The RISE Lab: Real-time Intelligent Secure Execution

Krste Asanovic, Ali Ghodsi, Ken Goldberg, Joey Gonzalez, Joe Hellerstein, Michael Jordan, Randy Katz, Dave Patterson, Raluca Ada Popa, Ion Stoica, ...
Berkeley’s lab tradition

• Working for 5-6 years on a new major problem

• Bringing faculty from different areas
AMPLab (2010—2016)

Created popular open-source big data analytics: Spark, Mesos, Tachyon..

AMPLab is coming to an end (December 2016)

What is the next vision?
RISE Lab
From **live data to real-time decisions**

AMP Lab
From **batch data to advanced analytics**
Big Data → Training → Big Model

MLC  TensorFlow  Caffe  VW  GraphLab  scikit-learn
Apache Spark  MLbase  KeystoneML  GraphX  Splash  CoCoA
Learning

Big Data

Training

Big Model

Conference Papers

Dashboards and Reports

Drive Actions
Big Data → Training → Big Model → Drive Actions

- Learning
- Drive Actions

- Big Data
- Training
- Big Model
- Drive Actions
Big Data → Training → Big Model → Inference
Often overlooked
Timescale: ~10 milliseconds
An area of focus in the RISELab
why is **Inference** challenging?

Need to render **low latency** (< 10ms) predictions for **complex** models under **heavy load** with system **failures**.

**Models**

**Queries**

SELECT * FROM users JOIN items, click_logs, pages WHERE ...

**Features**
Inference is moving beyond the cloud

Augmented Reality
Home Security
Home Automation
Mobile Assistants
Self Driving Cars
Personal Robotics
Inference is moving beyond the cloud

Opportunities
- Reduce latency and improve privacy
- Address network partitions

Research Challenges
- Minimize power consumption
- Limited hardware & long life-cycles
- Develop new hybrid models to leverage cloud and devices
Robust Inference is critical

Self “Parking” Cars

Self “Driving” Cars

Chat AIs
Learning

Big Data

Training

Timescale: hours to weeks
Often re-run training
Another area of focus in RISE

Inference

Decision

Application

Feedback
Why is **Closing the Loop** challenging?

**Implicit and Delayed Feedback**

**Self Reinforcing Feedback Loops**

**World Changes at varying rates**
Big Data

Big Model Training

Application

Responsive (~10ms)

Adaptive (~1 seconds)

Learning

Inference

Feedback

Responsive

Query

Decision
Intelligence in **Sensitive Contexts**

AR/VR Systems  |  Home Monitoring  |  Voice Technologies  |  Medical Imaging

Protect the **data**, the **model**, and the **query**
Protect the **data**, the **model**, and the **query**

**High-Value Data is Sensitive**
- Medical Info.
- Home video
- Finance

**Models capture value in data**
- Core Asset
- Sensitive

**Queries** can be as sensitive as the data
Adaptive

Responsive

Secure
Goal

**Real-time decisions**

decide in ms

on live data

with strong **security**
Goal

Real-time decisions

decide in ms

on live data

the current state of the environment

with strong security
Goal

Real-time decisions

decide in ms

on live data

the current state of the environment

with strong security

privacy, confidentiality, integrity
<table>
<thead>
<tr>
<th>Challenges</th>
<th>RISE Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automated decisions on live data are hard</td>
<td>Real-time, sophisticated decisions that guarantee worst-case behavior on noisy and unforseen live data</td>
</tr>
<tr>
<td>Poor security: exploits are daily occurrences</td>
<td>Ensure privacy and integrity without impacting functionality</td>
</tr>
<tr>
<td>One-off solutions, expensive and slow to build</td>
<td>General platform: Secure Real-time Decision Stack</td>
</tr>
</tbody>
</table>
Example: Zero-time defense

Problem: zero-day attacks can compromise millions of sites in seconds

Solution: analyze network flows to detect attacks and patch sites/software in real-time

- Intermediate data: create attack model
- Decision: detect attack, patch

<table>
<thead>
<tr>
<th>Quality</th>
<th>sophisticated, accurate, robust</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>sec (decision) / sec (update)</td>
</tr>
<tr>
<td>Security</td>
<td>privacy (encourage users to share logs), integrity</td>
</tr>
</tbody>
</table>
Example: “Fleet” driving

**Problem**: suboptimal driving decisions

**Solution**: collect & leverage info from other cars and drivers in **real-time**

- **Intermediate data**: automatically annotate maps, actions of other drivers
- **Decision**: avoid obstacles, congestions

<table>
<thead>
<tr>
<th>Quality</th>
<th>sophisticated, accurate, noise tolerant</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td>sec (decision) / sec (update)</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td>privacy, data integrity</td>
</tr>
</tbody>
</table>
Example: Infectious disease discovery

Problem: infectious diseases spread quickly (Zika), may need quarantine (Ebola)

Solutions: real-time DNA seq. & analysis to identify pathogens
- Rapid analysis to trace evolution, source
- 100x faster → 100x people tested
  - Intermediate data: evolution, spread, symptoms
  - Decision: quarantine or not, diagnosis

<table>
<thead>
<tr>
<th>Quality</th>
<th>sophisticated, accurate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency</td>
<td>min (decision) / hour (update)</td>
</tr>
<tr>
<td>Security</td>
<td>privacy, integrity</td>
</tr>
</tbody>
</table>

MinION Nanopore (Dr. Charles Chiu UCSF using it to identify Zika virus)
<table>
<thead>
<tr>
<th>Applications</th>
<th>Quality</th>
<th>Latency</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Decision</td>
<td>Update</td>
</tr>
<tr>
<td>Zero-time defense</td>
<td>sophisticated, accurate, robust</td>
<td>sec</td>
<td>sec</td>
</tr>
<tr>
<td>Parking assistant</td>
<td>sophisticated, robust</td>
<td>sec</td>
<td>sec</td>
</tr>
<tr>
<td>Disease discovery</td>
<td>sophisticated, accurate</td>
<td>sec/min</td>
<td>hours</td>
</tr>
<tr>
<td>IoT (smart buildings)</td>
<td>sophisticated, robust</td>
<td>sec</td>
<td>min/hour</td>
</tr>
<tr>
<td>Earthquake warning</td>
<td>sophisticated, accurate, robust</td>
<td>ms</td>
<td>min</td>
</tr>
<tr>
<td>Chip manufacturing</td>
<td>sophisticated, accurate, robust</td>
<td>sec/min</td>
<td>min</td>
</tr>
<tr>
<td>Fraud detection</td>
<td>sophisticated, accurate</td>
<td>ms</td>
<td>min</td>
</tr>
<tr>
<td>“Fleet” driving</td>
<td>sophisticated, accurate, robust</td>
<td>sec</td>
<td>sec</td>
</tr>
<tr>
<td>Virtual companion</td>
<td>sophisticated, robust</td>
<td>sec</td>
<td>min/hour</td>
</tr>
<tr>
<td>Video QoS at scale</td>
<td>sophisticated</td>
<td>ms/sec</td>
<td>min</td>
</tr>
<tr>
<td>Challenges</td>
<td>RISE Lab</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-----------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automated decisions on live data are hard</td>
<td><strong>Real-time, sophisticated decisions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>that guarantee worst-case behavior</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>on noisy and unforseen live data</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Poor security: exploits are daily occurrences</td>
<td><strong>Ensure privacy and integrity</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>without impacting functionality</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>One-off solutions, expensive and slow to build</td>
<td><strong>General platform:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Secure Real-time Decision Stack</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A bird’s eye view

Secure Real-time Decision Stack

decision
query
push decision
end-point info
end-points (e.g., users, devices)
What exists today?
Pull decisions

Example: recommendation system

- Ingestion (Kafka)
- ETL/ML (Hadoop, Spark, Cloud Dataflow)
- Storage (HDFS)

Service Layer (WebApp, Cassandra, MySQL)

recommendation

request

per-user recommendations

collaborative filtering

user purchase and rating history

Example: recommendation system
Pull decisions

Problem: cannot have sophisticated decisions

Cannot pre-compute all decisions
Cannot touch much data
Pull decisions

- **Ingestion (Kafka)**
- **ETL/ML (Hadoop, Spark, Cloud Dataflow)**
- **Service Layer (WebApp, Cassandra, MySQL)**

Cannot pre-compute all decisions
Cannot touch much data

Secure (encryption at rest)
## Solution scorecard

<table>
<thead>
<tr>
<th>Solution</th>
<th>Decision Quality</th>
<th>Latency</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pull decisions</td>
<td>simple</td>
<td>ms</td>
<td>min/day</td>
</tr>
</tbody>
</table>
Pull decisions: contextual decisions

Ingestion (Kafka) -> Prediction Serving Layer (WebApp, Velox, on-line MWT) -> ML (Hadoop, Spark, Cloud Dataflow) -> Storage (HDFS)

- Decision
- Request

ML:
- off-line, large models
- light-weight, ensemble and correction models, policies

Ingestion (Kafka)
## Solution scorecard

<table>
<thead>
<tr>
<th>Solution</th>
<th>Decision Quality</th>
<th>Latency Decision</th>
<th>Latency Update</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pull decisions</td>
<td>simple</td>
<td>ms</td>
<td>min/day</td>
<td>weak</td>
</tr>
<tr>
<td>Pull decisions: prediction service</td>
<td>sophisticated, specialized</td>
<td>ms</td>
<td>min*</td>
<td>weak</td>
</tr>
</tbody>
</table>

*light-weight, ensemble + correction models, policies
Push Decisions

Example: anomaly detection

- Ingestion (Kafka)
- ETL/ML (Hadoop, Spark)
- Storage (HDFS)
- Streaming (Storm, Spark, IBM Streams)

Outlier detection
Support vector machine (SVM)
Machine logs
Alert
Push Decisions

Example: anomaly detection

- Ingestion (Kafka)
- ETL/ML (Hadoop, Spark)
- Streaming (Storm, Spark, IBM Streams)
- Storage (HDFS)

Alerts: ms/sec

Outlier detection
Support vector machine (SVM)
Machine logs
## Solution scorecard

<table>
<thead>
<tr>
<th>Solution</th>
<th>Decision Quality</th>
<th>Latency Decision</th>
<th>Latency Update</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pull decisions</td>
<td>simple</td>
<td>ms</td>
<td>min/day</td>
<td>weak</td>
</tr>
<tr>
<td>Push decisions</td>
<td>simple</td>
<td>ms/sec</td>
<td>hour</td>
<td>weak</td>
</tr>
</tbody>
</table>

*light-weight, ensemble + correction models, policies*
Security tools

- Computation on encrypted data
- Hardware enclaves
State-of-the-art security solutions: Computation on encrypted data
State-of-the-art security solutions: Computation on encrypted data

Ensure security, but relatively simple algorithms

SQL (CryptDB), ML classification, web serving (Mylar), etc.
State-of-the-art security solutions: Hardware enclaves

TCB, side-channel leakage

- Ingestion (Kafka)
- Key-value store (Cassandra, MySQL)
- ETL/ML (Hadoop, Spark)
- Web App

Intel SGX, ARM Trustzone, RISC V (e.g., Haven, VC3)

TCB, side-channel leakage
## Solution scorecard

<table>
<thead>
<tr>
<th>Solution</th>
<th>Decision Quality</th>
<th>Latency</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Decision</td>
<td>Update</td>
</tr>
<tr>
<td>Pull decisions</td>
<td>simple</td>
<td>ms</td>
<td>min/day</td>
</tr>
<tr>
<td>Push decisions</td>
<td>simple</td>
<td>ms/sec</td>
<td>hour</td>
</tr>
<tr>
<td>State-of-the-art security</td>
<td>simple</td>
<td>ms</td>
<td>min/hour</td>
</tr>
</tbody>
</table>

*light-weight, ensemble + correction models, policies*
## Solution scorecard: RISE

<table>
<thead>
<tr>
<th>Solution</th>
<th>Decision Quality</th>
<th>Latency</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Decision</td>
<td>Update</td>
</tr>
<tr>
<td>Pull decisions</td>
<td>simple</td>
<td>ms</td>
<td>min/day</td>
</tr>
<tr>
<td>Push decisions</td>
<td>simple</td>
<td>ms/sec</td>
<td>hour</td>
</tr>
<tr>
<td>Security</td>
<td>simple</td>
<td>ms</td>
<td>min/hour</td>
</tr>
<tr>
<td>RISE</td>
<td>sophisticated, accurate, robust</td>
<td>ms</td>
<td>sec</td>
</tr>
</tbody>
</table>

*light-weight, ensemble + correction models, policies
Research areas

Systems: Spark-like functionality with 100x lower response time, and 1000x higher job throughput

Machine Learning:
• On-line ML algorithms
• Robust algorithms: handle noisy data, guarantee worst-case behavior

Security: achieve privacy, confidentiality, and integrity without impacting performance
Early Projects

Opaque: A Data Analytics Platform with Strong Security

Abstract

One key question when implementing the oblivious computation should not leak any access pattern. This is especially important when dealing with sensitive data, such as financial data or medical records. Opaque, as a distributed and novel query planning technique, aims to provide strong security guarantees without significantly reducing performance.

Introduction

The name "Opaque" stands for Oblivious Platform for Analytic Queries, as well as opacity, meaning no sensitive information is visible. Opaque is designed to work at the query optimization layer, allowing for secure execution of arbitrary computations without revealing access patterns.

Early Projects

Opaque promises a much-needed solution to data breaches, as seen in the use of hardware enclaves. During a single run of a spellcheck application, these enclaves can extract hundreds of kilobytes of data from confidential access patterns. Previous work such as Haven has shown that it is possible to run unmodified binaries and computation at processor speeds while protecting the data.

Memory Encryption

Hardware enclaves (such as Intel SGX) promise support for arbitrary computation at the memory level and at the attack vector: MapReduce jobs using enclaves.

Recent developments and trends in big data processing frameworks provide us with a compelling opportunity: the elasticity and cost savings. However, the risk of data breaches is hampering this trend. While hardening approaches are being explored, they still suffer from an important leakage vector: the access pattern leakage.

Therefore, to truly secure sensitive data like user information (emails, social interactions, shopping history), medical data, and financial data, these systems extract value out of this data, which contains patterns that reveal information.

Regarding the attacks are of two kinds: at the memory level and at the attack vector: MapReduce jobs using enclaves. Recent developments and trends in big data processing frameworks provide us with a compelling opportunity: the elasticity and cost savings. However, the risk of data breaches is hampering this trend. While hardening approaches are being explored, they still suffer from an important leakage vector: the access pattern leakage.

Fortunately, recent developments and trends in big data processing frameworks provide us with a compelling opportunity: the elasticity and cost savings. However, the risk of data breaches is hampering this trend. While hardening approaches are being explored, they still suffer from an important leakage vector: the access pattern leakage.

Early Projects

Opaque promises a much-needed solution to data breaches, as seen in the use of hardware enclaves. During a single run of a spellcheck application, these enclaves can extract hundreds of kilobytes of data from confidential access patterns. Previous work such as Haven has shown that it is possible to run unmodified binaries and computation at processor speeds while protecting the data.

Memory Encryption

Hardware enclaves (such as Intel SGX) promise support for arbitrary computation at the memory level and at the attack vector: MapReduce jobs using enclaves.
Research area: Systems

- on-going work
- future work

- unified model (rich experience w/ both models)
- IndexedRDDs
- GPU/ASICs algos
- Drizzle
  - sharded driver,
  - in-memory processing,
  - per-core NIC, HBM
  - system-state store

- Support task-graph & BSP execution models
- Support fine grain updates
- Support heterogeneous hardware
- **Millisecond** level parallel jobs
- Handle **10K-100K** jobs/sec
- Ability to faithfully **replay** jobs
THE MEANING AND VALUE OF DATA DEPENDS ON CONTEXT

**Application context**
Multiple data models

because truth is subjective

**Behavioral context**
Logs and Lineage

because behavior determines meaning

**Historical context**
Immutable versions for code and data

because things change

A broader context for big data
LatticeFlow and Bedrock (working names)

Driving Hypotheses for LatticeFlow
• A core programming API for both real-time and scale
• Everything is (async) data: event dispatch, real-time data streams
• Coordination Avoidance: lattices + async dataflow = no locks/barriers

Initial proof point this fall: LatticeKVS
MultiScale, MultiConsistency key-value store
• Same system beats Redis on one node, Cassandra on scale-out…
• …while providing family of rich consistency and transactional isolation options
• …with lean codebase, derived from a core LatticeFlow library in C++

Prototype toward “Project Bedrock”
• Immutable, never-forget versioned storage under ground
Research area: Systems & ML

Focus on robust, on-line algorithms

Focus on ML/DL support (e.g., Clipper, Ray)

already gathering industry support

ML algorithms, Tools, Apps

Apache Spark

Ray

Clipper

Succinct

shim layer

shim layer

shim layer

shim layer

 scheduler

 optimizer

 ...  RISE μkernel

in-memory object store

Ground (metadata manager)

App Layer

Comp. Layer

Storage Layer

60

RISE

3rd party

RISE + 3rd party

cassandra

HDFS

S3

mongoDB
Research area: ML

Robust optimization methods:
  • noise tolerant and parallelizable

Handle uncertainty:
  • robust control techniques to handle unforseen real world situations

Quantify decision accuracy:
  • confidence and credible intervals on outputs of ML systems

On-line ML algorithms
  • handle time-varying models with high performance and reliability
End-to-end system

ML algorithms, Tools, Apps

Secure Real-time Decision Stack

Apache Spark
Ray
Clipper
Succinct

shim layer
shim layer
shim layer
shim layer

scheduler
optimizer

in-memory object store

Ground (metadata manager)

cassandra
HDFS
S3
mongoDB

RISE υkernel
Research area: Security

Split comp. between enclave and on encrypted data
Research area: Security

- ML (decisions) on encrypted data
- **Opaque**: oblivious & encrypted analytics
- New hardware support (RISC V, HBM)
- Fine grained access control
- **Arx**: queries on encrypted data

### ML algorithms, Tools, Apps
- Apache Spark
- Ray
- Clipper
- Succinct

- scheduler
- optimizer
- **RISE μkernel**
- in-memory object store
- Ground (metadata manager)

- cassandra
- S3
- mongoDB
- HDFS
Why now, why us?

Latency

Quality

Security

Hardware Trends

Processing
• RISC V*, GPUs, ASICs
• Built in security support (enclaves)

Storage & networking
• 3D Xpoint

Next gen rack designs
• FireBox*

Systems
• Berkeley’s BDAS*

Machine Learning
• Robust algorithms*
• On-line ML* and DL

Security
• Code verification tools*
• Comp. on encrypted data*
Summary

**Goal**: develop Secure Real-time Decision Stack, an open source platform, tools and algorithms for real-time decisions on live data with strong security.

Five year project, similar to AMPLab
We are uniquely positioned to tackle this challenge

Looking to partner with companies