Real-Time Data-to-Decision

IT & Wireless Convergence Group

Dakshi Agrawal, agrawal@us.ibm.com

Outline

- Overview of real-time network analytics platform
- Example analytics
- InfoSphere Streams deep dive/Analysis of twitter data/time-series toolkit

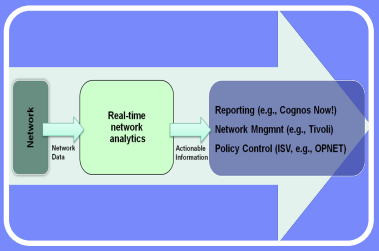
Mobile Network Operator (MNO) challenges and opportunities

MNOs want to monetize data flowing on their network and realize the vision of Smarter Network

- Over-the-top providers are reaping benefits of explosive growth in data volume leaving MNOs as just network pipe providers

Innovation in offered services for new revenue generation

Optimization of MNO infrastructure for cost avoidance

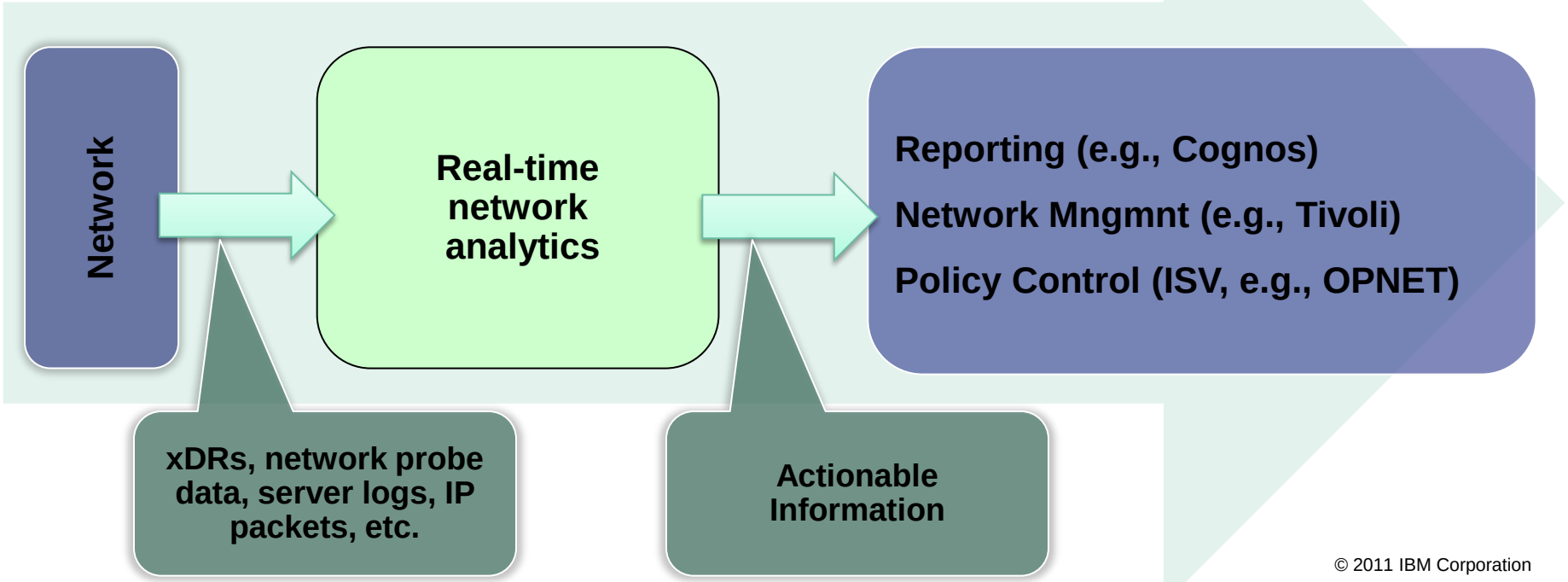


Real-time data-to-decision

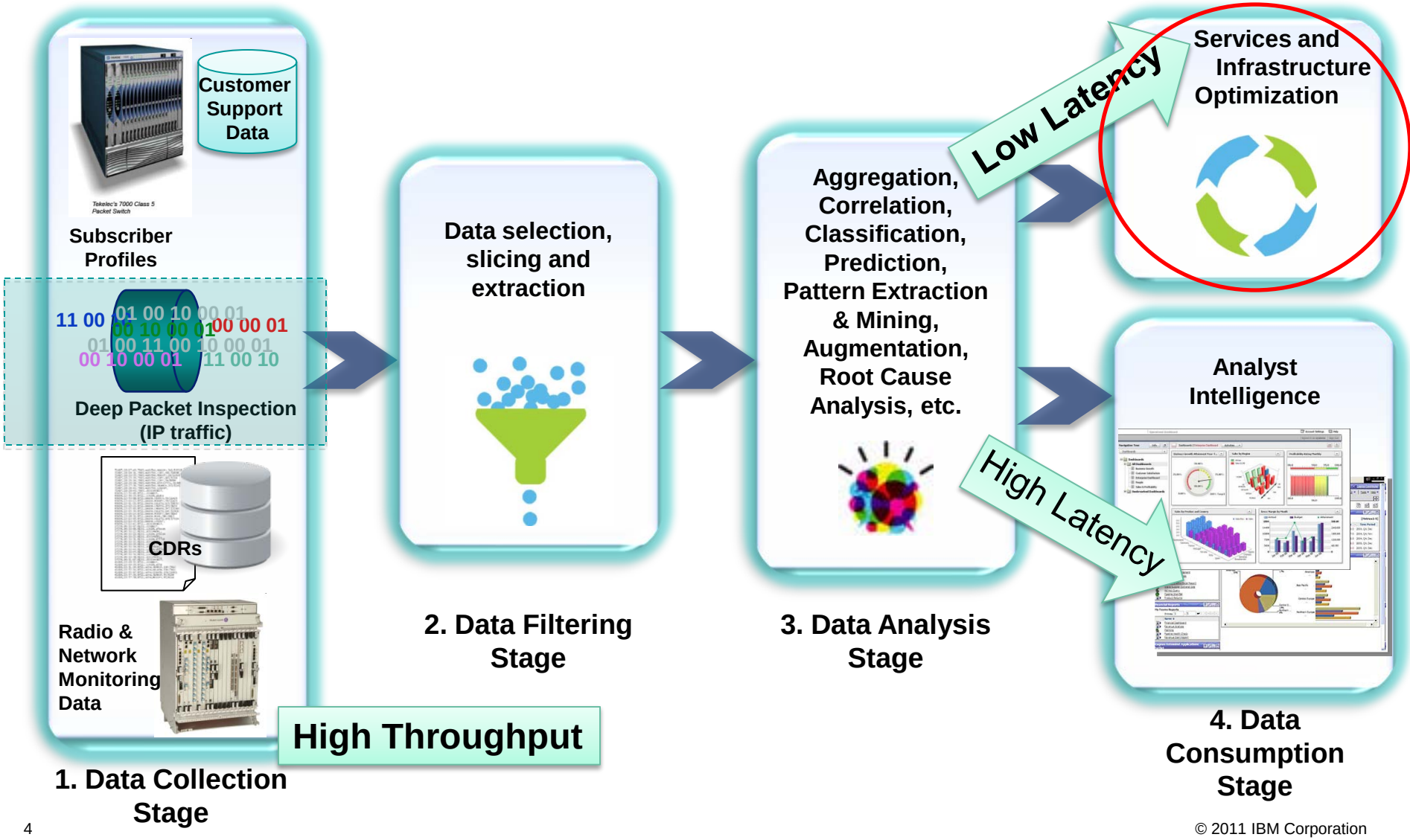
- ingest *high throughput* network data and create *actionable* information

What is real-time network analytics

- An infrastructure that works in the network
  - receives network data (e.g., xDRs, network probe data, server logs, IP packets) as an input stream and integrates them into analytics products (e.g., IBM InfoSphere Streams)
  - performs analytics in real-time on the incoming network data
  - produces actionable information (to detect situations, produce high-level reports and metadata information)
- Actionable information can then be fed into other components to enable real-time reporting, network management, policy control etc.



Real-time network analytics for services and infrastructure optimization



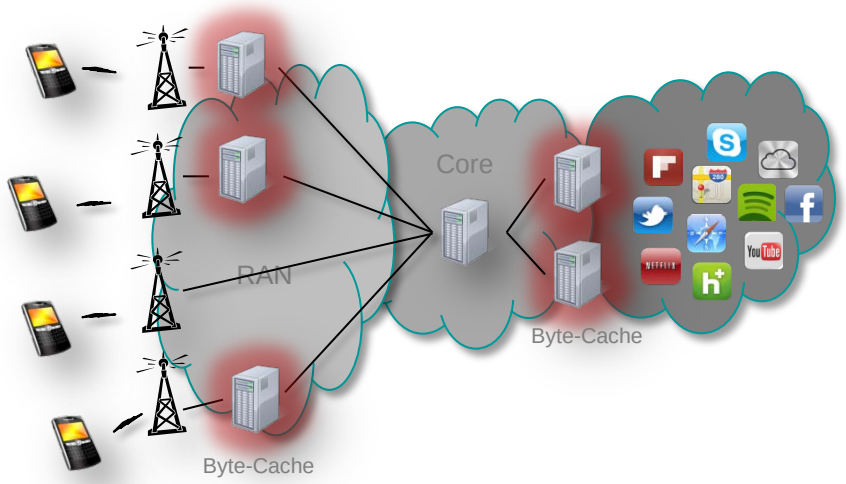
Example usecases for service optimization for new revenue generation

- Abusive/fraudulent user detection in real-time*
  - Identify the users who are tethering without paying for additional tethering services
  - Identify the users who are the heaviest users of traffic in real-time and alert network management systems to react to them
- Wireless bandwidth on-demand
  - Enable new services for mobile network operators (e.g., “1-800” service for data, one time passes)
- Location based services
  - location based adverts, consumer tracking

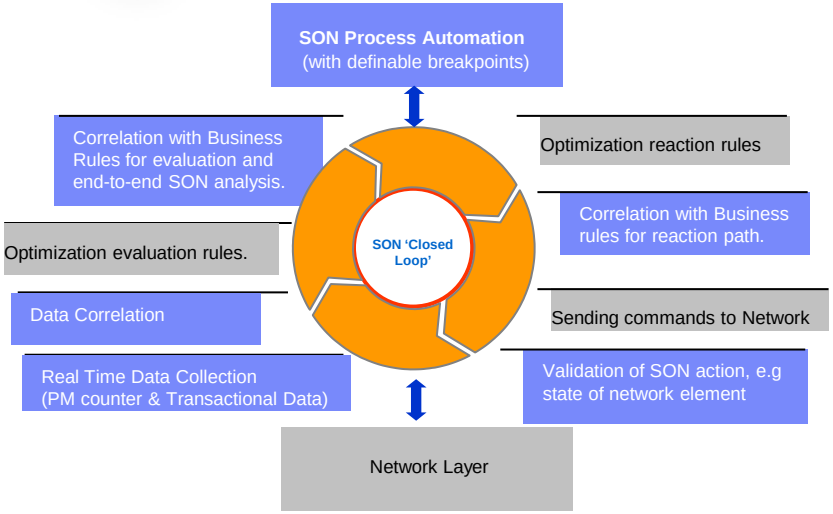
Before you continue…
Check out this
one time tethering pass

Example usecases for infrastructure optimization for cost avoidance

- Frequent sites and frequent applications*
  - Identify the highest frequency sites and applications in real-time so network management tools can optimize/modify their configurations



- Real-time self-organizing networks (e.g., unicast to multicast, Automatic Neighbor Relation (ANR) configuration, etc.)*
  - Adjustment of network configuration in response to current situation



Real-time network analytics platform

Variable Scalability

Different network interception points in the network require different scale

Edge of the Network: Thousands of users

Core of the Network: Millions of users

- *InfoSphere Streams clustering capability to seamlessly scale across multiple nodes*

Real-time

Network operational improvements require packet-processing and decision-enablement capabilities in the order of a few seconds

- *Adaptors to ingest xDR feeds/other network data into IBM InfoSphere Streams*
- *High-throughput low-latency operators that provide real-time network analytics*

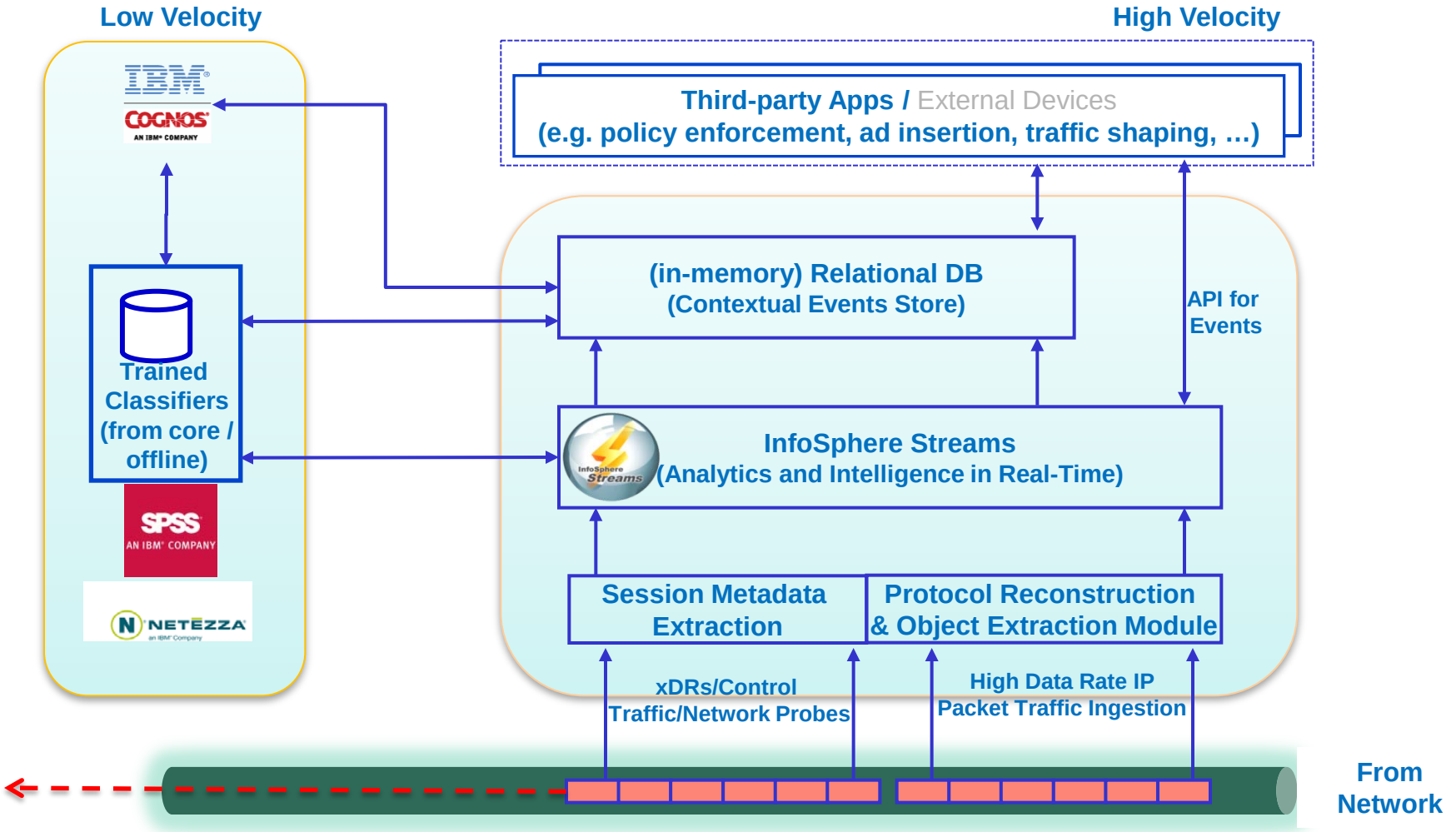
Proactive Decision Making

The latencies inherent in network controls require real-time network analytics platform to predict network conditions as they will be sometime in the future – models will be based on past and current conditions

- *Leverage SPSS and IBM Research developed predictive algorithms and modeling assets*

SPSS
AN IBM COMPANY

Real-time network analytics platform architecture



- ***High Volume* of data:** faster than a database can handle
- ***Complex Analytics*:** correlation from multiple sources and/or signals
- ***Time Sensitive*:** lower latency than possible with the store-and-process paradigm
- ***Scalability*:** scale out through multiple cores/machines for processing

Example deployments in other industries

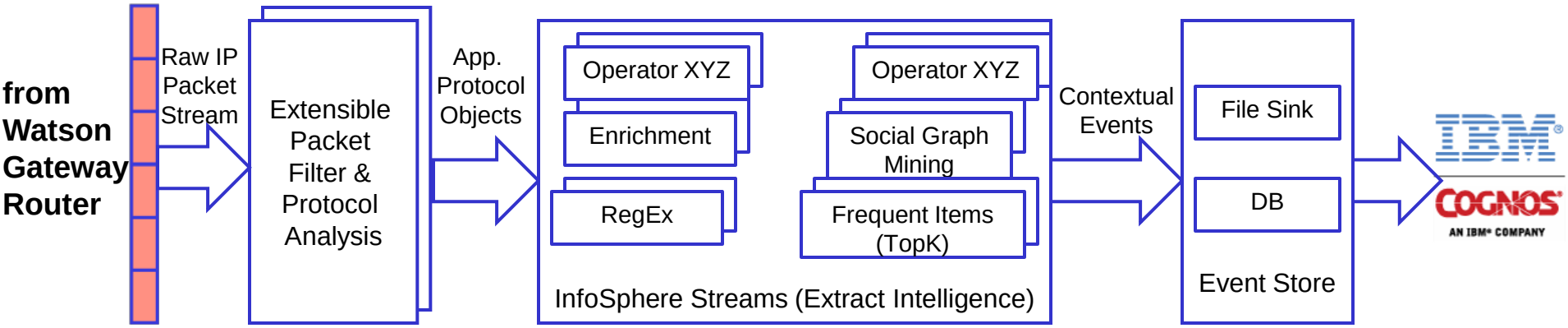
- Healthcare (Univ of Ontario Medical Center, Columbia Univ, etc.)
  - Continuous monitoring of patients and prediction of future patient conditions
  - Significant speed-up in responding to patient conditions (from minutes/hours to seconds)

- Transportation (Stockholm, Dublin, etc.)
  - Processing hundreds of thousands of GPS records per second
  - Estimate traffic conditions in city using data from various sources
  - Real-time, traffic-dependent shortest path algorithms can cut up to 65% of travel time in Stockholm

- Cybersecurity (US Federal Aviation Administration, etc.)
  - Detection of botnets, worms, infected hosts and anomalous traffic
  - Combination of online and offline analytics helps in rapid detection

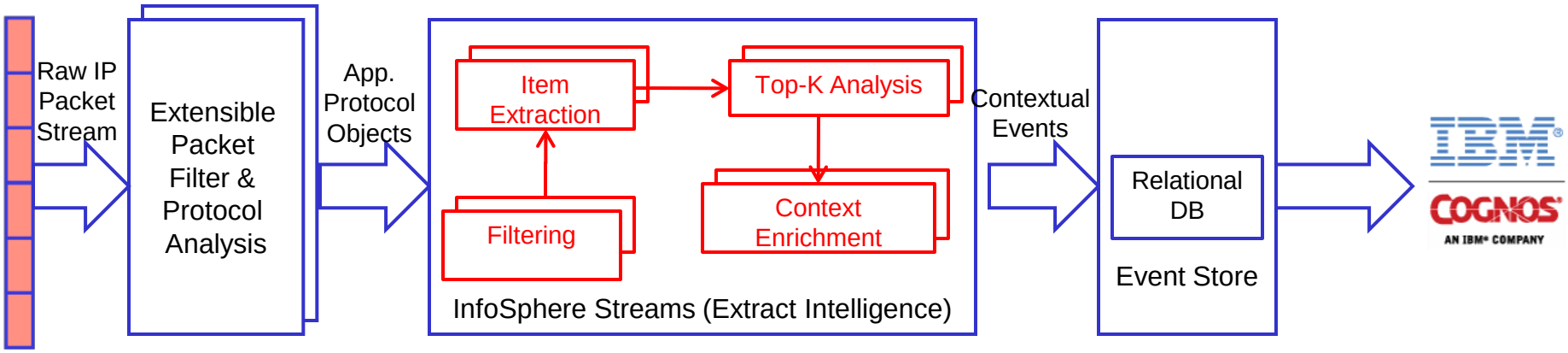
- Government
  - Various intelligence applications with governments around the world

Advanced real-time analytics pipeline – testbed@IBM Watson



- **Went live in Aug. 2011 at Watson network for deep analysis**
  - Basis to evaluate end-to-end use cases in a production-like environment
  - X5550 8x2 @ 2.67 GHz, 32GB RAM, 3TB encrypted storage
- **Streams operators implementing high-throughput analytics**
  - Frequent items analysis (“heavy-hitters”)
  - Facebook social graph mining
  - Time series analysis and forecasting
- **Cognos Dashboard used *only* for visualization (low velocity data consumption)**

Application 1. Top-K ("Heavy Hitters") Discovery



- **High-throughput and memory-efficient Frequent Item ("Top-K") analysis**
- **Generic Top-k item lists for any type of entity & metric over a time- or item- window**
  - Any item represented by a string: host, URLs, IP addresses, keywords, content types, etc.
  - Any numerical metric: number of connection requests, number of bytes transferred, etc.
- **Filtering based on IP address/protocol headers/content allows for different views:**
  - Network-centric: top-k statistics on network-wide metrics: traffic volume, # sessions, etc.)
  - User-centric: statistics on single IP addresses for building user profiles
  - Content-centric: items of interest extracted from content (e.g. keywords, tags, etc.)

Frequent Items : algorithm families*

- Counter-based: keep (approximate) counters for each item
  - Frequent(k)
  - LossyCounting(k) : deltas for upper bound of frequency
  - SpaceSaving(k) : replace item with lowest count

- Quantile algorithms: for a frequent item i with f_i > 2*epsilon*n, item i is the phi-quantile for all phi's in the range rank(i)+epsilon to rank(i+1)-epsilon. This problem is more general that frequent items (and also slower to solve)
  - GK algorithm: similar to LossyCounting, but keeps a *total order* of items according to their count
  - Qdigest

- Sketches: use of (hash) functions to define linear projections of input. Only approach that supports deletions
  - CountSketch & CountMin Sketch
  - Hierarchical CountSketch & CountMin
  - Group Testing

- **Note 1**: Quantiles and Sketches solve a bigger problem (part of which is frequency estimation), hence they are slower
- **Note 2**: A naïve algorithm has O(n) space complexity (i.e. to the order of the whole input data). The objective of the approximation algorithms is to **bound memory** (for example, to O(k))

* Source: "Finding Frequent Items in Data Streams", in Proc. VLDB 2008

SpaceSaving(k)

Algorithm 3.3: SPACESAVING(k)

$n \leftarrow 0$;
$T \leftarrow \emptyset$;
for each i :
do
- $n \leftarrow n + 1$;
- **if** $i \in T$
  - **then** $c_i \leftarrow c_i + 1$;
  - **else if** $|T| < k$
    - **then**
      - $T \leftarrow T \cup \{i\}$;
      - $c_i \leftarrow 1$;
    - **else**
      - $j \leftarrow \arg\min_{j \in T} c_j$;
      - $c_i \leftarrow c_j + 1$;
      - $T \leftarrow T \cup \{i\} \backslash \{j\}$;

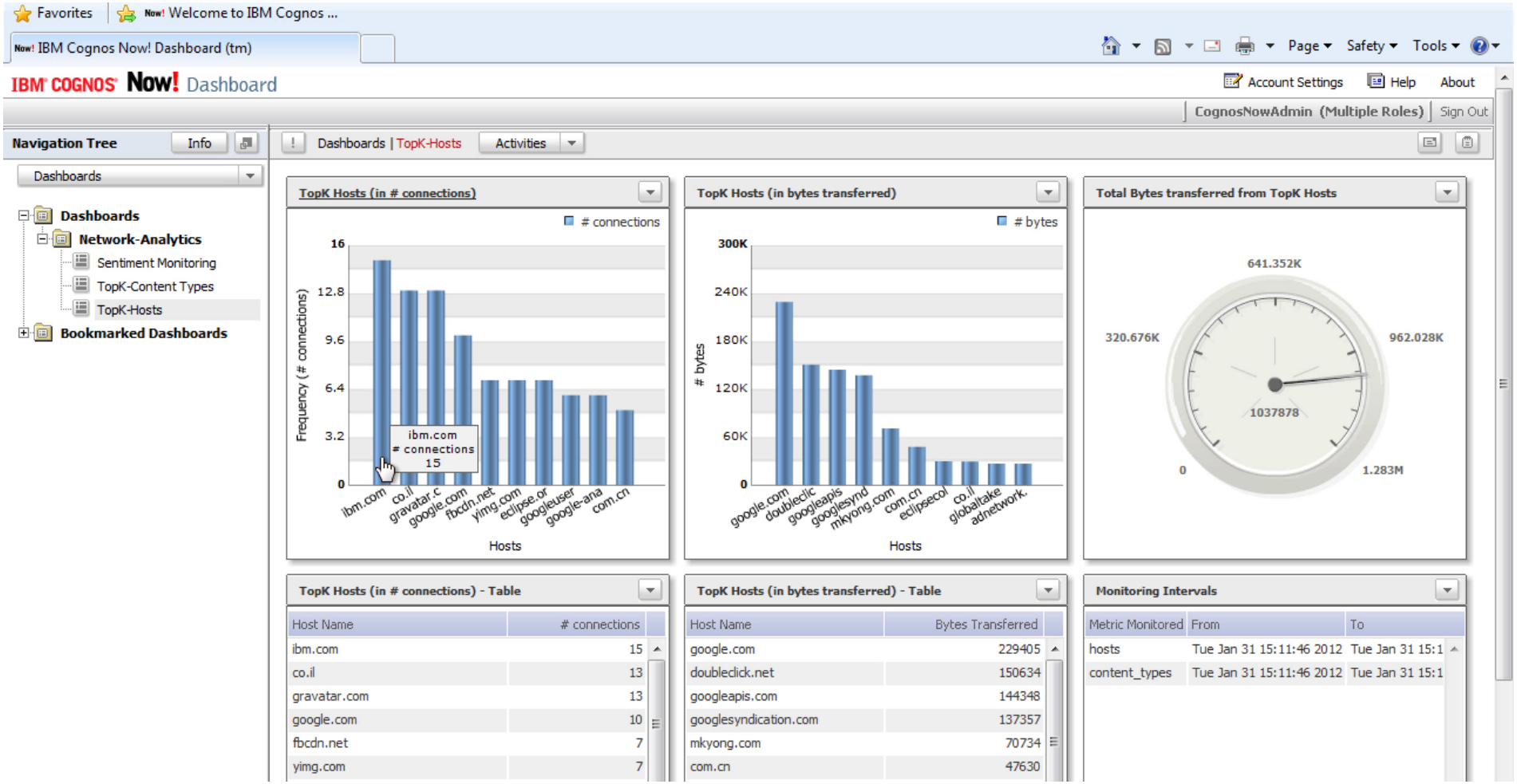
Notes

- If new item does not match a previously stored one, replace (item, count) pair with smallest count with the new item
- **Accuracy**: error of (epsilon*n)
- **Space** requirements: O(k)
- **Time** complexity: O(log(k)) for implementations using heap

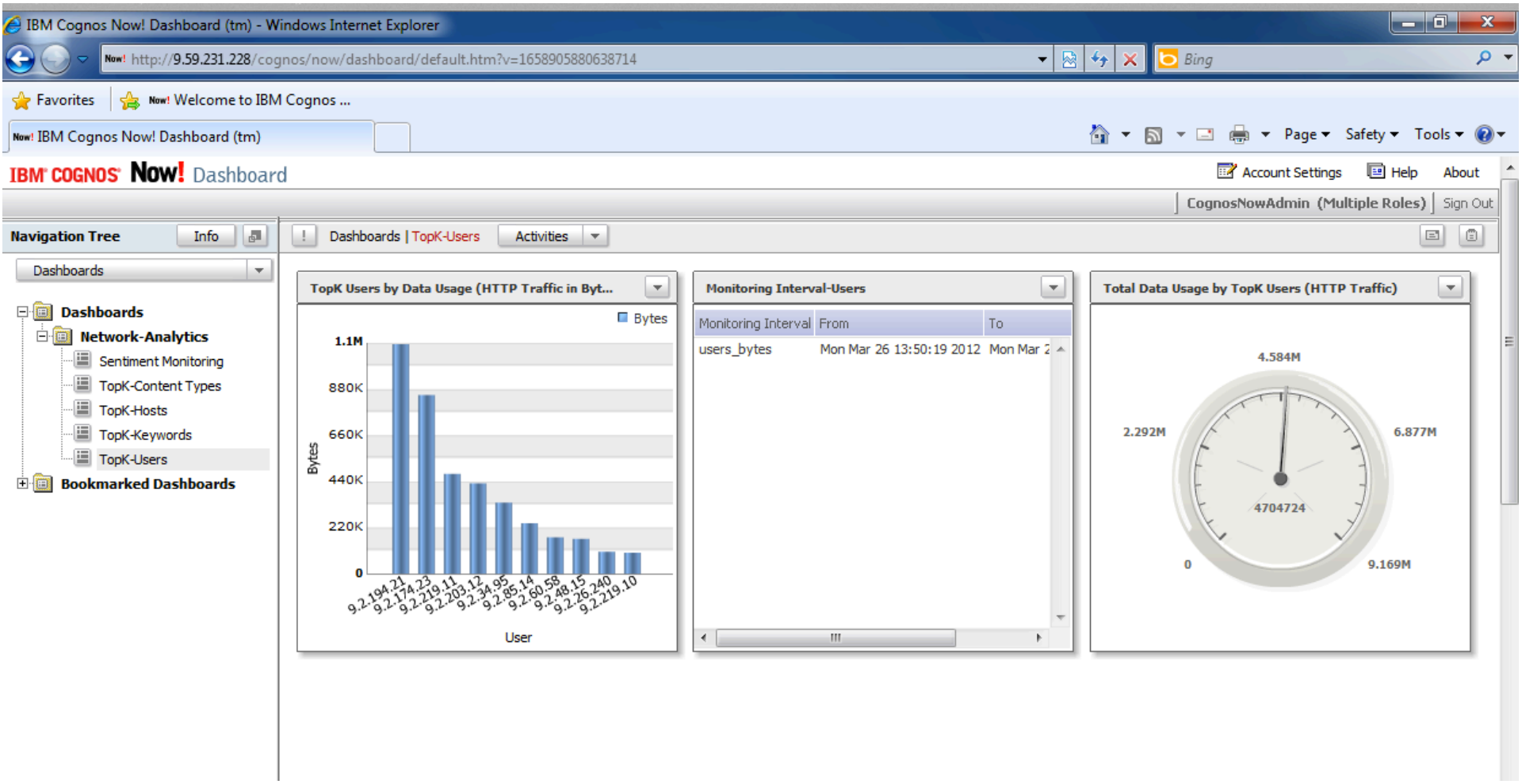
Freq Items Algorithms: Comparison chart on *synthetic data*

Category	Algorithm	Update Speed (/msec)	Recall (Accuracy)	Precision (Accuracy)	Avg. Relat. Error (Accuracy)	Comments
Counting-based	F	~9k-12k	100%	~15%	0.27-0.45	
	LC	~2k-4k	100%	~60%	0	
	LCD	~2k-4k	100%	60%-100%	0	Deltas per item
	SSL	~13k-16k	100%	100%	0	**Only Unary count updates (+1)**
	SSH	~6k	100%	100%	0	Heap implementation
Quantile-based	GK	200-800	NA	3%-5%	0.15-0.28	
	QD	~4k-8k	NA	30%-60%	0.18-0.28	
Sketches	CS	~1000	94%-100%	95%-100%	0.02-0.3	
	CMH	~2k-2.5k	100%	74%-88%	0.07-0.15	
	CGT	~2.7k-4k	100%	87%-100%	0.12-0.17	

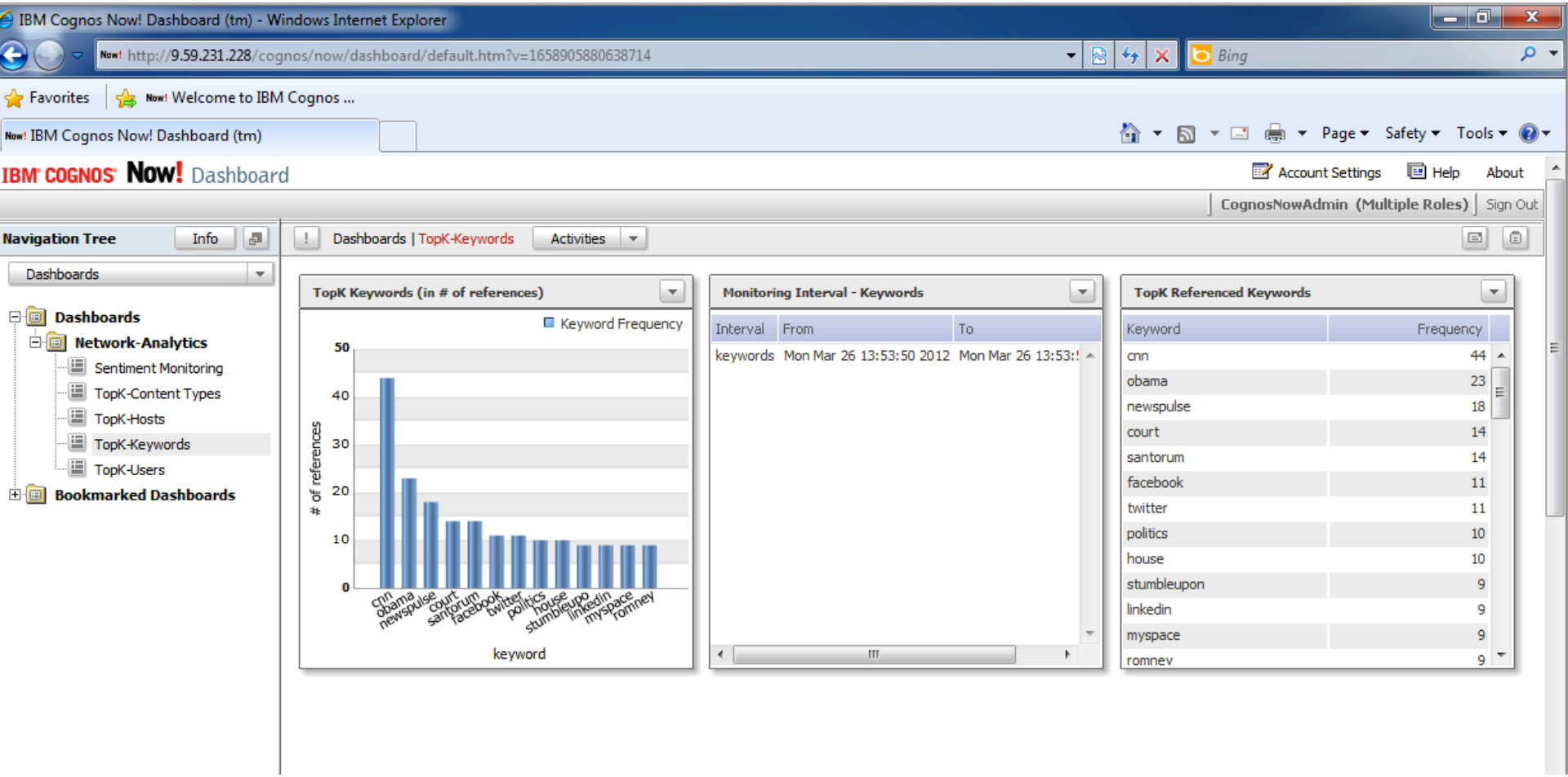
Application 1. Cognos for Top-*k* Dashboards - **Websites**



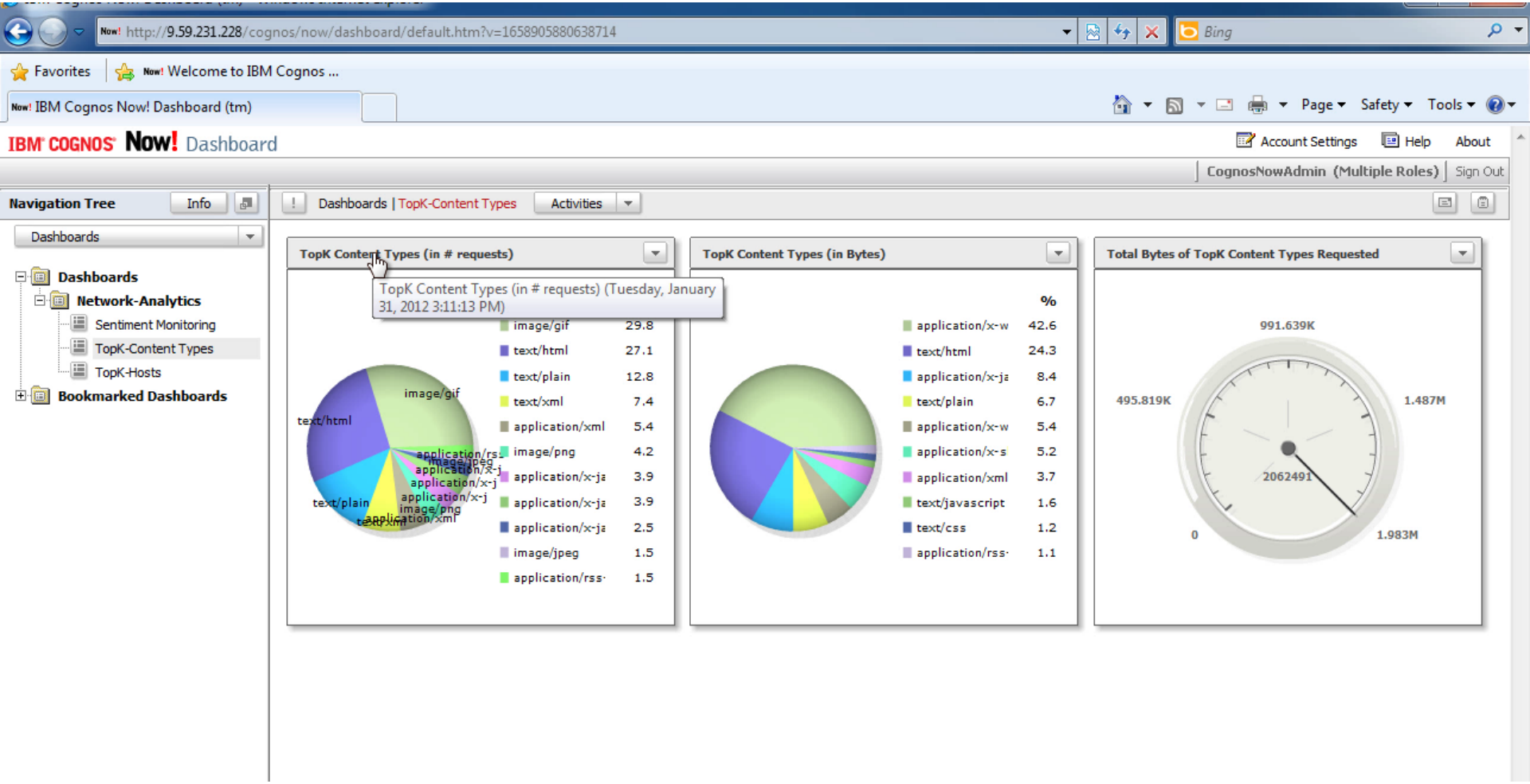
Application 1. Cognos for Top-*k* Dashboards – **Users**



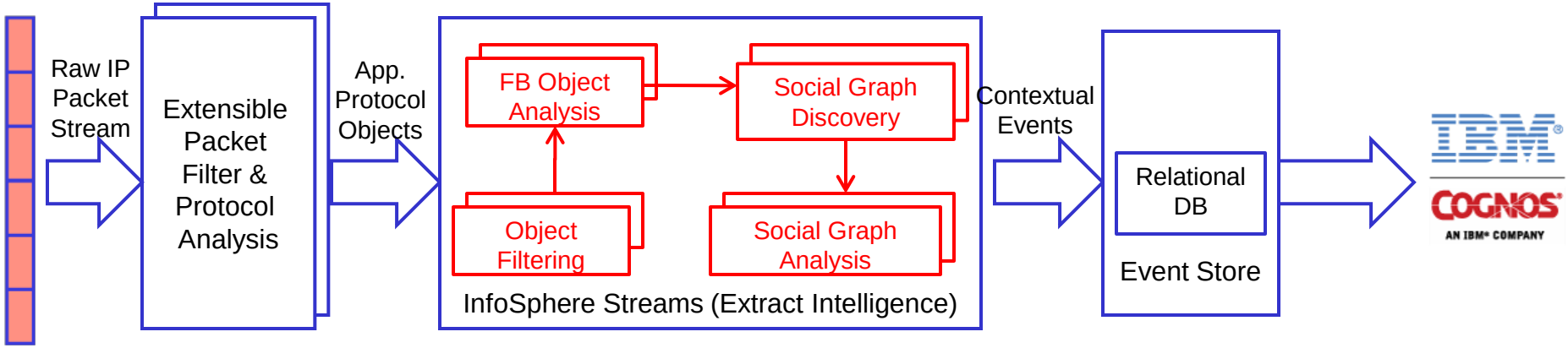
Application 1. Cognos for Top-*k* Dashboards - **Keywords**

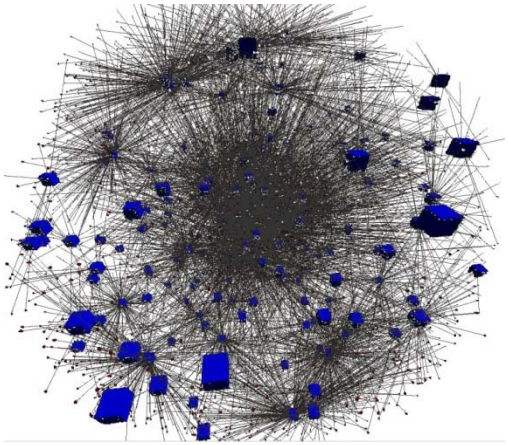


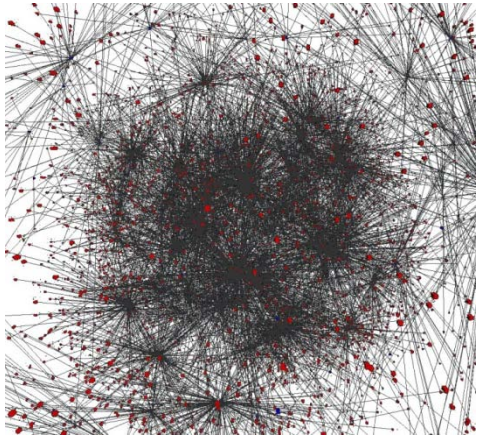
Application 1. Cognos for Top-*k* Dashboards – **Content Types**



Application 2. Real-time Social Network Discovery and Analysis







- **Real-time social network (Facebook) mined from Watson traffic**
  - Blue nodes are IBM Researchers (~300); Red nodes are friends of IBM Researchers (~6000)
- **Identify "currently active" social links and the nature of interaction between two users**
  - E.g., facebook chat, wall post, comments, etc.

Application 3: Real-time Sentiment Analysis

- Example: Filter user comments on cnn.com for the topic “Occupy Wall Street”
- Sentiment analysis picks up positive, neutral, or negative comments for further analysis

“Welcome to Day 9: Let's name those whom we're protesting against. 'Corporations' and 'bankers' and 'politicians' are good umbrella terms. But we also need to be specific for those new members of the movement who may benefit from our knowledge and experiences. Name the culprits and name their crimes!”

“wall street cats are getting record bonuses this year **** jeeze memo to thugs stop robbing the hood start robbing wall street”

“A violent clash with marching members of the loose protest movement Occupy Wall Street on Saturday suggested the flip side of a police force trained to fight terrorism.”

“why must wall street ****ing mess with me and you”

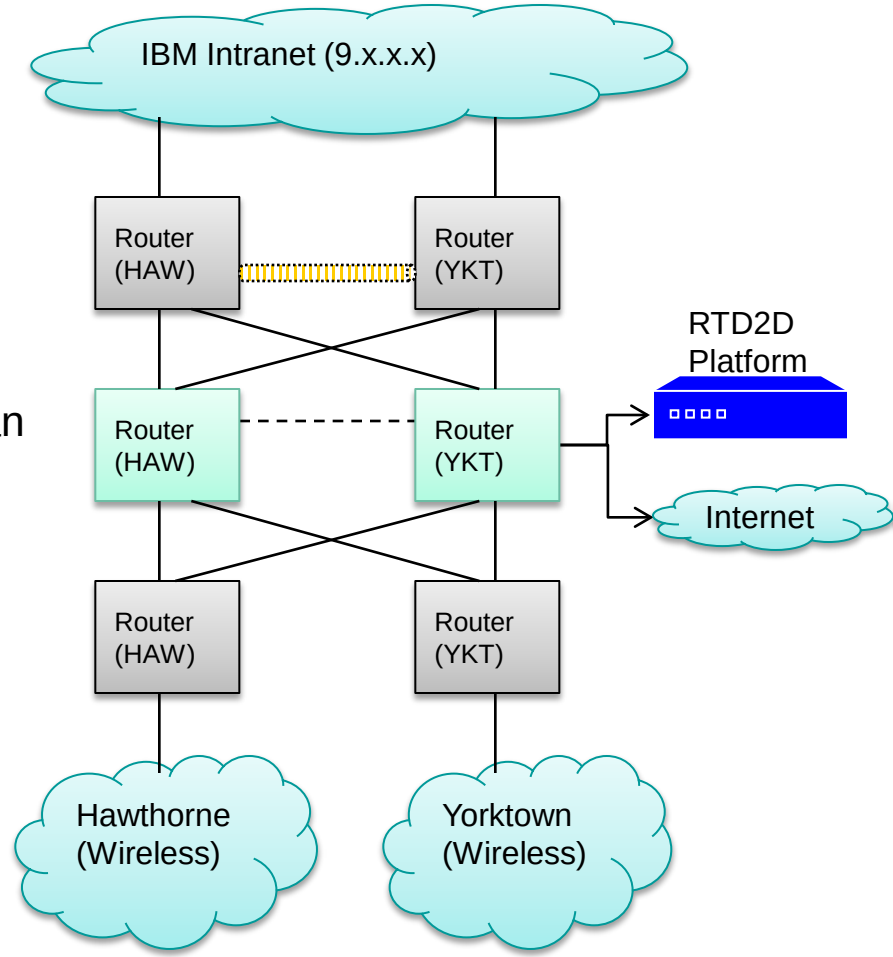
“wall street abuses the term talent”

Topic: “Occupy Wall Street” from www.cnn.com user comments

Statistical approach to sentiment analysis: P. Melville, W. Gryc and R. D. Lawrence. Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification. In KDD 2009.

Throughput measurements on real-time data-to-decision platform

- In Watson network
  - peak traffic of less than 100 Mbps
  - CPU utilization for analytics ~5% for all
- For testing peak performance, real-traces at accelerated rates were passed through RTD2D platform
  - on IBM X3650M3 (2.67 GHz), the system can support a throughput of 2Gbps per core for packet reassembly

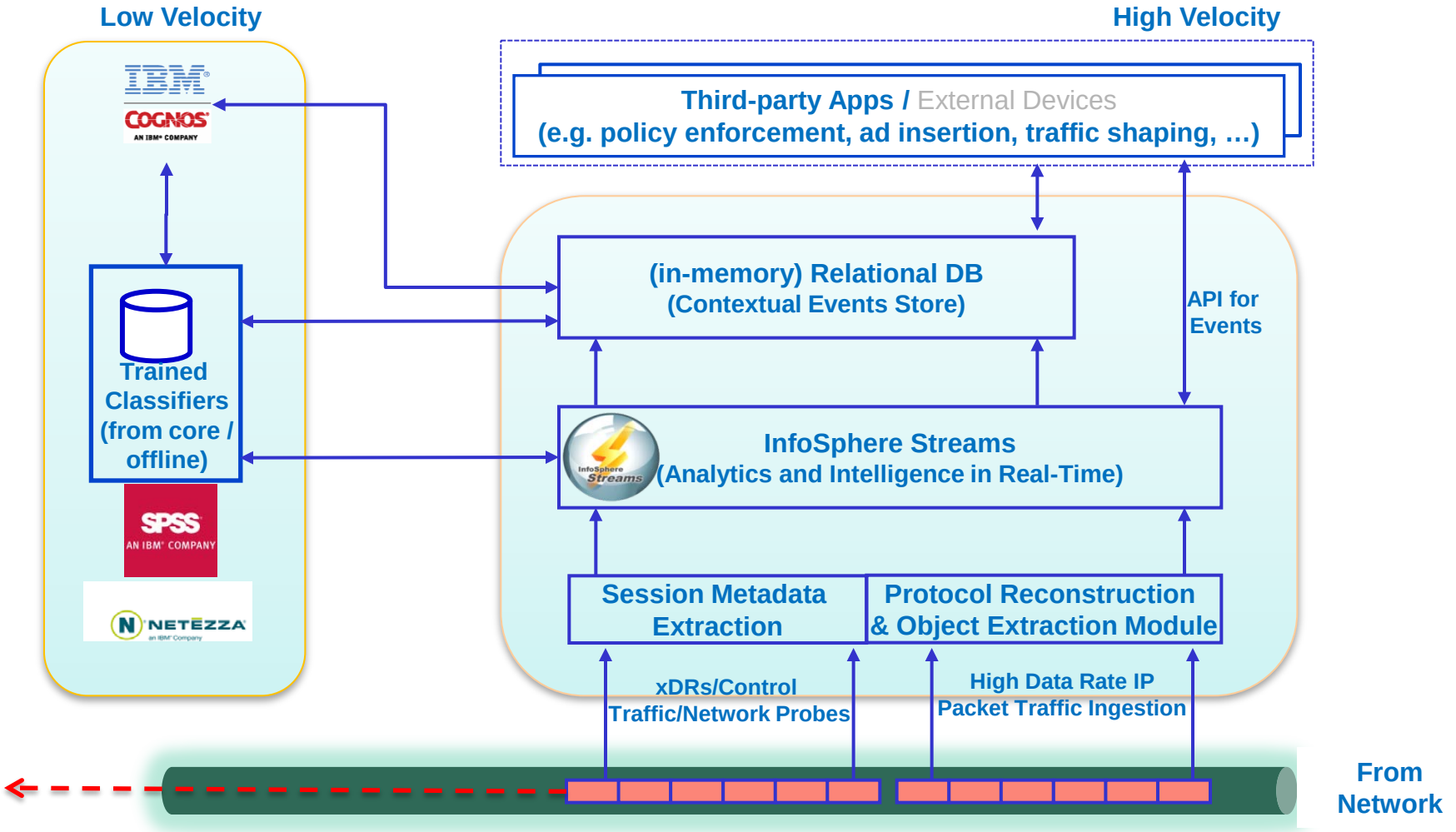


Performance Per Core for Watson Network Trace

Analytics	Throughput (Mbps)
Packet Reassembly	2000.0
1. URL Extraction	1918.9
2. Top-K	1903.1
3. Social Network Synthesis	137.7
4. Network Flow Fingerprinting	88.1

DEEPER DIVE INTO INFOSPHERE STREAMS

Real-time network analytics platform architecture



- ***High Volume* of data:** faster than a database can handle
- ***Complex Analytics*:** correlation from multiple sources and/or signals
- ***Time Sensitive*:** lower latency than possible with the store-and-process paradigm
- ***Scalability*:** scale out through multiple cores/machines for processing

What is InfoSphere Streams?

- InfoSphere Streams is an
  - Extremely scalable platform to run powerful real time analytics (RTAP)… *on*
  - Incredible volumes and variety of streaming data.. *with*
  - Sub-millisecond latency and response time*.. while*
  - **Data is still in motion!!!**

Key Advantages of Using Streams

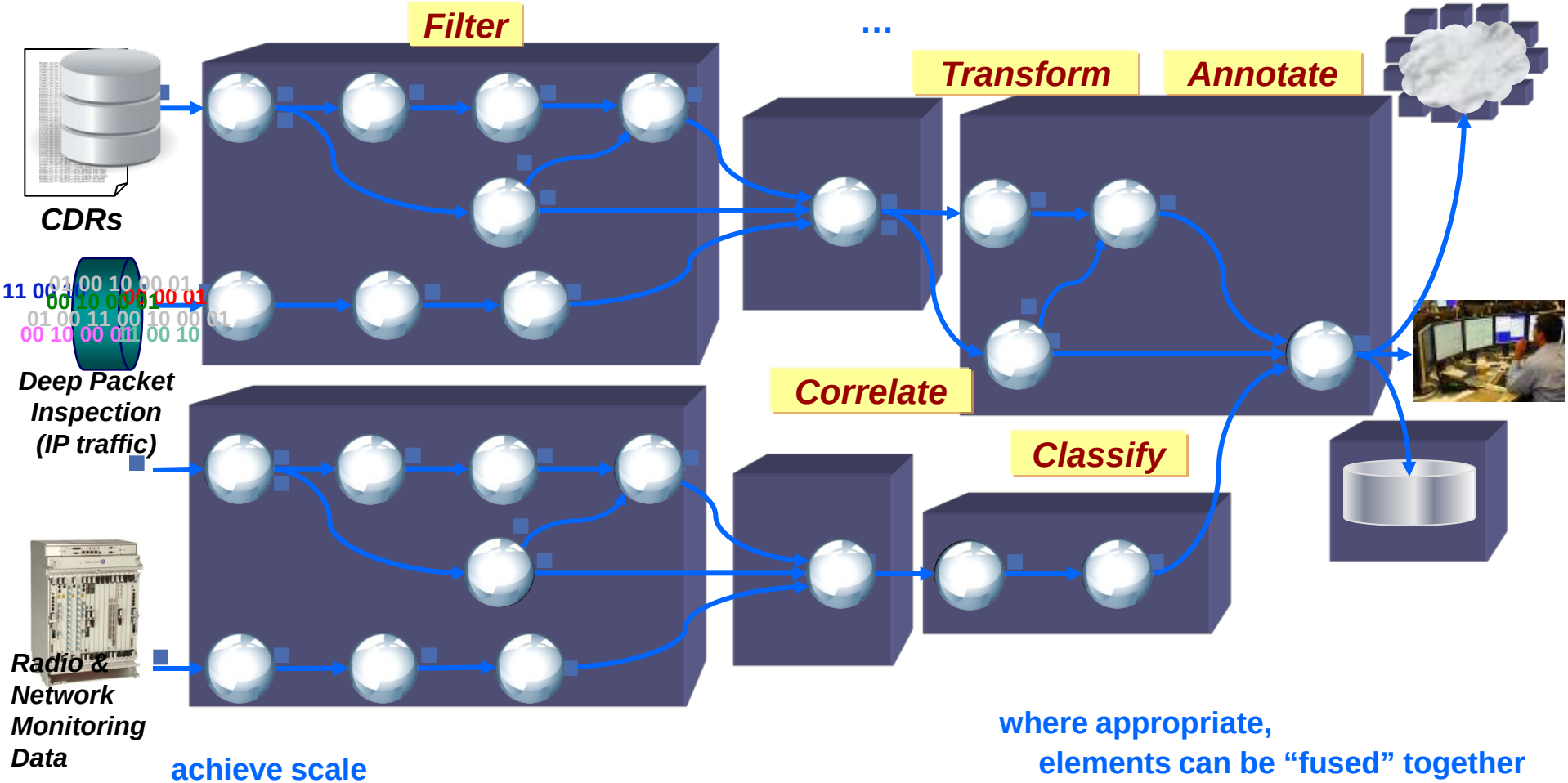
- Flexibility
  - Perform different kinds of analysis on the network data.
    - like processing text in different languages, specialized processing for key sites like facebook, youtube, etc, advanced text mining, image recognition, speech to text (in some languages), etc .
  - Highly customizable platform
  - Can integrate any number external analytics
  - Can also integrate with data mining products (like Warehouses and SPSS)
    - Allows analyzing historical data and correlating it with real-time data

- Scalability
  - Is linearly scalable (more machines can give more throughput)
  - Can process 30 Gbps on a cluster of about 10 machines

→ continuous ingestion

→ continuous analysis

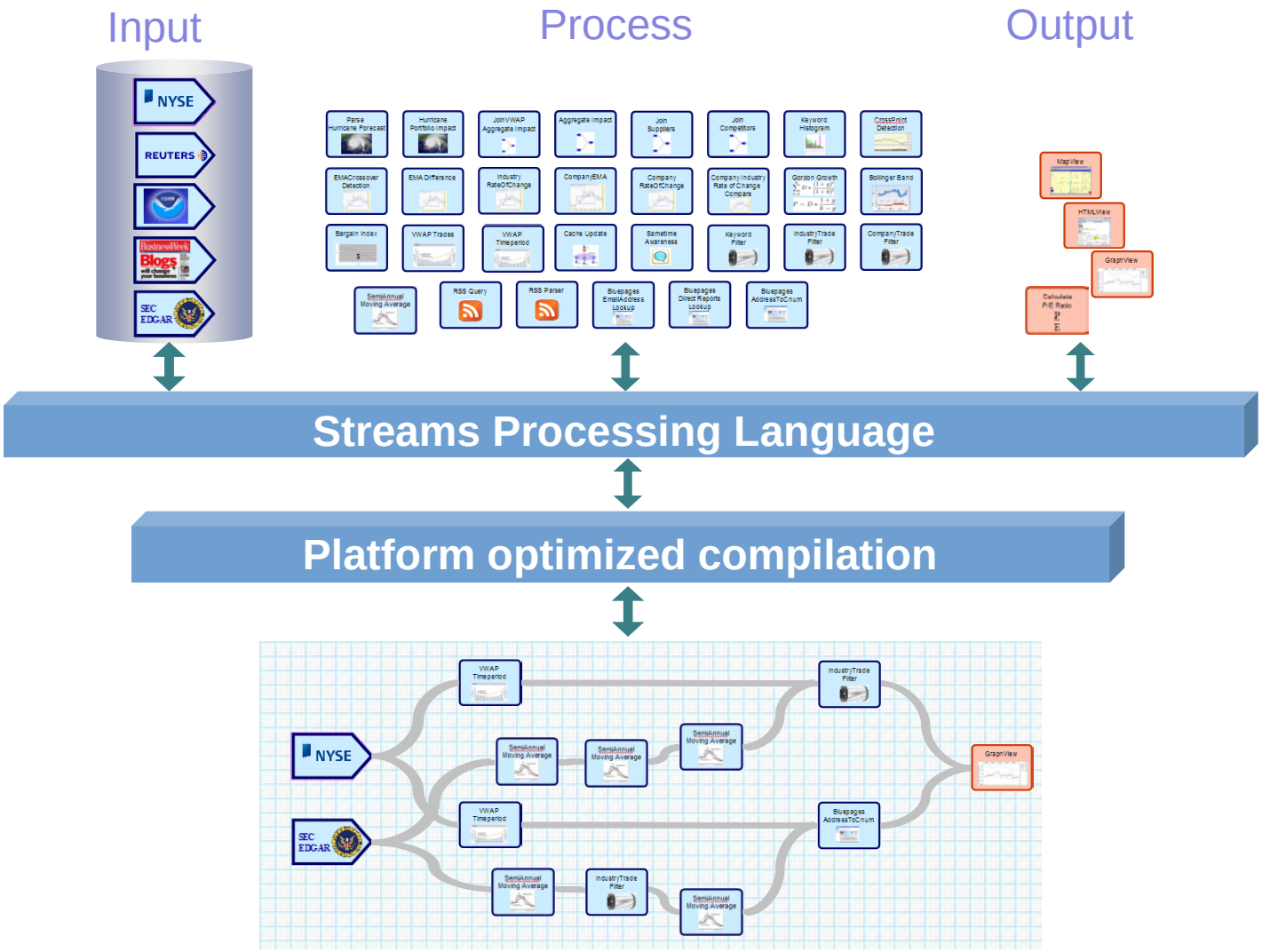
infrastructure provides services for
scheduling analytics across h/w nodes
establishing streaming connectivity
…



achieve scale
by partitioning applications into components
by distributing across stream-connected hardware nodes

where appropriate,
elements can be "fused" together
for lower communication latencies

Streams Programming Model



Toolkits and operators to speed and simplify development

Standard Toolkit

Relational Operators

Filter	Sort
Functor	Join
Punctor	Aggregate

Adapter Operators

FileSource	UDPSource
FileSink	UDPSink
DirectoryScan	Export
TCPSource	Import
TCPSink	MetricsSink

Utility Operators

Custom	Split
Beacon	DeDuplicate
Throttle	Union
Delay	ThreadedSplit
Barrier	DynamicFilter
Pair	Gate
JavaOp	

Compatibility Operators

V1TCPSource	V1TCPSink

Internet Toolkit

InetSource

HTTP	FTP
HTTPS	FTPS
RSS	file

Database Toolkit

ODBCAppend	ODBCEnrich
ODBCSource	SolidDBENrich

- Financial Toolkit
- Data Mining Toolkit
- Time Series Toolkit
- User Defined Toolkits
- And more…

GEOTAGGING OF UNSTRUCTURED DATA

Inference of Spatio-temporal Tags from Unstructured Text

Location	Tips
New York Penn Station	The **seating** area says **Acela express ticket** holders only but that's just for mornings
	Really big **station**, but they don't announce tee **train track** till few min before it **boards**. Not a lot of customer service in there either.
	Instead of waiting on line for your **Amtrak train**, take the stairs directly to the platform from the **NJT** level below.
Metropolitan Museum of Art	Take the elevator in the **European sculpture** and decorative **arts gallery** up to the top and grab a drink at the **roof garden cafe** and **martini bar** (open fro May through the fall)
	Everyone knows The **Met** is the city's most epic **museum**, with a vast collection from **ancient** to **modern**. dont' have to tell you that it is a must see. I love to twirl around the period rooms alone.
	It's tricky to navigate, and overwhelmingly humongous, but that's all part of the **Met**'s charm. We love losing ourselves in the miles of corridors and ogline over the many world famous **treasures**.
Magnolia Bakery	Known for their butter-cream **cupcakes** and floral decor, it's a lovely place to grab one or two **desserts** for after dinner.
	Whoopie cookie is the freaking best thing I've ever tasted. Forget the **cupcakes**! They are too sweet, make sure u have water if u eat them
	Get the **red velvet mini cheesecake**, the **lemon bar**, and their **banana pudding**. Thank me later!

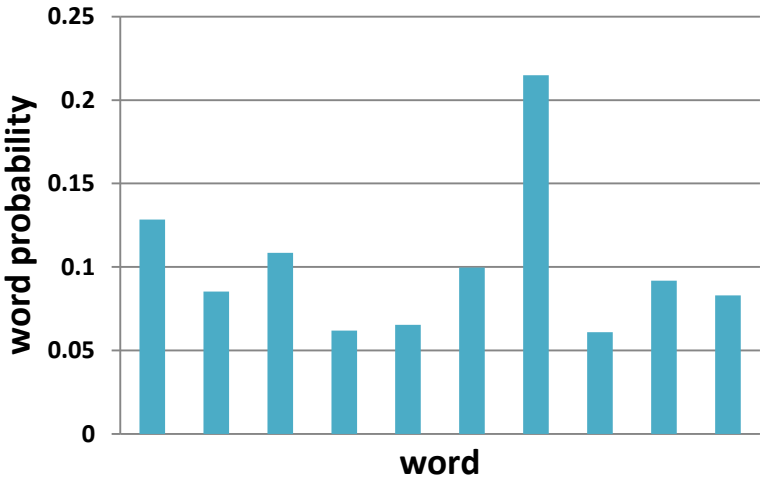
Contributions

- **Data collection** utility for crawling FourSquare data (a popular social network providing *check-in* and *tip* information for popular venues)
- **Feasibility study** for spatio-temporal tagging from unstructured text
  - Language models for unstructured text
  - Quantifiable metrics for spatio-temporal information content in unstructured text
- Algorithms for **inferring spatio-temporal** tags from **unstructured text**
  - A supervised classification approach
  - Evaluation using FourSquare and Twitter datasets
- **Code availability**
  - All the above code in Java
  - In addition, simple utilities for filtering and data cleaning (e.g., stemmer, stop-words, frequency, short-URLs)
- **ns-CTA funding** for FY2013 to infer spatial and temporal attributes in unstructured text

Language Models

- Can we build models from unstructured text at specific locations?
- Build LM for a given location based on the unstructured text at that location
  - Handling unstructured texts (e.g. “The seating area says Acela express ticket holders only”)
    - Tokenizing => [the] [seating] [area] [says] [acela] [express] [ticket] [holders] [only]
    - Removing stop words => [seating] [area] [acela] [express] [ticket] [holders]
    - Stemming => [seat] [area] [acela] [express] [ticket] [holder]
  - Select only locations with more than minimum amount of unstructured texts
  - Consider only commonly used words for that location

word	frequency
wait	98
ticket	56
line	59
amtrak	55
penn	83
…	…



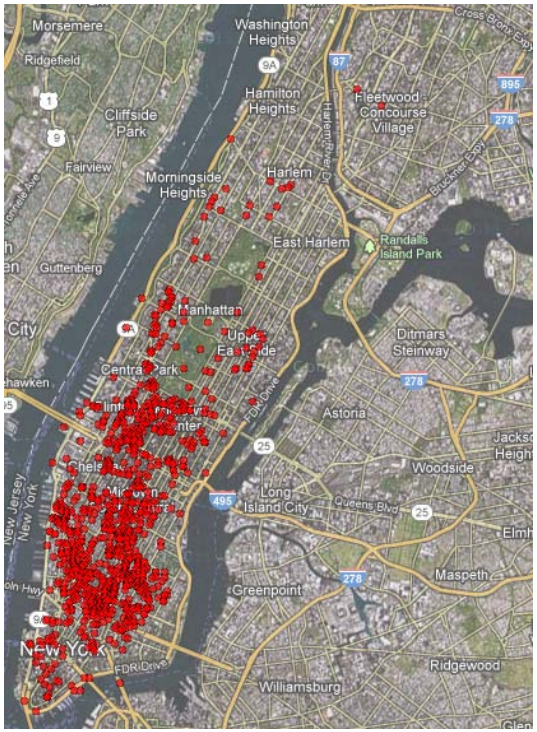
Algorithms for Geotagging: High Level Approach

- Step 1: Filtering out general “I don’t know” unstructured texts using heuristics
  - Ignoring unstructured texts having no clue for their location
  - If a text doesn’t have any local keyword, we classify the text as a “I don’t know” text
    - e.g. “**This sun is BLAZING and there's no shade.**”
- Step 2: Predict the location of selected unstructured texts
  - By ranking locations based on generated LMs and a given unstructured text
  - Use of tf*idf and perplexity metrics to predict locations (from unstructured text)
- Step 3: Differentiating the referred location (LW) of unstructured texts and their physical location (LP), examples:
  - Example 1)
    - “**I hope you all have a GREAT weekend but also take time to remember those we've lost; those who are still fighting for our freedom!!**”
    - Referred location: World Trade Center
  - Example 2)
    - **“Let's Go Yankees!!!”**
    - Referred location: Yankees Stadium

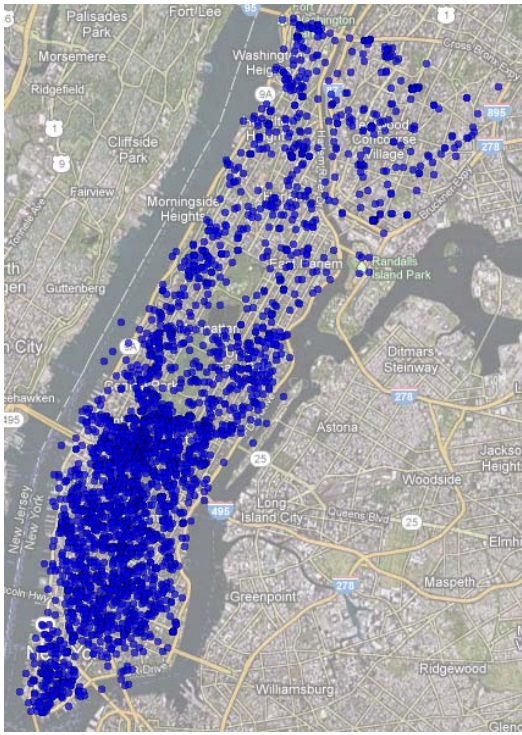
Datasets Collected

- Foursquare Data
  - Collected by our FourSquare crawler
  - About 400,000 tips across about 55,000 locations in NYC collected over 4 years
  - 1,066 locations with more than 50 tips in Manhattan
- Twitter Data
  - Collected by InfoSphere Streams using the GNIP decahose (10%) feed
  - About 4million tweets per day from NYC, 400k part of the decahose feed
  - A total of 109,074 tweets in NYC (June 2012) geo-tagged
    - 40,624 tweets in Manhattan

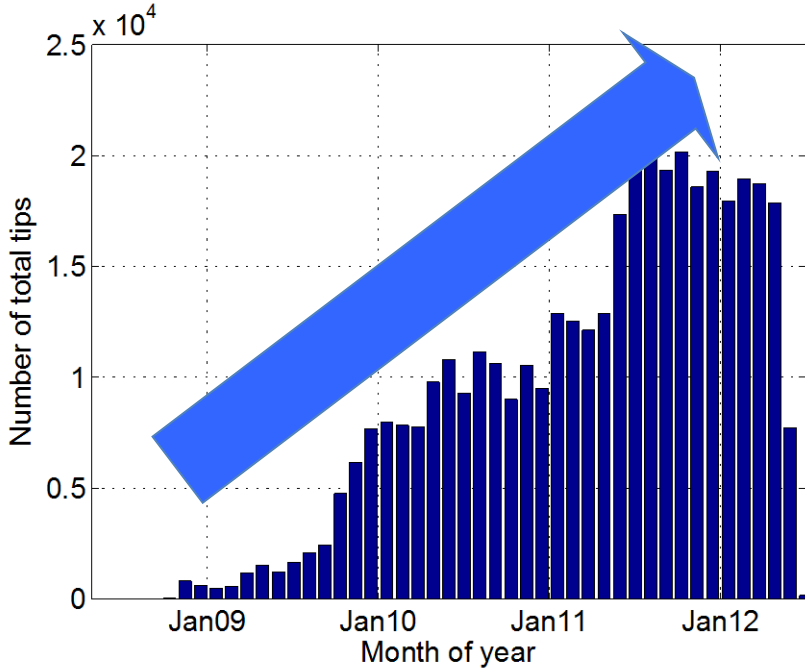
FourSquare

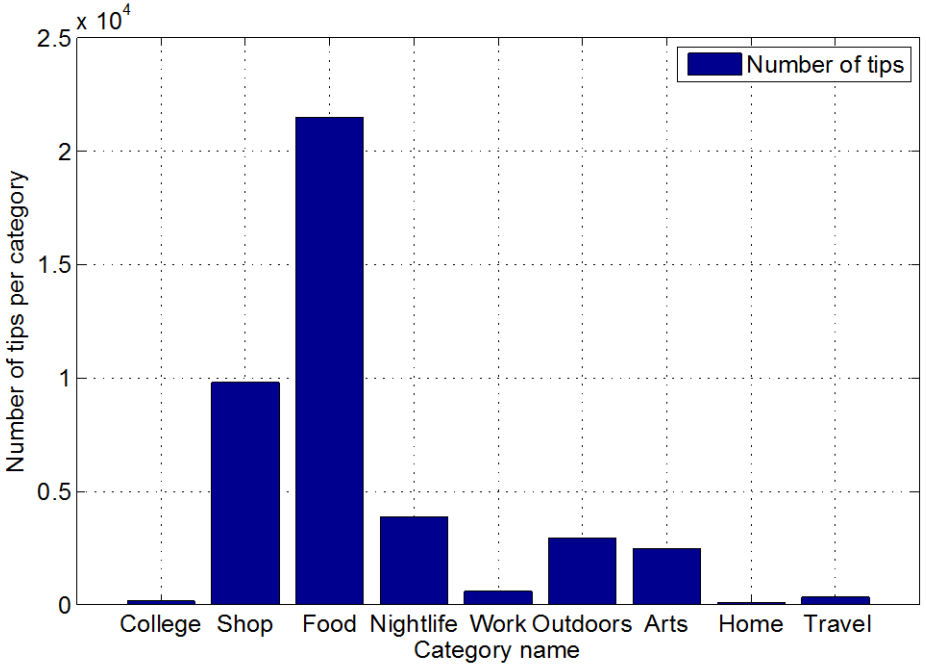


Twitter



FourSquare Data: Details



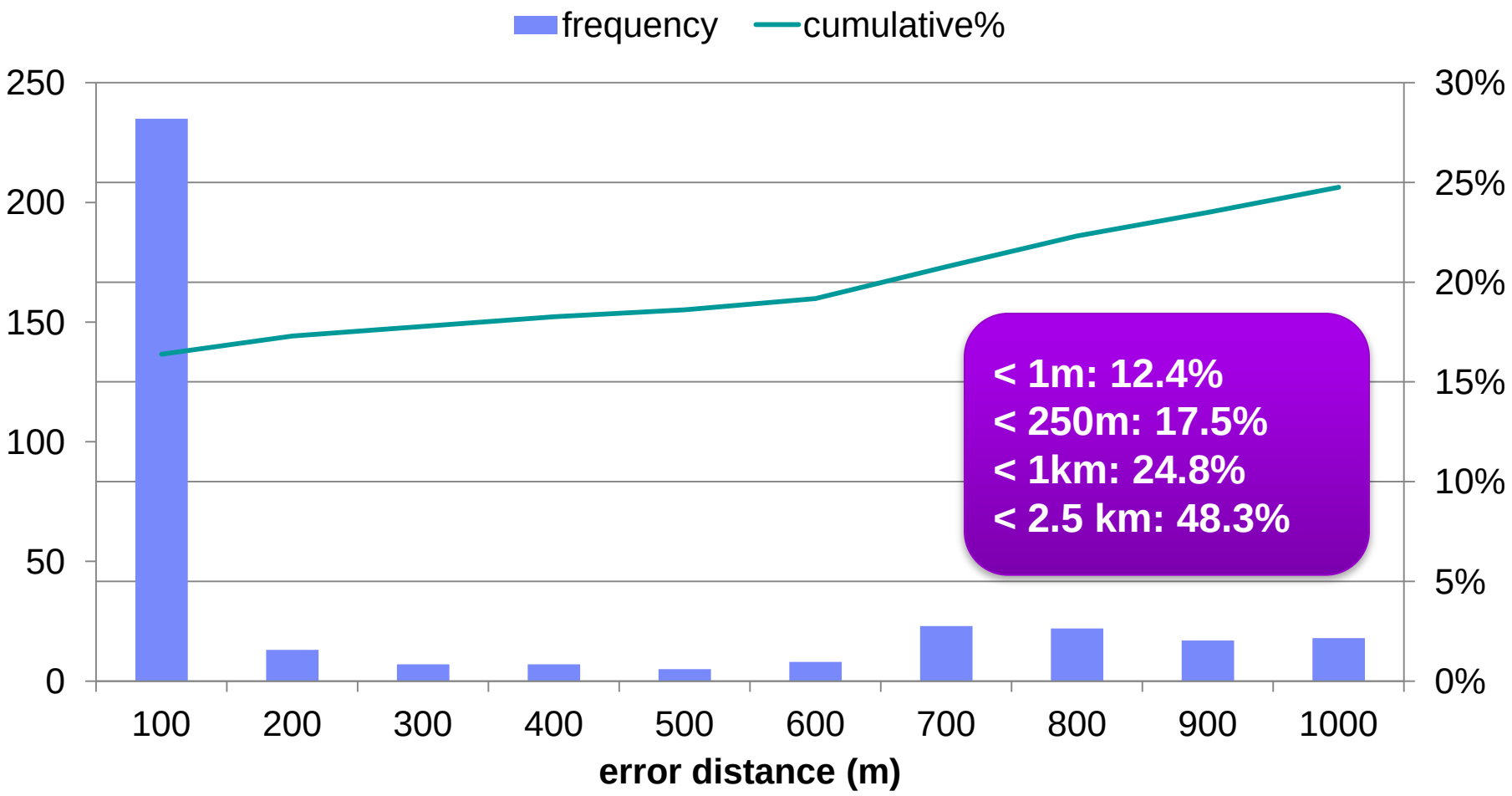


An example of explosion in spatio-temporal data

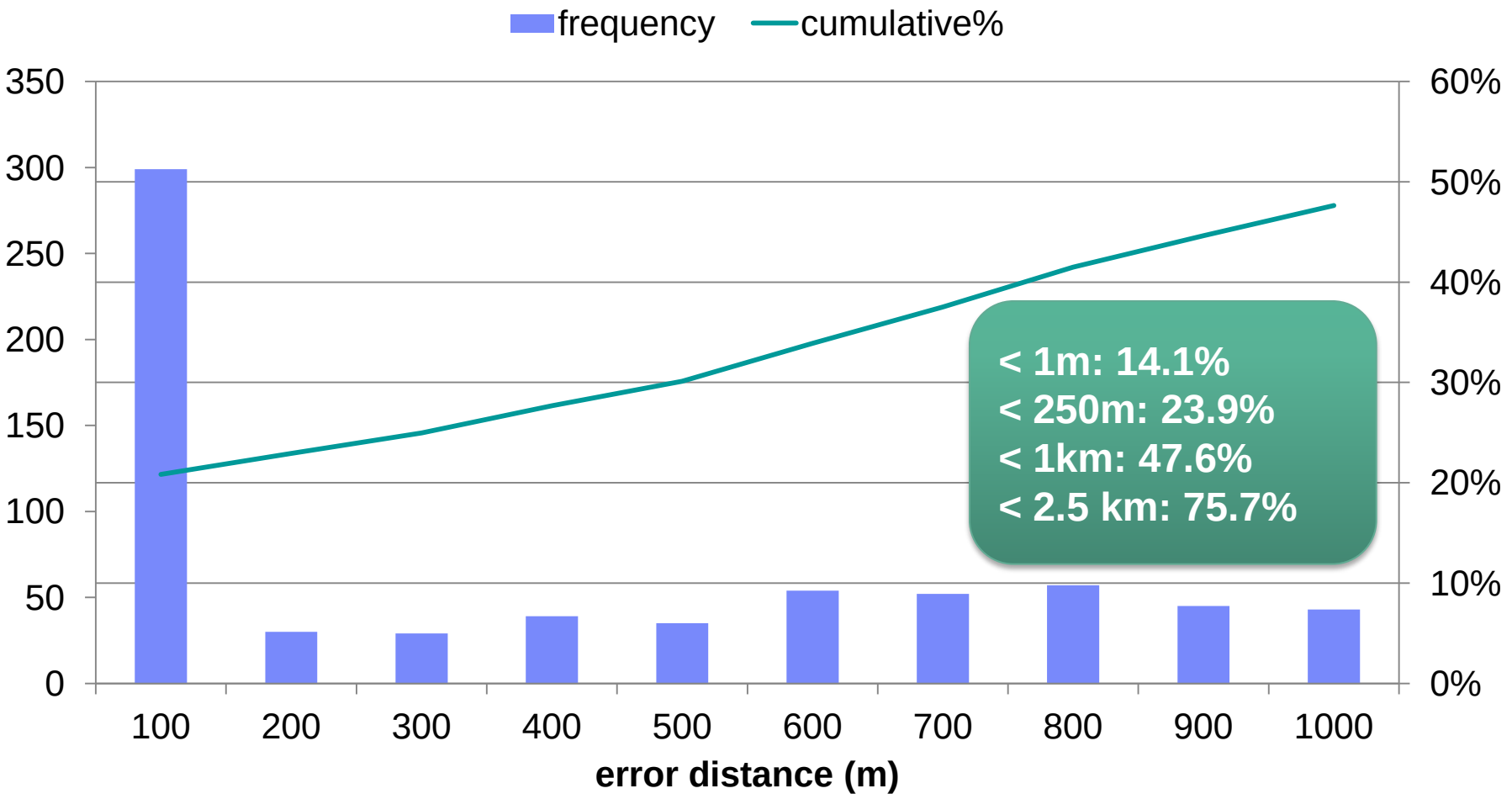
Experimental Settings

- Classifying target tweets
  - Filtering out Foursquare and Instagram tweets
    - They have explicit location name in their text
  - **15,158 tweets** are selected as target tweets
- Filtering out "I don't know" tweets
  - 275 past and 32 future tweets are filtered out
  - 384 tweets having no valid word are filtered out
- We predict the location of **1,434 tweets** for evaluation purposes
  - Results for prediction for top 1 and top 5 locations

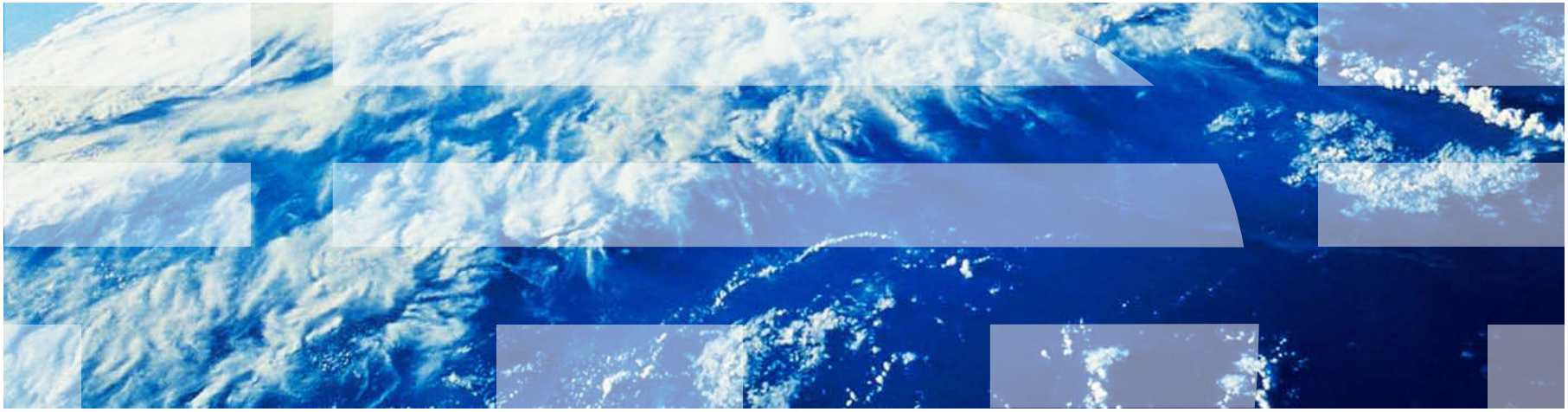
Distance between top 1 prediction and the actual location



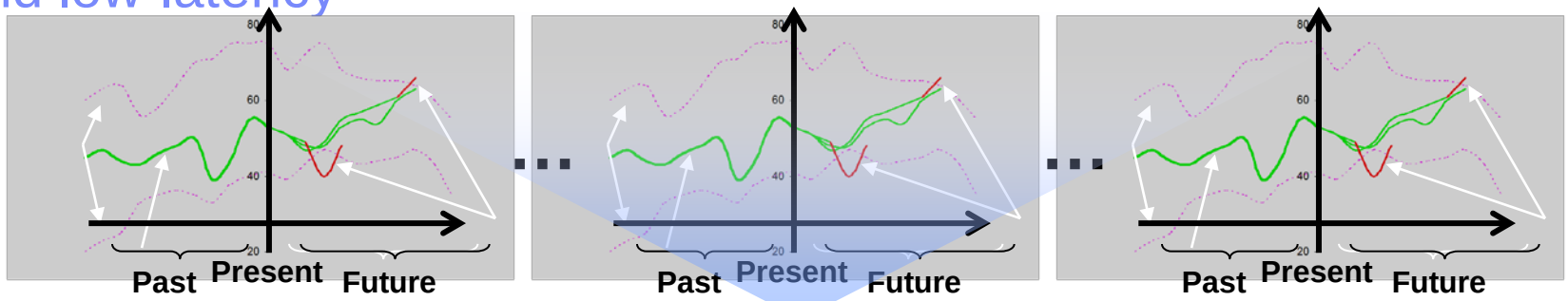
Distance between the best of top 5 predictions and the actual location



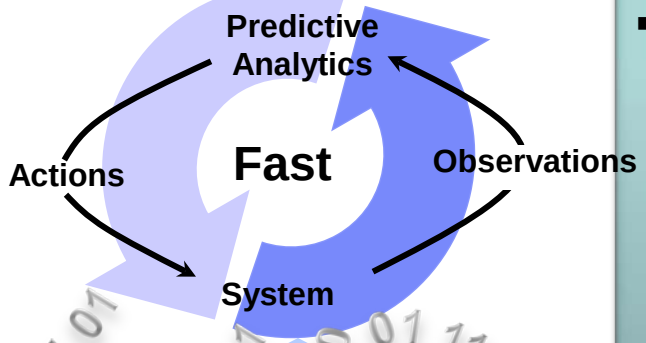
Time Series Data Mining



Predictive analysis of time series data-in-motion at high-throughput and low-latency



- Time-stamped data measurements
  - Variety of sources
  - **High-throughput**
  - **Low-latency**



- Sample Questions:
  - What are realistic baselines for my environment?
  - How do you characterize aberrant behavior?
  - Can you provide an early warning for an outage?
  - What will the resource consumption be in 15 mins?

5.9 billion mobile devices in 2012

Radio & Network Monitoring Data

2 billion Internet users by 2011

30 billion RFID tags in 2010

Streaming Predictive Analytics

Challenge

- *Enable proactive ("look-ahead") analytics in IBM's portfolio of performance management products for processing data-in-motion at high-throughput and low-latency*

Solution

- *Lightweight software library with online statistical time series analysis algorithms*
- *Designed for tight integration with IBM products for high-performance analysis*

Benefits

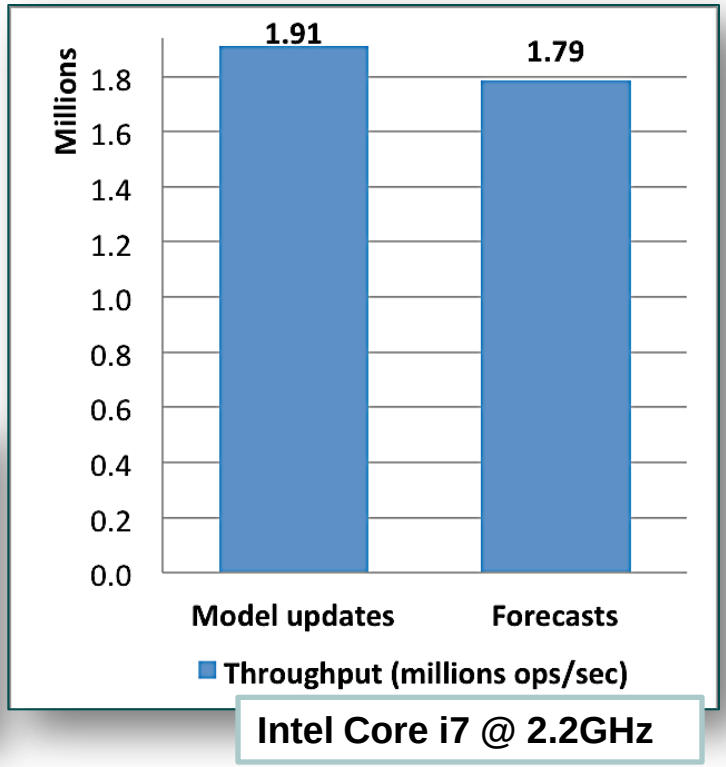
- Provides proactive intelligence for responses to business, network and system events, in real-time
- Minimizes effort required to enhance products and services with predictive capabilities

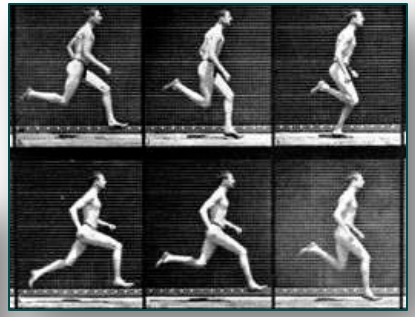
Class	Features/Models
Linear Modeling	• **ARIMA / ARMA** • **Linear Regression** • **Moving Averaging**
Seasonal + Trend Modeling	• **Holt-Winters Additive** • **Holt-Winters Multiplicative** • **Segmented Models** • **Seasonal-Trend Decomposition**
Automatic Learning	• **Grid search (for H-W)** • **Maximum Likelihood Estimation*** • **Automatic Model Selection***
Data Transformation Framework	• **Logarithmic, Shifting, Differencing, etc.**
Auxiliary Functions	• **Automatic seasonality detection, MA Filtering, FFT, PACF, etc.**
Error Framework	• **MAPE, MSE, MASE, AIC***
APIs	• **Java / C++ / Streams / REST+JSON** • **Model Repositories**

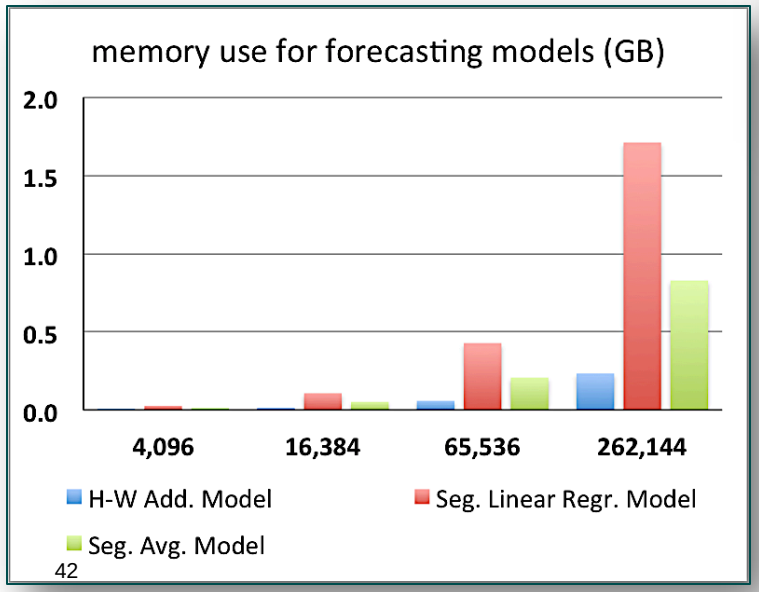
***Under development**

Technical features: scalability for analyzing data at high-throughput

- **Incremental model updates** when new data is made available
  - Eliminate need to access/read historical data
- **Scalable** to a large number of time series



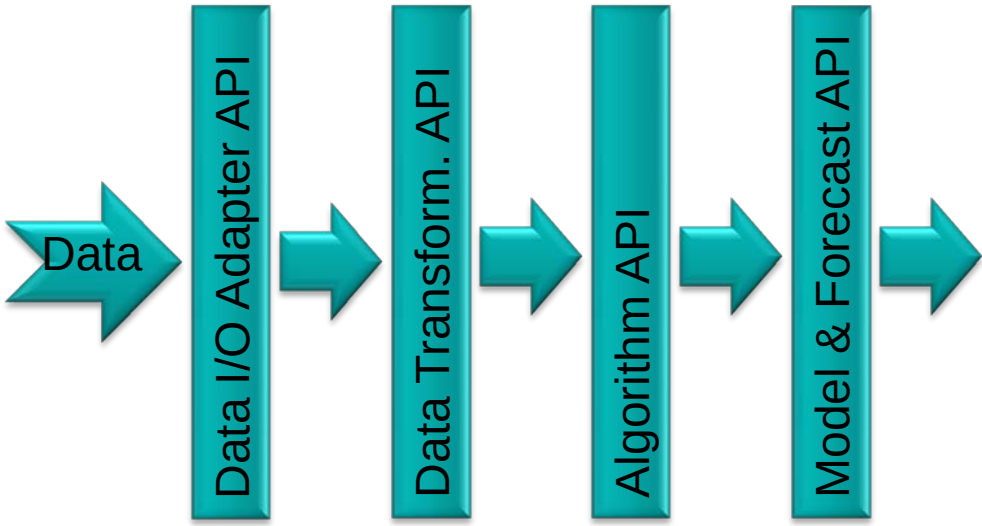




- **Compact representation** of time series: only model definition and data-transformations are kept in memory
- **Models** can be **persisted** onto storage for later retrieval, once new data is available for updates

Technical features: modularity and embedability (contd.)

- **Modularity**:
  - Data-agnostic design: modeling does not depend on source or type of data
  - De-coupling of data collection and pre-processing stages from model updates & predictions




```
public static void main(String[] args) throws PMException {
    int maxPeriod = 4; int seasons = 2;
    long[] timestamps = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11
    double[] samples = { 0.0, 10.0, 1.2, -10, -2.5, 10.0, 3.7,

    ITimeseries ts = new Timeseries(TimeUnits.Seconds, timestam
    // Create model
    IForecastingModel fm = new ForecastingModel(new HWSAdditive
    // Initialize and update model with data
    fm.updateModel(ts);
    // Create forecasts
    double forecast = fm.forecastAt(20);

}
```

Questions?

Thank you!

Contact: Dakshi Agrawal / agrawal@us.ibm.com