N. Tatbul. SECRET: A Model for Analysis of the Execution Semantics of Stream Processing Systems. ***VLDB***, September 2010.

**[BS03]** A. Bulut and A.K. Singh. SWAT: Hierarchical Stream Summarization in Large Networks. ***ICDE***, March 2003.

**[CCD+03]** S. Chandrasekaran, O. Cooper, A. Deshpande, M.J. Franklin, J.M. Hellerstein, W. Hong, S. Krishnamurthy, S.R. Madden, V. Raman, F. Reiss, and M.A. Shah. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. ***CIDR***, January 2003.

**[CG08]** G. Cormode and M. Garofalakis. Approximate Continuous Querying over Distributed Streams. ***ACM TODS***, 2008.

**[CS03]** E. Cohen and M. Strauss. Maintaining Time-Decaying Stream Aggregates. ***PODS***, June 2003.

# References

- Data Streams (*cont'd*)


**[FM85]** P. Flajolet and G.N. Martin. Probabilistic Counting Algorithms for Database Applications. ***Journal of Computer and Systems Sciences***, 1985.

**[GO05]** L. Golab and M. Tamer Ozsu. Update-Pattern-Aware Modeling and Processing of Continuous Queries. ***SIGMOD***, June 2005.

**[JMS+08]** N. Jain, S. Mishra, A. Srinivasan, J. Gehrke, J. Widom, H. Balakrishnan, U. Cetintemel, M. Cherniack, R. Tibbetts, and S. Zdonik. Towards a Streaming SQL Standard. ***VLDB***, August 2008.

**[JMSS05]** T. Johnson, S. Muthukrishnan, V. Shkapenyuk, O. Spatscheck. A Heartbeat Mechanism and its Application in Gigascope. ***VLDB***, September 2005.

**[LMP+05]** J. Li, D. Maier, K. Tufte, V. Papadimos, P. Tucker. Semantics and Evaluation Techniques for Window Aggregates in Data Streams. ***SIGMOD***, June 2005.

**[MPN+09]** L. Al Moakar, T. Pham, P. Neophytou, P. Chrysanthis, A. Labrinidis, and M. Sharaf. Class-based Continuous Query Scheduling for Data Streams. ***DMSN***, August 2009.

**[PVK+04]** T. Palpanas, M. Vlachos, E. Keogh, D. Gunopulos, and W. Truppel. Online Amnesic Approximation of Streaming Time Series. ***ICDE***, March 2004.

**[PS06]** K. Patroumpas and T. Sellis. Window Specification over Data Streams. ***ICSNW***, March 2006.

**[PS09b]** K. Patroumpas and T. Sellis. Window Update Patterns in Stream Operators. ***ADBIS***, September 2009.

**[PS10]** K. Patroumpas and T. Sellis. Multi-granular Time-based Sliding Windows over Data Streams. ***TIME***, September 2010.

**[PS11]** K. Patroumpas and T. Sellis. Maintaining Consistent Results of Continuous Queries under Diverse Window Specifications. ***Information Systems Journal***, March 2011.

**[SCZ05]** M. Stonebraker, U. Cetintemel, and S. Zdonik. The 8 Requirements of Real-Time Stream Processing. ***SIGMOD Record***, December 2005.

**[TMSS07]** P. Tucker, D. Maier, T. Sheard, and P. Stephens. Using Punctuation Schemes to Characterize Strategies for Querying over Data Streams. ***TKDE***, September 2007.

# References

- Stream Processing Engines

**StreamBase**

http://www.streambase.com/

**Sybase CEP**

http://www.sybase.com/products/financialservicessolutions/sybasecep

**Oracle CEP**

http://www.oracle.com/us/technologies/soa/service-oriented-architecture-066455.html

**Microsoft StreamInsight**

http://msdn.microsoft.com/en-us/library/ee362541.aspx

**Truviso**

http://www.truviso.com/

**IBM System S**

http://www-01.ibm.com/software/data/infosphere/streams/

**SQLStream**

http://www.sqlstream.com/

**Esper and NEsper**

http://esper.codehaus.org/

# References

- Moving Objects


**[BHT05]** P. Bakalov, M. Hadjieleftheriou, and V. Tsotras. Time Relaxed Spatiotemporal Trajectory Joins. ***ACM GIS***, November 2005.

**[DBG09]** C. Düntgen, T. Behr, and R.H. Güting. BerlinMOD: a benchmark for moving object databases. ***VLDBJ***, 2009.

**[GL06]** B. Gedik, L. Liu. Mobieyes: A Distributed Location Monitoring Service using Moving Location Queries. ***Transactions on Mobile Computing***, 2006.

**[GLWY07]** B. Gedik, L. Liu, K.L. Wu, and P.S. Yu. Lira: Lightweight, Region-aware Load Shedding in Mobile CQ Systems. ***ICDE***, April 2007.

**[FGPT07]** E. Frentzos, K. Gratsias, N. Pelekis, Y. Theodoridis. Algorithms for Nearest Neighbor Search on Moving Object Trajectories. ***GeoInformatica***, 2007.

**[HXL05]** H. Hu, J. Xu, and D. L. Lee. A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects. ***SIGMOD***, June 2005.

**[JYZ+08]** H. Jeung, M. Lung Yiu, X. Zhou, C.S. Jensen, and H. Tao Shen. Discovery of convoys in trajectory databases. ***PVLDB***, August 2008.

**[KDA+10]** S.J. Kazemitabar, U. Demiryurek, M. Ali, A. Akdogan, and C. Shahabi. Geospatial Stream Query Processing using Microsoft SQL Server StreamInsight. ***PVLDB***, September 2010.

**[MXA04]** M. Mokbel, X. Xiong, and W.G. Aref. SINA: Scalable Incremental Processing of Continuous Queries in Spatiotemporal Databases. ***SIGMOD***, June 2004.

**[MXHA05]** M. Mokbel, X. Xiong, M. Hammad, and W.G. Aref. Continuous Query Processing of Spatio-Temporal Data Streams in PLACE. ***Geoinformatica***, December 2005.

**[MHP05]** K. Mouratidis, M. Hadjieleftheriou, and D. Papadias. Conceptual Partitioning: An Efficient Method for Continuous Nearest Neighbor Monitoring. ***SIGMOD***, June 2005.

**[PS04]** K. Patroumpas and T. Sellis. Managing Trajectories of Moving Objects as Data Streams. ***STDBM***, August 2004.

**[PPS06]** M. Potamias, K. Patroumpas, and T. Sellis. Sampling Trajectory Streams with Spatiotemporal Criteria. ***SSDBM***, July 2006.

# References

- Moving Objects (*cont'd*)


**[PS07]** K. Patroumpas and T. Sellis. Semantics of Spatially-aware Windows over Streaming Moving Objects. ***MDM***, 2007.

**[PPS07]** M. Potamias, K. Patroumpas, and T. Sellis. Online Amnesic Summarization of Streaming Locations. ***SSTD***, 2007.

**[PMS07]** K. Patroumpas, T. Minogiannis, and T. Sellis. Approximate Order-k Voronoi Cells over Positional Streams. ***ACM GIS***, November 2007.

**[PS08]** K. Patroumpas and T. Sellis. Prioritized Evaluation of Continuous Moving Queries over Streaming Locations. ***SSDBM***, July 2008.

**[PKS08]** K. Patroumpas, E. Kefallinou, and T. Sellis. Monitoring Continuous Queries over Streaming Locations (demo paper). ***ACM GIS***, November 2008.

**[PS09a]** K. Patroumpas and T. Sellis. Monitoring Orientation of Moving Objects around Focal Points. ***SSTD***, July 2009.

**[PJT00]** D. Pfoser, C. Jensen, and Y. Theodoridis. Novel Approaches in Query Processing for Moving Objects. ***VLDB***, September 2000.

**[SG09]** M. Attia Sakr and R. H. Güting. Spatiotemporal Pattern Queries in Secondo. ***SSTD***, July 2009.

**[SS06]** M. Sharifzadeh and C. Shahabi. Utilizing Voronoi Cells of Location Data Streams for Accurate Computation of Aggregate Functions in Sensor Networks. ***GeoInformatica***, March 2006.

**[TKC+04]** Y. Tao, G. Kollios, J. Considine, F. Li, and D. Papadias. Spatio-Temporal Aggregation Using Sketches. ***ICDE***, March 2004.

**[VBT09]** M. Vieira, P. Bakalov, and V. Tsotras. On-Line Discovery of Flock Patterns in Spatio-Temporal Data. ***ACM GIS***, November 2009.

**[WGT07]** W. Wu, W. Guo, and K.-L. Tan. Distributed Processing of Moving k-Nearest-Neighbor Query on Moving Objects. ***ICDE***, April 2007.

**[XMA05]** X. Xiong, M. Mokbel, and W. Aref. SEA-CNN: Scalable Processing of Continuous k-Nearest Neighbor Queries in Spatiotemporal Databases. ***ICDE***, April 2005.

**[YPK05]** X. Yu, K. Q. Pu, and N. Koudas. Monitoring k-Nearest Neighbor Queries Over Moving Objects. ***ICDE***, April 2005.