MacroBase: A Search Engine for Fast Data Streams

Sahaana Suri
Firas Abuzaid, Peter Bailis, Jialin Ding, Edward Gan, Kexin Rong
The Team

Peter Bailis  
Professor

Edward Gan  
3rd year Ph.D.

Kexin Rong  
3rd year Ph.D.

Sahaana Suri  
3rd year Ph.D.

Firas Abuzaid  
3rd year Ph.D.

Jialin Ding  
4th year B.S.

macrobase@cs.stanford.edu
Reduced storage costs due to Big Data systems (e.g., HDFS, S3, Kafka), cloud

Need to monitor complex applications relying on sensors, processes, production telemetry
e.g., Microsoft, Facebook, Twitter, LinkedIn collect 12M+ events/sec today

Monitoring & Telemetry Drive Data Volumes

Projected Data Growth % (IDC)
Ability + need to monitor complex applications relying on sensors, processes, production telemetry.

Projected Data Growth % (IDC)

Monitoring & Telemetry Drive Data Volumes
Ability + need to monitor complex applications relying on sensors, processes, production telemetry.

Reduced storage costs due to Big Data systems (e.g., HDFS, S3, Kafka), cloud.
Ability + need to monitor complex applications relying on sensors, processes, production telemetry

Reduced storage costs due to Big Data systems (e.g., HDFS, S3, Kafka), cloud

Microsoft, Facebook, Twitter, LinkedIn collect 12M+ events/sec today
Current Monitoring Pipelines
Current Monitoring Pipelines
Current Monitoring Pipelines

Collect + Process
Current Monitoring Pipelines

Collect + Process → Store → Viz/Dashboards
Current Monitoring Pipelines

Collect + Process → Store → Viz/Dashboards → Rules Alerts Failures
Current Monitoring Pipelines

Collect + Process

Store

Data for Analysis

Viz/Dashboards

Rules
Alerts
Failures
Current Monitoring Pipelines

Collect + Process → Store → Data for Analysis

Viz/Dashboards

Top SV orgs: < 6% data read!

Rules Alerts Failures
$12+ \, m \, \text{events/sec}$
12+ m events/sec
Can we enable more productive, automated analyses to efficiently monitor data and telemetry at scale?
Key Bottleneck in Monitoring: Human Attention

Human attention is scarce!
Infeasible to manually inspect large volumes
In practice: data only accessed for post-hoc root cause analyses
Top SV orgs: < 6% data read
Key Bottleneck in Monitoring: Human Attention

Human attention is scarce! Infeasible to manually inspect large volumes.
Current Monitoring Pipelines

Collect + Process

Store

Viz/Dashboards

Data for Analysis

Rules Alerts Failures
Current Monitoring Pipelines

Collect + Process

Rules
Alerts
Failures

Store

Data for Analysis

Spark
Spark Streaming
ALLUXIO
Apache Storm
Apache HBase
InfluxDB
Prometheus
Grafana
Cassandra
Omniscient
Loggly
Kafka
Beam
Nifi
Key Bottleneck in Monitoring: Human Attention

Human attention is scarce! Infeasible to manually inspect large volumes.

Dataflow engines provide a means of processing this data... but what functions?
Key Bottleneck in Monitoring: Human Attention

Human attention is scarce! Infeasible to manually inspect large volumes

Dataflow engines provide a means of processing this data... but what functions?

Stats + ML literature provides many, many algorithms, but few implemented or tested at scale
Key Bottleneck in Monitoring: Human Attention

We need a system that composes efficient, higher-level statistical operators

Human attention is scarce! Infeasible to manually inspect large volumes

Dataflow engines provide a means of processing this data... but what functions?

Stats + ML literature provides many, many algorithms, but few implemented or tested at scale
Monitoring with MacroBase

1. Collect + Process
2. Store
3. Viz/Dashboards
4. Rules Alerts Failures
5. MacroBase

Diagram showing the process flow of monitoring with MacroBase.
MacroBase is a monitoring engine that prioritizes user attention by combining classification and explanation with streaming dataflow.

This talk: Share our goals, architecture, early results
Outline

Prioritizing Attention in Fast Data

MacroBase Demo

MacroBase Architecture + Usage

Recent Work

Future + Conclusion
Demo: Scenario
Demo: Scenario
Demo: UI Recap

Input data

Database Configuration

<table>
<thead>
<tr>
<th>Database URL</th>
<th>localhost</th>
<th>submit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base query</td>
<td>csv://core/demo/mobile_data.csv</td>
<td>submit</td>
</tr>
</tbody>
</table>
Demo: UI Recap

Input data

Select metrics
Demo: UI Recap

Input data

Select metrics

Select attributes
Demo: UI Recap

Input data
Select metrics
Select attributes
Explore results
Case Study: CMT

Cambridge Mobile Telematics:
Monitors driving behavior via mobile application available for smartphones
Case Study: CMT

Cambridge Mobile Telematics:
Monitors driving behavior via mobile application available for smartphones

Question: Is the application behaving correctly on every platform?
Case Study: CMT

Cambridge Mobile Telematics:
Monitors driving behavior via mobile application available for smartphones

Question: Is the application behaving correctly on every platform?

Challenge: Spending even 1 second per combination requires 7 days

25 Major API Releases
Over 24K Android device types
Case Study: CMT

Cambridge Mobile Telematics:
Monitors driving behavior via mobile application available for smartphones

Question: Is the application behaving correctly on every platform?

Challenge: Spending even 1 second per combination requires 7 days

“IOS 9.0 beta 1–5 (but not 9.0.1) had a buggy BLE stack that prevented iOS devices from connecting to devices.”
Outline

Attention Prioritization in Fast Data
MacroBase Demo
MacroBase Architecture + Usage
Recent Work
Future + Conclusions
MacroBase Architecture: Operator Cascades

Execute operator cascades to transform, segment, aggregate streams
MacroBase Architecture: Operator Cascades

Execute operator cascades to transform, segment, aggregate streams
MacroBase Architecture: Operator Cascades

Execute operator cascades to transform, segment, aggregate streams
MacroBase Architecture: Operator Cascades

Execute operator cascades to transform, segment, aggregate streams
Transformation

Feature extraction, dimensionality reduction, streaming ETL
Transformation

Optional

e.g., time series dimensionality reduction (via FFT, PCA)

e.g., image-specific features (e.g., hue and luminosity)
Transformation

Optional

Domain-specific data preprocessing

e.g., time series dimensionality reduction (via FFT, PCA)

e.g., image-specific features (e.g., hue and luminosity)
Transformation

Optional

Domain-specific data preprocessing

Combine and chain many transformations to build complex features
Classification

Segmentation, rule evaluation, data filtering, inference
Classification

Segment and filter stream by target behavior (e.g., abnormalities)

More than $k$ standard deviations from $\mu$
Classification

Segment and filter stream by target behavior (e.g., abnormalities)

Default: identify unlikely data points (e.g., via density estimation)
Classification

Segment and filter stream by target behavior (e.g., abnormalities)

Default: identify unlikely data points (e.g., via density estimation)

Combine with thresholds, predicates, or custom classifiers
Explanation

Aggregation to interpretable outputs via difference detection
Explain classification results by identifying behavior correlated with being filtered.

<table>
<thead>
<tr>
<th>Errors</th>
<th>Non-Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>{iPhone7, Canada}</td>
<td>{iPhone8, USA}</td>
</tr>
<tr>
<td>{iPhone7, USA}</td>
<td>{iPhone7, USA}</td>
</tr>
<tr>
<td>{iPhone8, Canada}</td>
<td>{iPhoneX, USA}</td>
</tr>
<tr>
<td>{iPhone7, USA}</td>
<td>{iPhone7, USA}</td>
</tr>
<tr>
<td>{iPhone8, Canada}</td>
<td>{iPhone7, USA}</td>
</tr>
<tr>
<td>Canada may have a problem!</td>
<td>{iPhone8, USA}</td>
</tr>
<tr>
<td></td>
<td>{iPhone7, USA}</td>
</tr>
<tr>
<td></td>
<td>{iPhone8, USA}</td>
</tr>
<tr>
<td></td>
<td>{iPhone7, USA}</td>
</tr>
</tbody>
</table>
**Explaination**

Relative Risk

\[
P(\text{error} | \text{canada}) \quad \frac{\text{P(error | canada)}}{\text{P(error | not canada)}}
\]

Explain classification results by identifying behavior correlated with being filtered

**Default:** relative risk calculation based on data attributes
MacroBase Architecture: Operator Cascades

Execute operator cascades to transform, segment, aggregate streams
Usage
# Usage

## Basic

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Point and Click</strong></td>
<td>firmware_version</td>
</tr>
<tr>
<td></td>
<td>model</td>
</tr>
<tr>
<td></td>
<td>power_drain</td>
</tr>
</tbody>
</table>

## Web Interface

- **Web Browser**: View and interact with the interface.
- **Script, Stream**: Import data and commands through scripts or streams.
**Usage**

### Basic
- **Point and Click**
  - firmware_version: varchar
  - model: varchar
  - power_drain: numeric

### Intermediate
- Custom Pipeline Config
- `new LinearMetricNormalizer()
  .then(new MBGroupBy(groupByIndex,
  () -> new FeatureTransform(conf)))
  .then(new BatchingPercentileClassifier(conf))
  .then(new BatchSummarizer(conf))
  .consume(conf.constructIngester().getStream().drain());`

### Web Interface
- Web Browser
- Dataflow Pipeline
- Script, Stream
Usage

**Basic**

<table>
<thead>
<tr>
<th>Point and Click</th>
</tr>
</thead>
<tbody>
<tr>
<td>☑</td>
</tr>
<tr>
<td>▲</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>firmware_version</th>
<th>varchar</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>varchar</td>
</tr>
<tr>
<td>power_drain</td>
<td>numeric</td>
</tr>
</tbody>
</table>

**Intermediate**

```java
new LinearMetricNormalizer()
    .then(new MBGroupBy(groupByIndex, () -> new FeatureTransform(conf)))
    .then(new BatchingPercentileClassifier(conf))
    .then(new BatchSummarizer(conf))
    .consume(conf.constructIngester().getStream().drain());
```

**Advanced**

```java/c++
int k = data.get(0).metrics().getDimension();
int n = data.size();
List<double[]> metrics = new ArrayList<>(n);
for (Datum curDatum : data)
    metrics.add(curDatum.metrics().toArray());
List<double[]> trimmedMetrics = trimmer.process(metrics);
gModel = new Gaussian().fit(trimmedMetrics);
```
Usage—JSON Rest API

```json
{
  "inputURI": ..., 
  "metric": "Percentile Dropped records",
  "classifier": "quantile",
  "cutoff": 1.0,
  "includeHi": true,
  "includeLo": false,

  "minSupport": 0.005,
  "minRatioMetric": 1.5
}
```
# classify defined by user or default classification operators

WITH SELECT *, classify(metric) as label FROM mobile_data AS labeled
SELECT * FROM
MACRODIFF
SELECT * FROM labeled where label = 1 as outlier,
SELECT * FROM labeled where label = 0 as inlier
ON
device, make, model, country, version
COMPARE BY
    RISKRATIO(COUNT(*)) AS rr
WHERE rr > 3 AND SUPPORT > 0.25
LIMIT 25;
Usage—SQL Coming Soon

```sql
# classify defined by user or default classification operators
WITH SELECT *
    , classify(metric) as label
FROM mobile_data AS labeled

SELECT * FROM MACRODIFF

SELECT * FROM labeled where label = 1 as outlier,
SELECT * FROM labeled where label = 0 as inlier

ON device, make, model, country, version

COMPARE BY
    RISKRATIO(COUNT(*)) AS rr
WHERE rr > 3 AND SUPPORT > 0.25
LIMIT 25;
```

Input
# classify defined by user or default classification operators

WITH SELECT *, classify(metric) as label FROM mobile_data AS labeled
SELECT * FROM
MACRODIFF
SELECT * FROM labeled where label = 1 as outlier,
SELECT * FROM labeled where label = 0 as inlier
ON
device, make, model, country, version
COMPARE BY
    RISKRATIO(COUNT(*)) AS rr
WHERE rr > 3 AND SUPPORT > 0.25
LIMIT 25;
Usage—SQL Coming Soon

# classify defined by user or default classification operators

WITH SELECT *, classify(metric) as label FROM mobile_data AS labeled

SELECT * FROM

MACRODIFF

SELECT * FROM labeled where label = 1 as outlier,

SELECT * FROM labeled where label = 0 as inlier

ON

device, make, model, country, version

COMPARE BY

  RISKRATIO(COUNT(*)) AS rr

WHERE rr > 3 AND SUPPORT > 0.25

LIMIT 25;
Outline

Attention Prioritization in Fast Data
MacroBase Demo
MacroBase Architecture + Usage
Recent Work
Future + Conclusion
Online Progress Estimation in Dimensionality Reduction

Principal Component Analysis

Core dimensionality reduction operator for many applications

[Suri and Bailis, arXiv 2017]
Online Progress Estimation in Dimensionality Reduction

Principal Component Analysis
Core dimensionality reduction operator for many applications
Out-of-the-box implementations are extremely slow
$O(\min[mn^2,nm^2])$ via singular value decomposition

[Suri and Bailis, arXiv 2017]
Online Progress Estimation in Dimensionality Reduction

Principal Component Analysis

Core dimensionality reduction operator for many applications

Out-of-the-box implementations are extremely slow

$O(\min[mn^2,nm^2])$ via singular value decomposition

Two insights enable significantly faster performance in practice even with naïve PCA implementations

[Suri and Bailis, arXiv 2017]
Online Progress Estimation in Dimensionality Reduction

Data sources are structured; sample prior to model

Variable Star Brightness

Fan Power Consumption

[Suri and Bailis, arXiv 2017]
Online Progress Estimation in Dimensionality Reduction

Data sources are structured; sample prior to model

Dimensionality reduction is a preprocessing step; sample until too expensive

[Suri and Bailis, arXiv 2017]
Online Progress Estimation in Dimensionality Reduction

Data sources are structured; sample prior to model.

Dimensionality reduction is a preprocessing step; sample until too expensive.

50x speedup in dimensionality reduction, and 33x speedup in end-to-end pipelines compared to PCA via SVD.

[Suri and Bailis, arXiv 2017]
Predicate Pushdown in Density Estimation

Kernel Density Estimation
Each point contributes a small “kernel”
Asymptotically optimal estimation

[Gan and Bailis, SIGMOD 2017]
Predicate Pushdown in Density Estimation

Kernel Density Estimation
Each point contributes a small “kernel”
Asymptotically optimal estimation

[ Gan and Bailis, SIGMOD 2017 ]
Predicate Pushdown in Density Estimation

Kernel Density Estimation
Each point contributes a small “kernel”
Asymptotically optimal estimation

Compute density: $O(n^2)$
500K points: 2 hours on 2.4GHz CPU!

[Gan and Bailis, SIGMOD 2017]
Predicate Pushdown in Density Estimation

Kernel Density Estimation
Each point contributes a small “kernel”
Asymptotically optimal estimation
Compute density: $O(n^2)$
500K points: 2 hours on 2.4GHz CPU!
Can we do better?

[Gan and Bailis, SIGMOD 2017]
Predicate Pushdown in Density Estimation

Classification: only need to tell whether above or below target

Need not compute exact densities!

[Gan and Bailis, SIGMOD 2017]
Predicate Pushdown in Density Estimation

Classification: only need to tell whether above or below target

Need not compute exact densities!

Use branch and bound: 2 orders of magnitude speedup

[Gan and Bailis, SIGMOD 2017]
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

Raw time series are hard to read

Original: noisy

[Rong and Bailis, VLDB 2017]
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

Raw time series are hard to read; Smoothing can help!

Original: noisy

Good: retains “outlyingness”

[Rong and Bailis, VLDB 2017]
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

Raw time series are hard to read; Smoothing can help!

Challenge: Automatically choose smoothing parameters

Original: noisy

Good: retains “outlyingness”

[Rong and Bailis, VLDB 2017]
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

Raw time series are hard to read; Smoothing can help!

Challenge: Automatically choose smoothing parameters

Original: noisy

Bad: loses “outlyingness”

Good: retains “outlyingness”

[Rong and Bailis, VLDB 2017]
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

• Formulate as optimization problem: Smooth as much as possible while preserving long-term deviations
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

- Formulate as optimization problem: Smooth as much as possible while preserving long-term deviations

[Fig: NYC Taxi Passengers, Unsmoothed]
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

• Smooth as much as possible while preserving long-term deviations

[Image: NYC Taxi Passengers: Unsmoothed vs. ASAP (this paper)]

[Rong and Bailis, VLDB 2017]
Efficient Parameter Search in
Time Series Visualization

Time Series Smoothing

- Smooth as much as possible while preserving long-term deviations

[Images of NYC Taxi Passengers and Average Temperature in England graphs]

[Rong and Bailis, VLDB 2017]
Efficient Parameter Search in Time Series Visualization

Time Series Smoothing

- Smooth as much as possible while preserving long-term deviations

[Image of NYC Taxi Passengers and Average Temperature in England graphs]

[Rong and Bailis, VLDB 2017]
References

• MacroBase motivation [CIDR 2017]
• MacroBase architecture, sketches [SIGMOD 2017]
• tKDC classification [SIGMOD 2017]
• NoScope video classification [VLDB 2017]
• ASAP time-series presentation [VLDB 2017]
• MIC DROP dimensionality reduction [forthcoming]

Read our blog posts! http://dawn.cs.stanford.edu/blog
Outline

Attention Prioritization in Fast Data
MacroBase Demo
MacroBase Architecture + Usage
Recent Work
Future + Conclusion
Next Steps

• Operating over pre-aggregated data cubes [Edward, Jialin]

• New sketches for difference detection [Kai Sheng]

• Time series similarity search (earthquakes!) [Kexin]

• Distributed MacroBase [Sahaana]

• Temporal difference detection [Firas]
Conclusion

Increasing need for data monitoring demand new tools for prioritizing human attention; dataflow engines are not enough

Our proposal: combine feature extraction, classification, explanation in an end-to-end analytic monitoring engine

https://github.com/stanford-futuredata/macrobase
http://dawn.cs.stanford.edu/blog
DAWN Stack

Data Acquisition
- Snorkel, Babble Labble, Coral
- DeepDive

Feature Engineering
- MacroBase (Streaming Data)
- Data Fusion
- YellowFin (DL)
- NoScope (Video)
- *Headed, Mulligan (SQL+graph+ML)
- AutoRec, SimDex (Recommendation)

Model Training
- AutoML

Productionizing
- ModelQA

Interfaces
- Spark

Algorithms
- Compilers: Weld, Spatial, Sparser, Delite

Systems
- Hardware: Plasticine CGRA, FuzzyBit
- Mobile
- CPU
- GPU
- FPGA
DAWN Stack

Data Acquisition
- Snorkel, Babble Labble, Coral
- DeepDive

Feature Engineering
- MacroBase (Streaming Data)
- Data Fusion

Model Training
- AutoML
- YellowFin (DL)
- NoScope (Video)
- *Headed, Mulligan (SQL+graph+ML)
- AutoRec, SimDex (Recommendation)

Productionizing
- ModelQA

Hardware
- Compilers: Weld, Spatial, Sparser, Delite
- Hardware: Plasticine CGRA, FuzzyBit

Interfaces
- Algorithms
- Systems

...