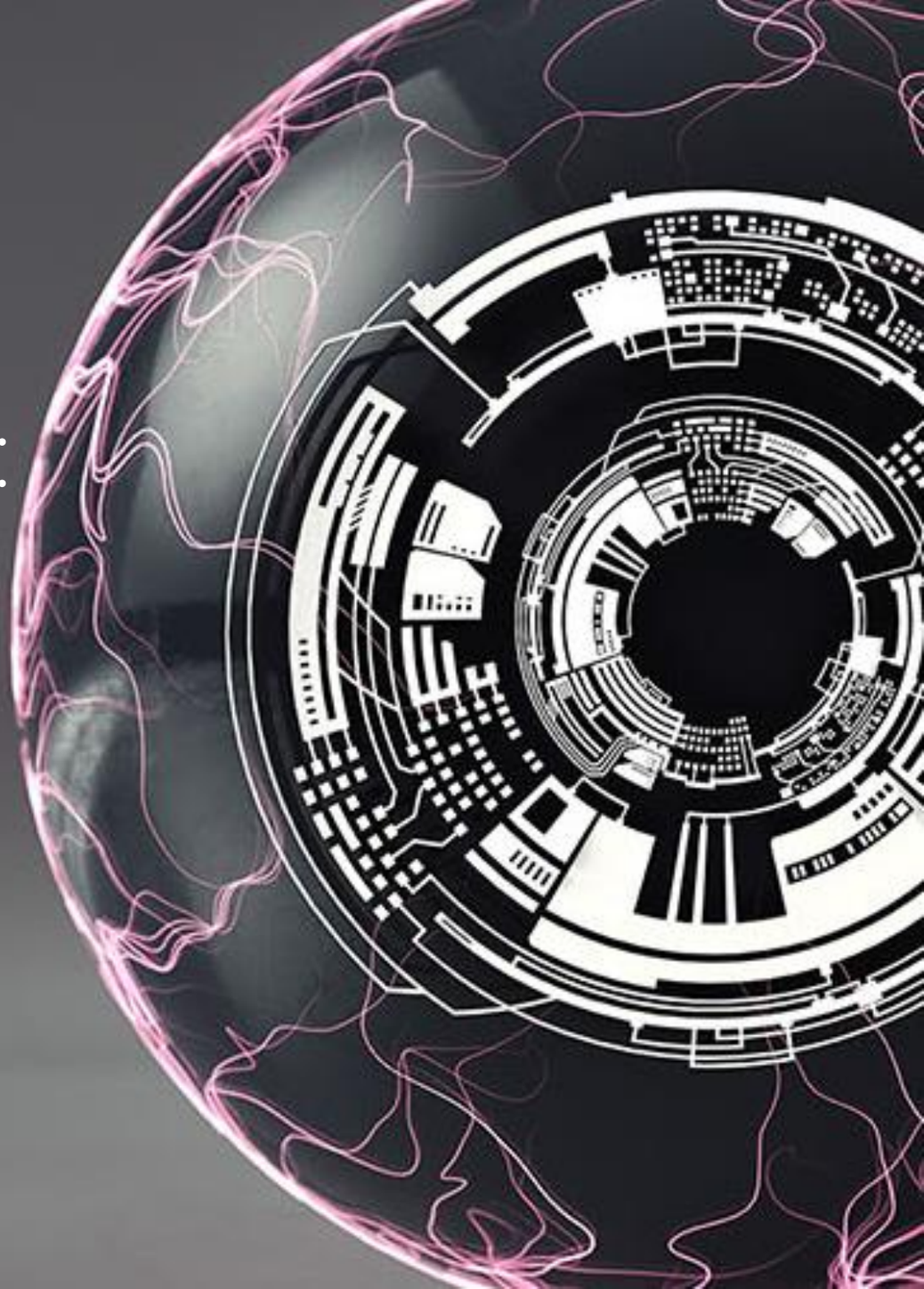
Microsoft

Big Data Infrastructure at Microsoft: From Research to Production

Lidong Zhou
Microsoft Research



Big Data Infrastructure: The Evolution

Foundation:

- Large-Scale Distributed Storage
- Data Flow Machinery
- Declarative Data Parallel Language

…… 2011 2012 2013 2014 2015 2016

SCOPE/Cosmos in Production: 2010 - 2013

Scale

Maximum Utilization and Throughput with High Reliability At Low Cost

110K Commodity machines
Across 4 clusters

60%-80% CPU Utilization

56s Terasort
1000 nodes

Storage growth
1.2 Exabytes
64 Petabytes
2010
2013

Ecosystem

Bing, Ad Center, MSN, Maps, Windows Phone, Xbox Live, Windows Live, Office365, STB, …

4K Developers

170 Feature Teams

100's Applications

Jobs per day
95K/day
6K/day
2010
2013

Simplicity

Developers, Researchers, Data Scientists, PM, Product Management, Marketing, and Sales

Customers — Big Data as a Service

Users — Familiar Tools
Excel, Web Apps, Reporting

Developers

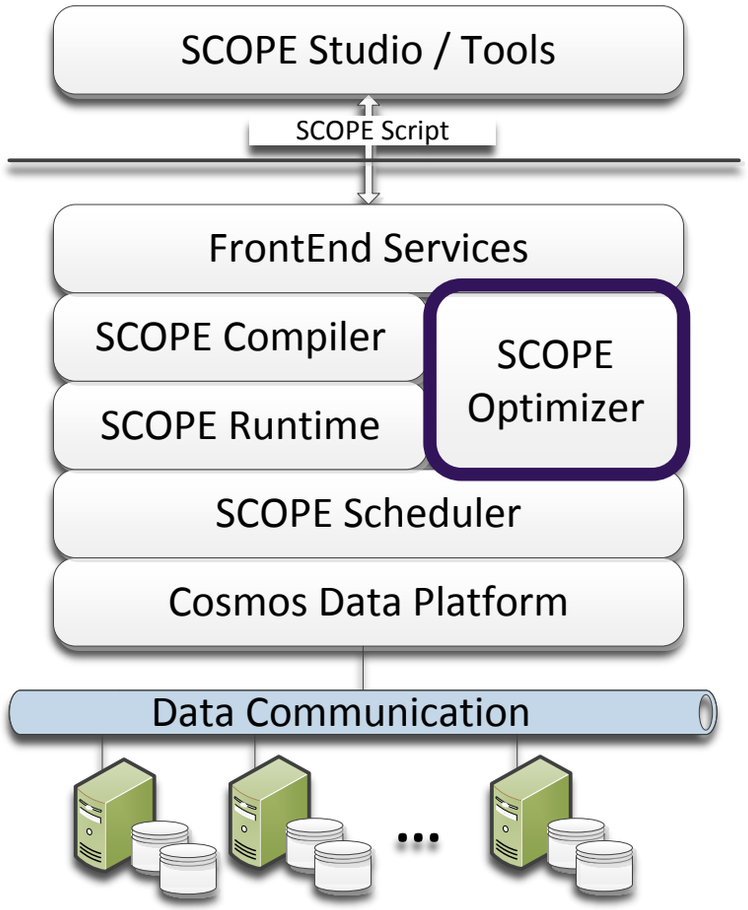
Scope (20 Lines)

```
//Define the View for Cluster Runs
SlapiPageView =
        VIEW "/shares/searchDM/distrib/released/SLAPI/SearchLogPageView.view"
        PARAMS (Start = @"2012-01-01", End = @"2012-01-01", UseSample = false);

//Select statement from data from the view
SELECT
        COUNT() AS Count
    ,Vertical
        ,Market
    ,Query_RawQuery
FROM
    SlapiPageView
WHERE
    Helper.FirstWordIs(Query_RawQuery,"windows")
ORDER BY
        Vertical, Count DESC;

//Output files to the cluster
OUTPUT
        TO @"/my/results.log";
```

MapReduce (200 Lines)

Courtesy of Big Data Team

SCOPE: Database Meets Map/Reduce




```
REFERENCE @"/shares/searchDM/SearchLogApi.dll";
USING MS.Internal.Bing.DataMining.SearchLogApi;

//Search Merge Log Impressions
SML =
    VIEW "/shares/searchDM/SearchLogPageView.view“
        PARAMS (Start = @"2013-07-10", End = @"2013-07-11“)
     ;

//Windows Blue distinct users
WindowsBlueClicks =
    SELECT
        Request_ClientId AS Client,
        QueryParser.GetFcsNormalizedQuery(Query_RawQuery) AS Query,
        SUM(PageClicks_Count > 0 ? 1 : 0) AS Clicks,
        MAX(Metrics_DwellTime) AS DwellTime
    FROM
        SMLPageView
    WHERE
        Market == "en-us“
        AND Request_OSInfo.ProductName == "Windows 8.1"
    ;

//Windows Blue user sessions
WindowsBlueSessions =
    REDUCE WindowsBlueClicks ON Client
        USING MySessionReducer()
    ;

//Cook for later use
OUTPUT WindowsBlueSessions
     TO SSTREAM @@WindowsBlueSessions@@
     CLUSTERED BY Vertical SORTED BY Client
    ;
```

SQL relational algebra

Predicates

Custom Reduce Function

Courtesy of Big Data Team

Big Data Infrastructure: The Evolution

Foundation:

- Large-Scale Distributed Storage
- Data Flow Machinery
- Declarative Data Parallel Language

Holistic Code Optimization

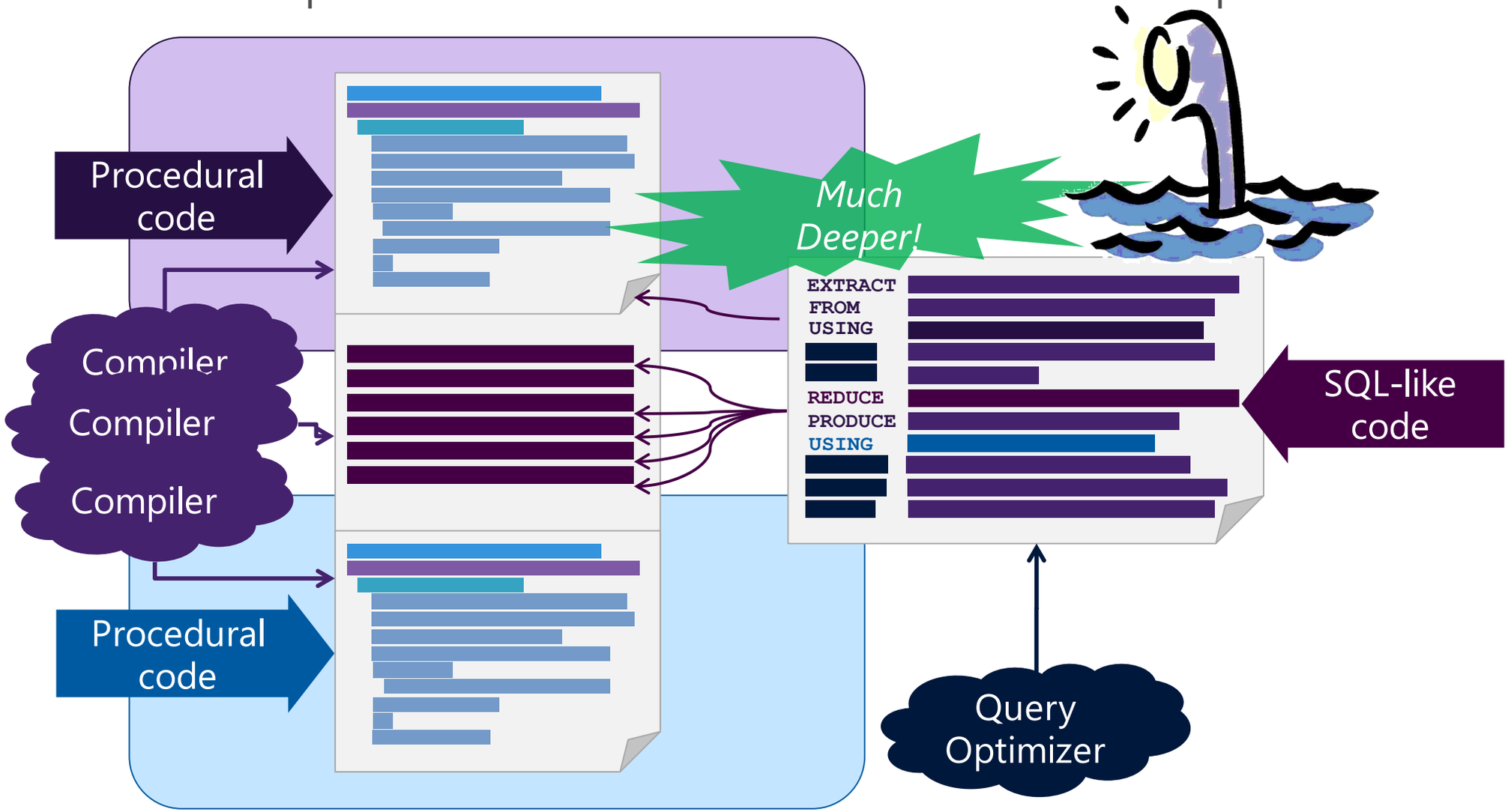
- Database Query Optimization
- Program Analysis and Compiler Optimization

…… 2011 2012 2013 2014 2015 2016

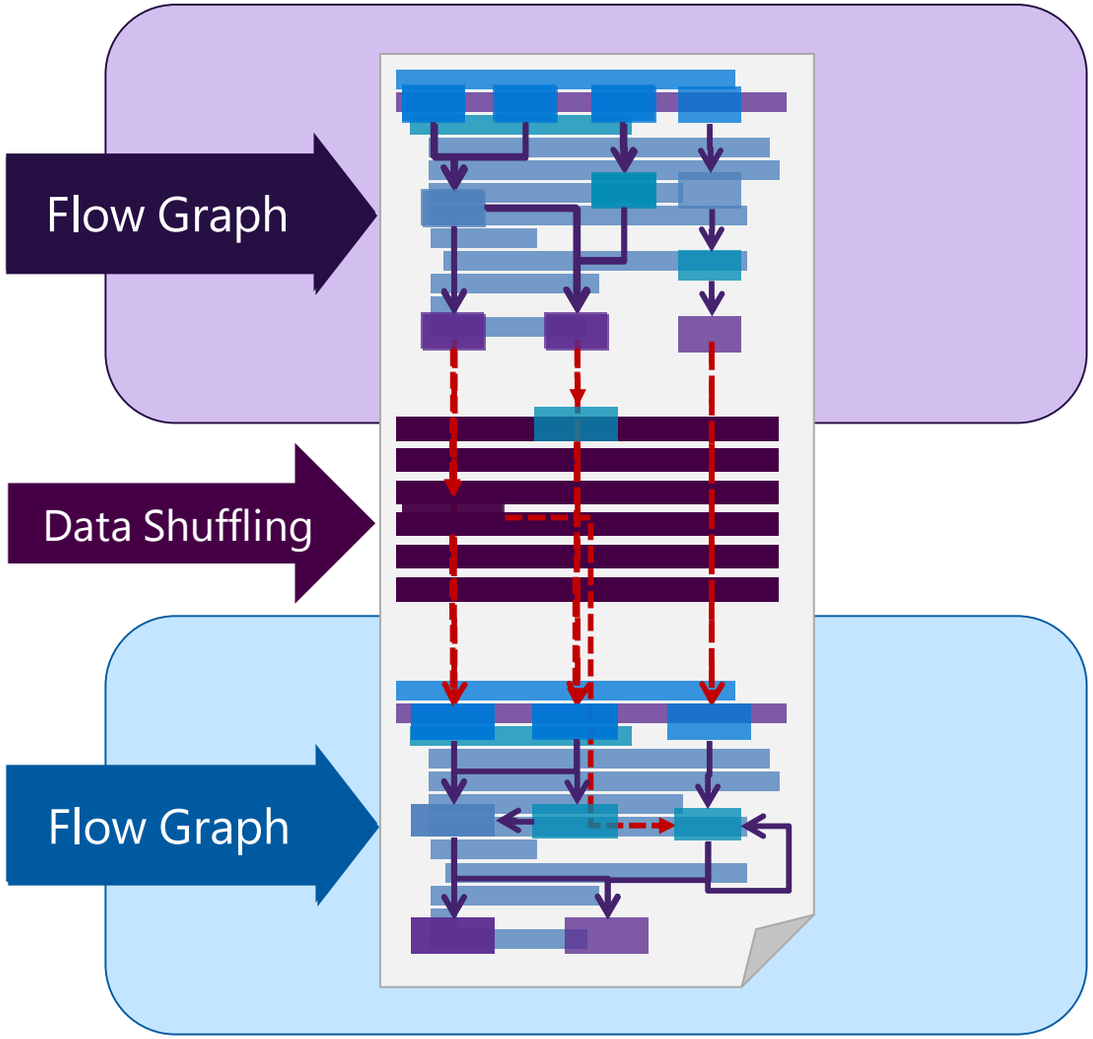
OSDI'12

NSDI'12

PeriSCOPE: Pipeline-aware Holistic Code Optimization



Optimization Steps

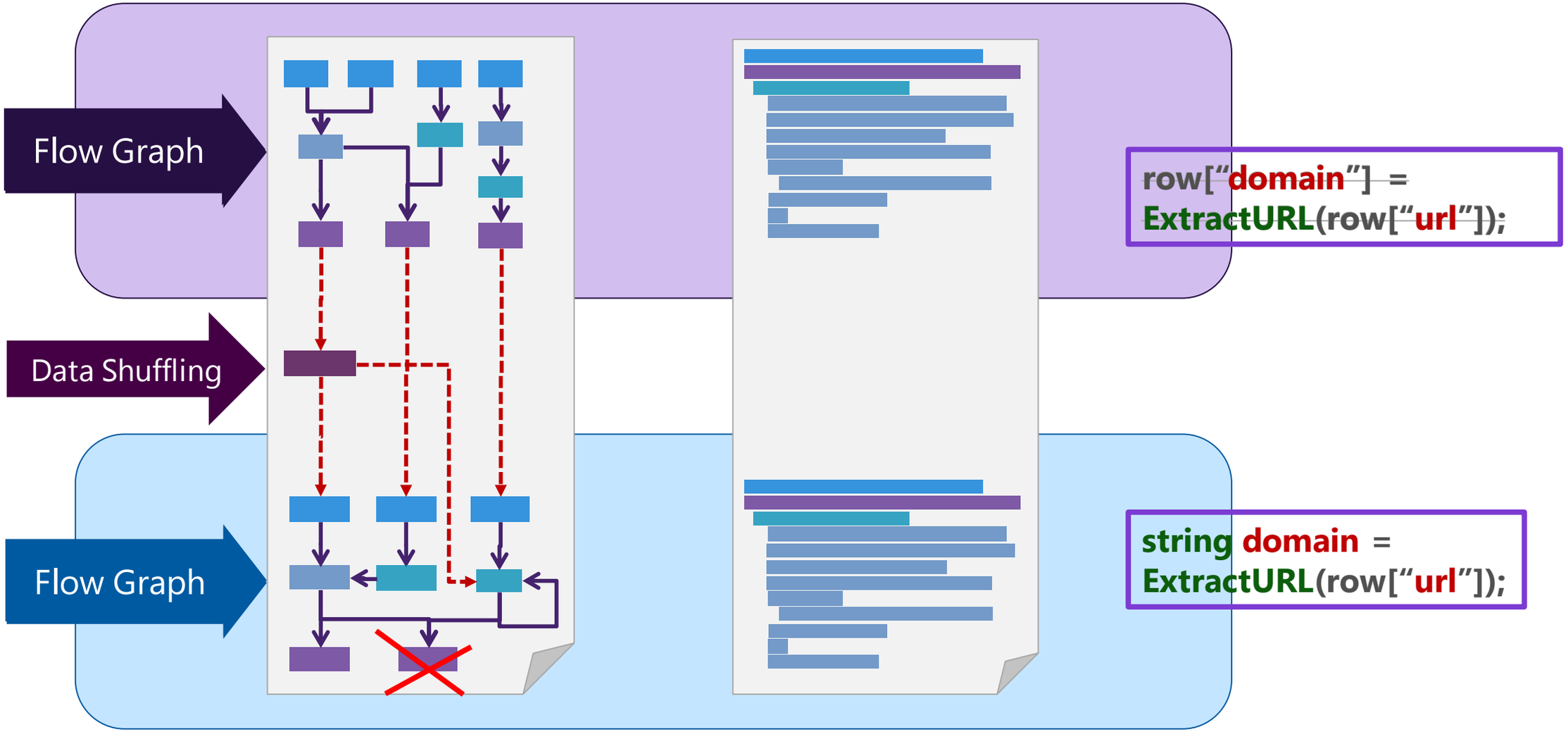


Step 1: Construct inter-procedural flow graph

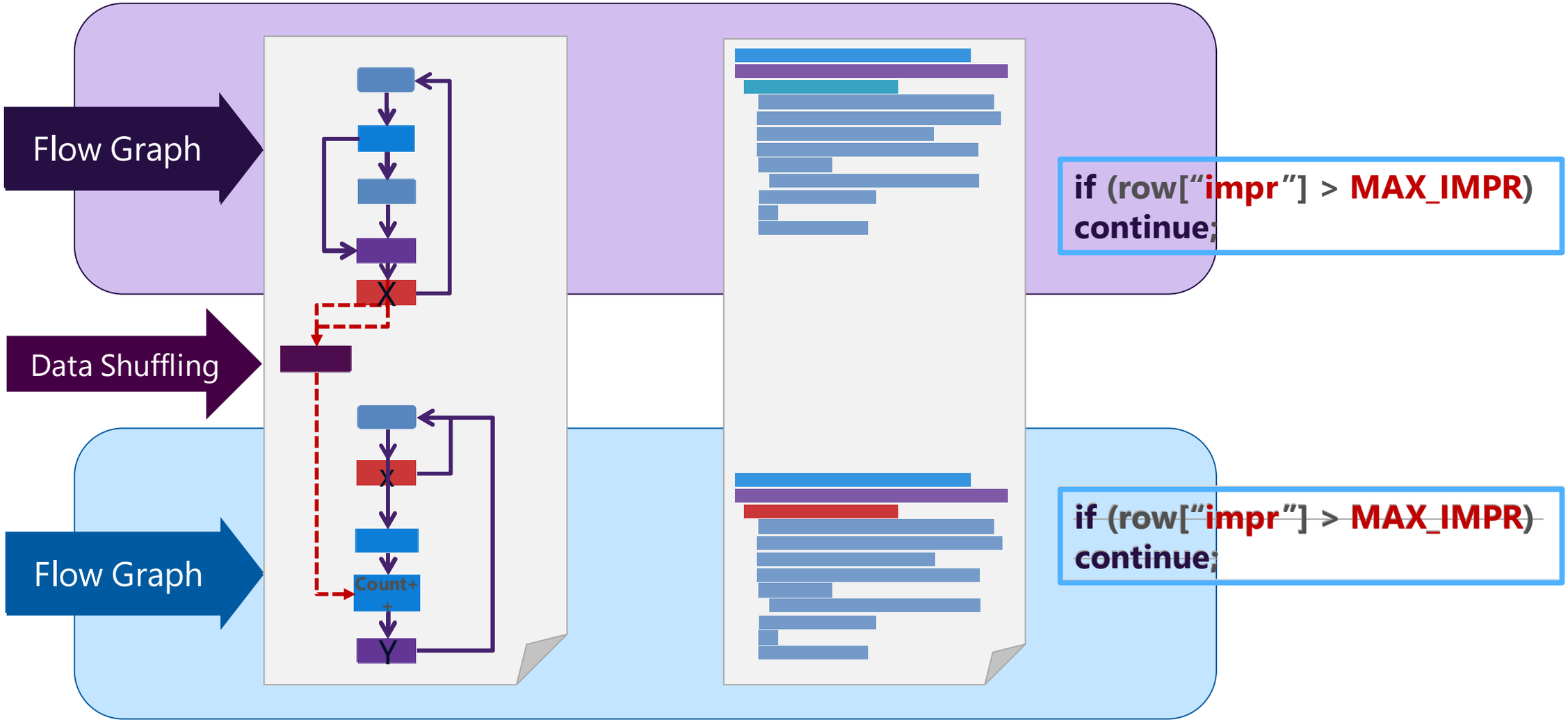
Step 2: Add safety constraints for skipping shuffling code

Step 3: Transform code for reducing shuffling I/O

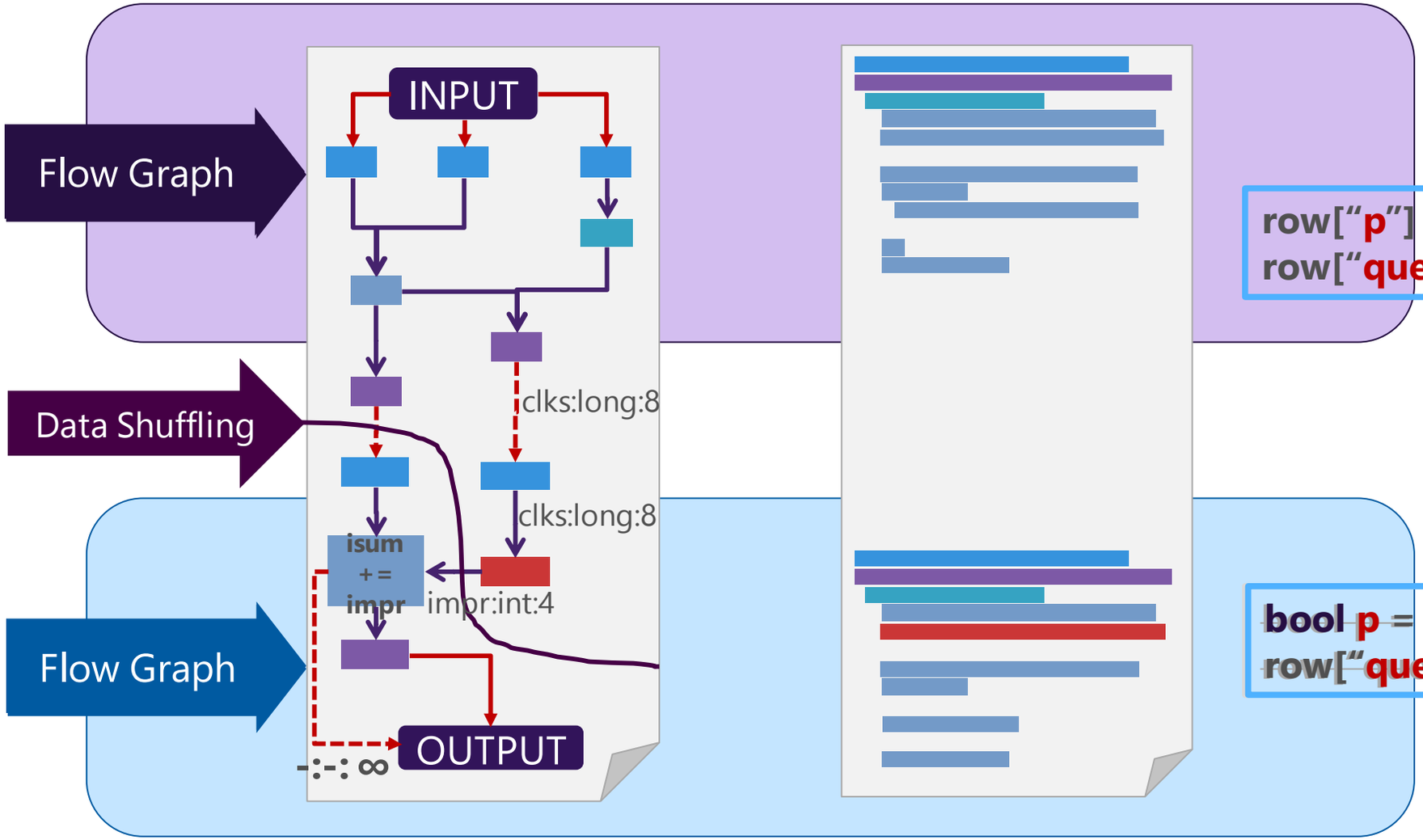
Column Reduction: Reduce Number of Columns



Early Filtering: Reduce Number of Rows



Smart Cut: Reduce Size of Each Row




```
row["p"] =
row["query"].Contains(KEYS);
```

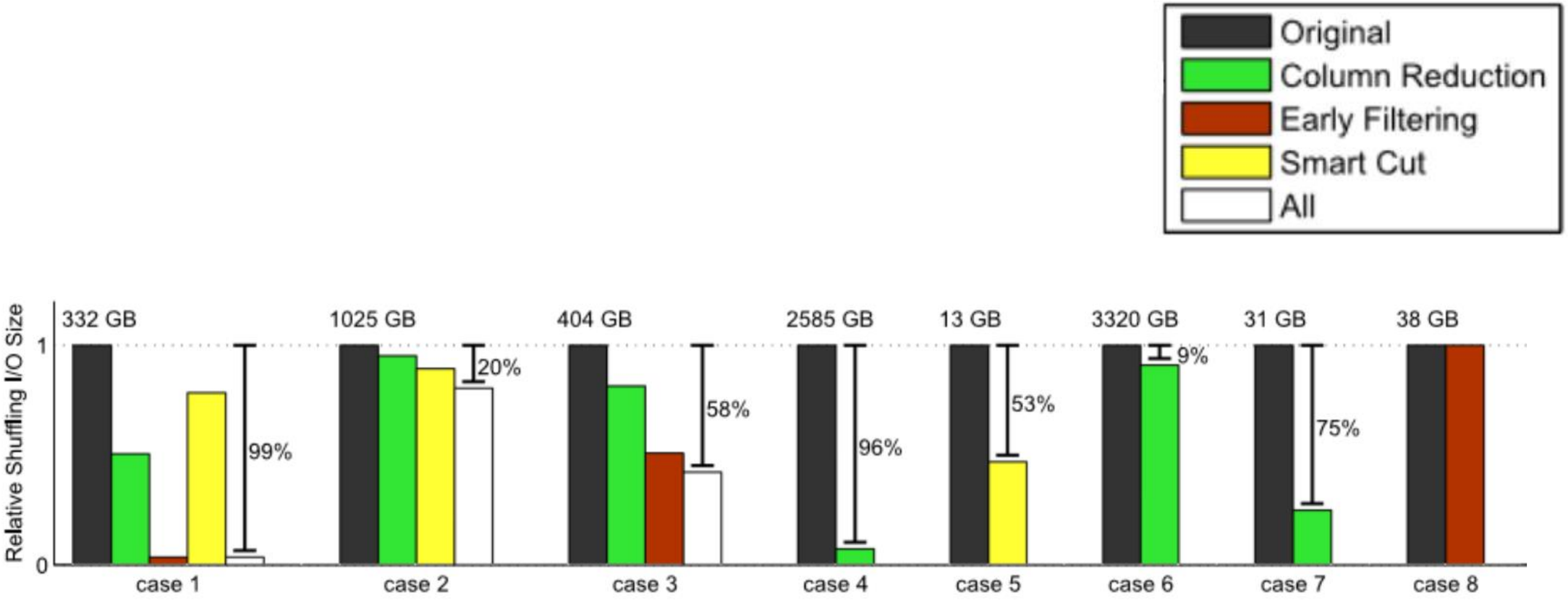
```
bool p =
row["query"].Contains(KEYS);
```

Coverage Study*

Optimization	Eligible jobs
Column Reduction	4,052 (14.05%)
Early Filtering	3,020 (10.47%)
Smart Cut	1,544 (5.35%)
Overlapped Total	6,397 (22.18%)

* Study on **28,838** jobs collected from SCOPE clusters in 2010/2011.

Significant I/O Reduction Observed



Research to Production

- State-of-art research in OSDI
- Validated with real jobs

Surprise: Not good enough!

- Absolutely do no harm: correctness and performance
- Coverage and overhead
- Complexity and tool maturity

Image credits:
http://m.rgbimg.com/cache1nvK96/users/o/oz/ozetsky/600/mfe0irG.jpg
http://cdn2.everyjoe.com/wp-content/uploads/2013/05/shocked-baby-146x104.jpg

Big Data Infrastructure: The Evolution

Foundation:

- Large-Scale Distributed Storage
- Data Flow Machinery
- Declarative Data Parallel Language

Holistic Code Optimization

- Database Query Optimization
- Program Analysis and Compiler Optimization

Scheduling and Resource Management

- Coordinated scheduling
- Opportunistic tasks
- Corrective actions

…… 2011 2012 2013 2014 2015 2016

OSDI'12

NSDI'12

OSDI'14

Scheduling at Scale

Jobs process **gigabytes to petabytes** of data

and issue peaks of **100,000 scheduling requests/ seconds**

Clusters run up to **170,000 tasks in parallel**

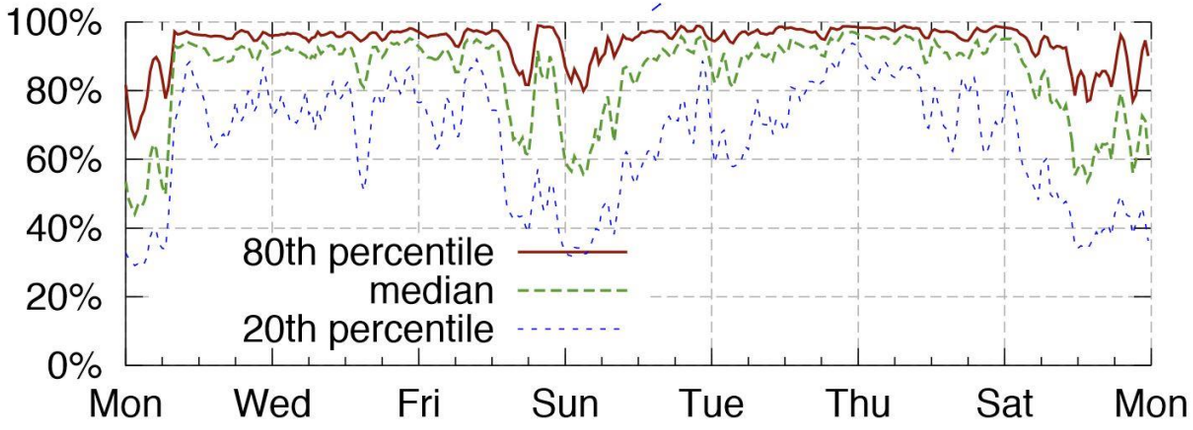
track **14,000,000 pending tasks**

and each contains **over 20,000 servers**

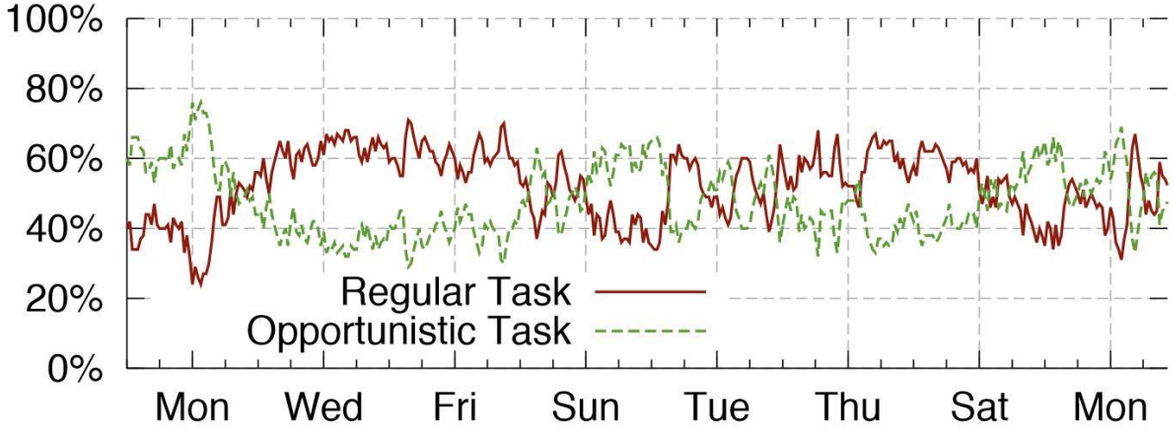
Incrementally rolled out from September to December 2013

Scheduling Quality

- 60-90% median CPU utilization
- Largely balanced load



- Opportunistic tasks fill the gaps (e.g., during weekends)
- Negligible queuing time for regular tasks



Big Data Infrastructure: The Evolution

Foundation:

- Large-Scale Distributed Storage
- Data Flow Machinery
- Declarative Data Parallel Language

Holistic Code Optimization

- Database Query Optimization
- Program Analysis and Compiler Optimization

Scheduling and Resource Management

- Coordinated scheduling
- Opportunistic scheduling
- Corrective actions

Beyond Batch Processing

- Graph Computation
- Machine Learning and Deep Learning
- **Streaming**

…… 2011 2012 2013 2014 2015 2016

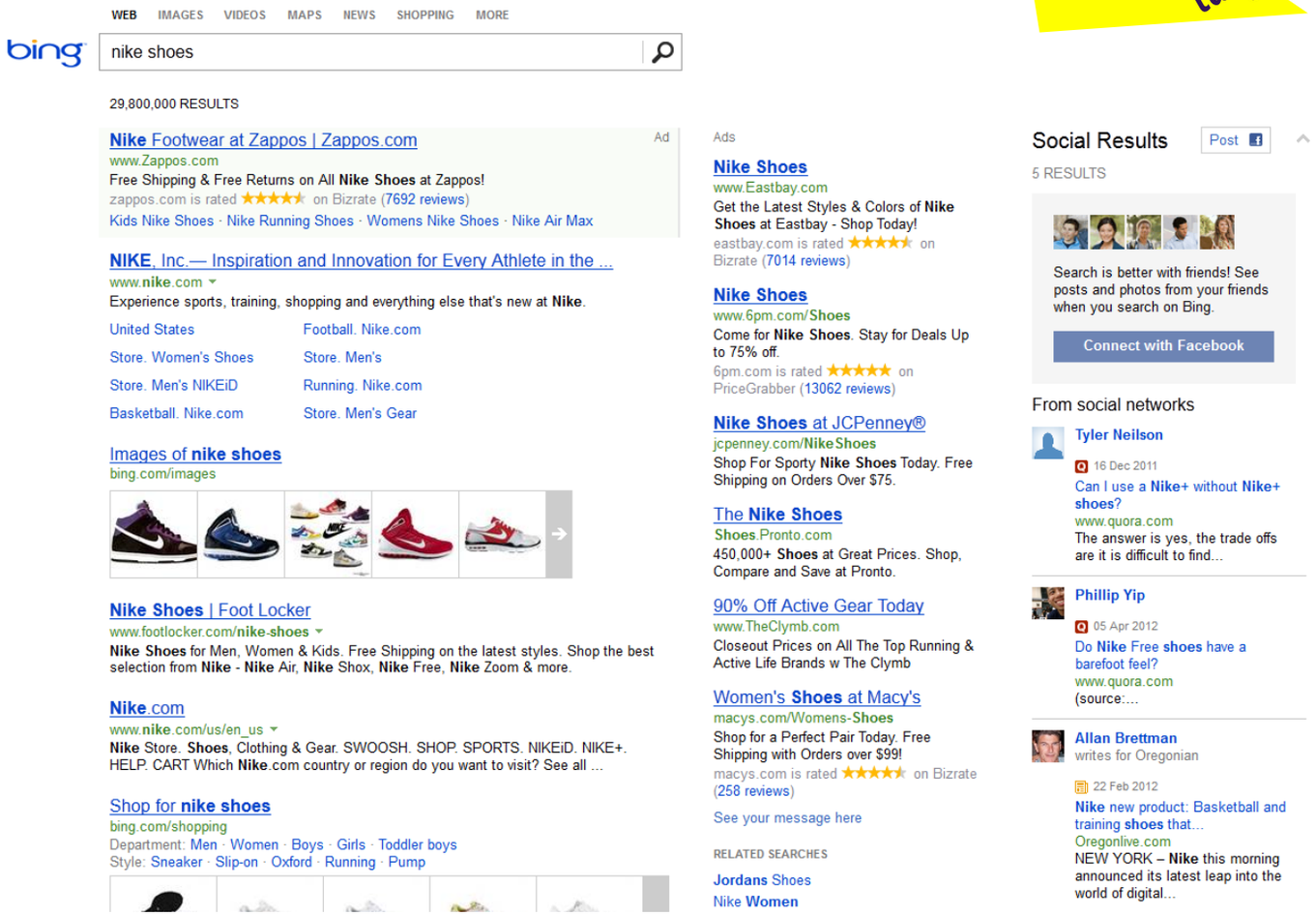
OSDI'12

NSDI'12 **Eurosys'13**

OSDI'14

SoCC'15 **NSDI'16**

Big Stream Computation



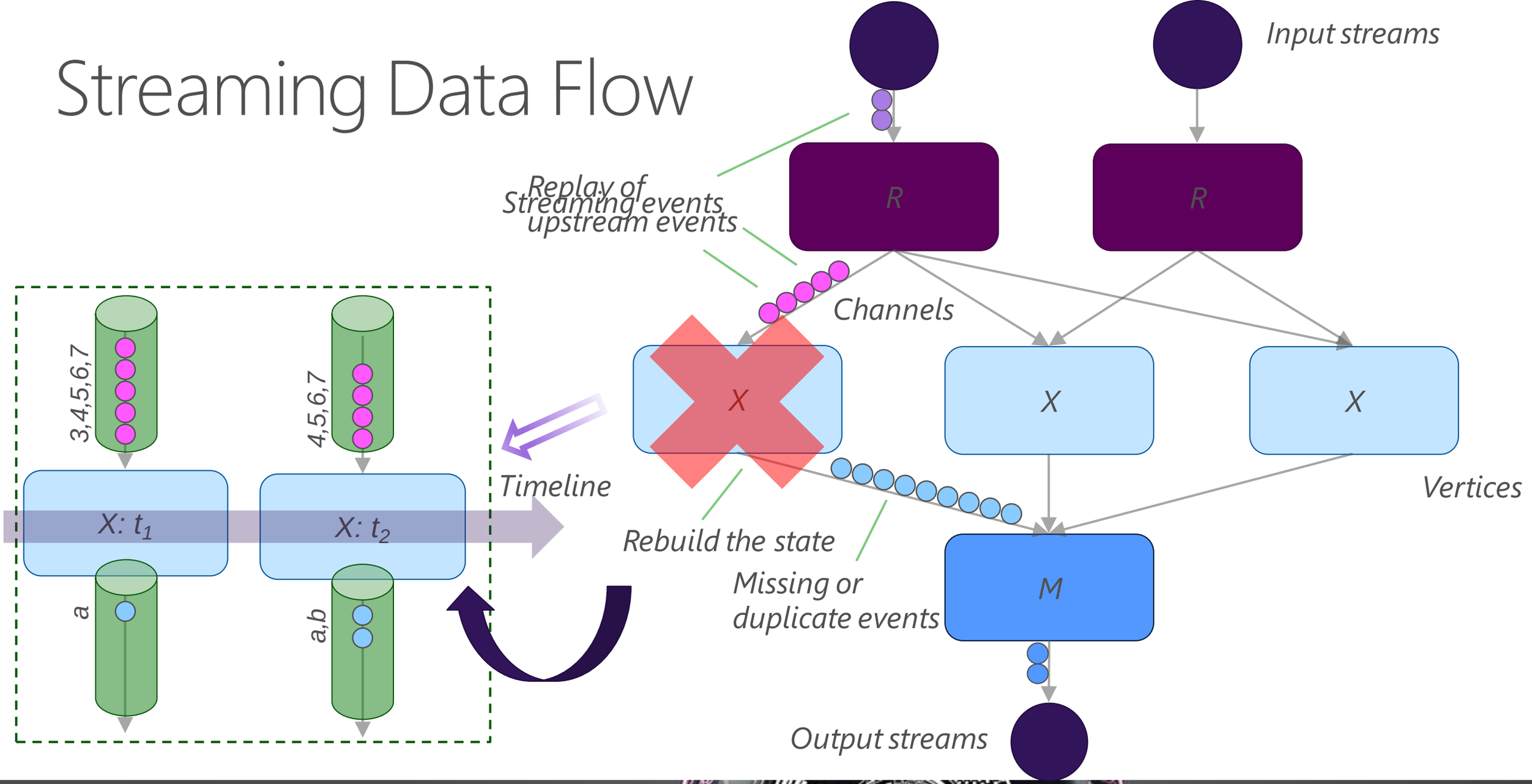
Click prediction, behavior targeting, etc.

$

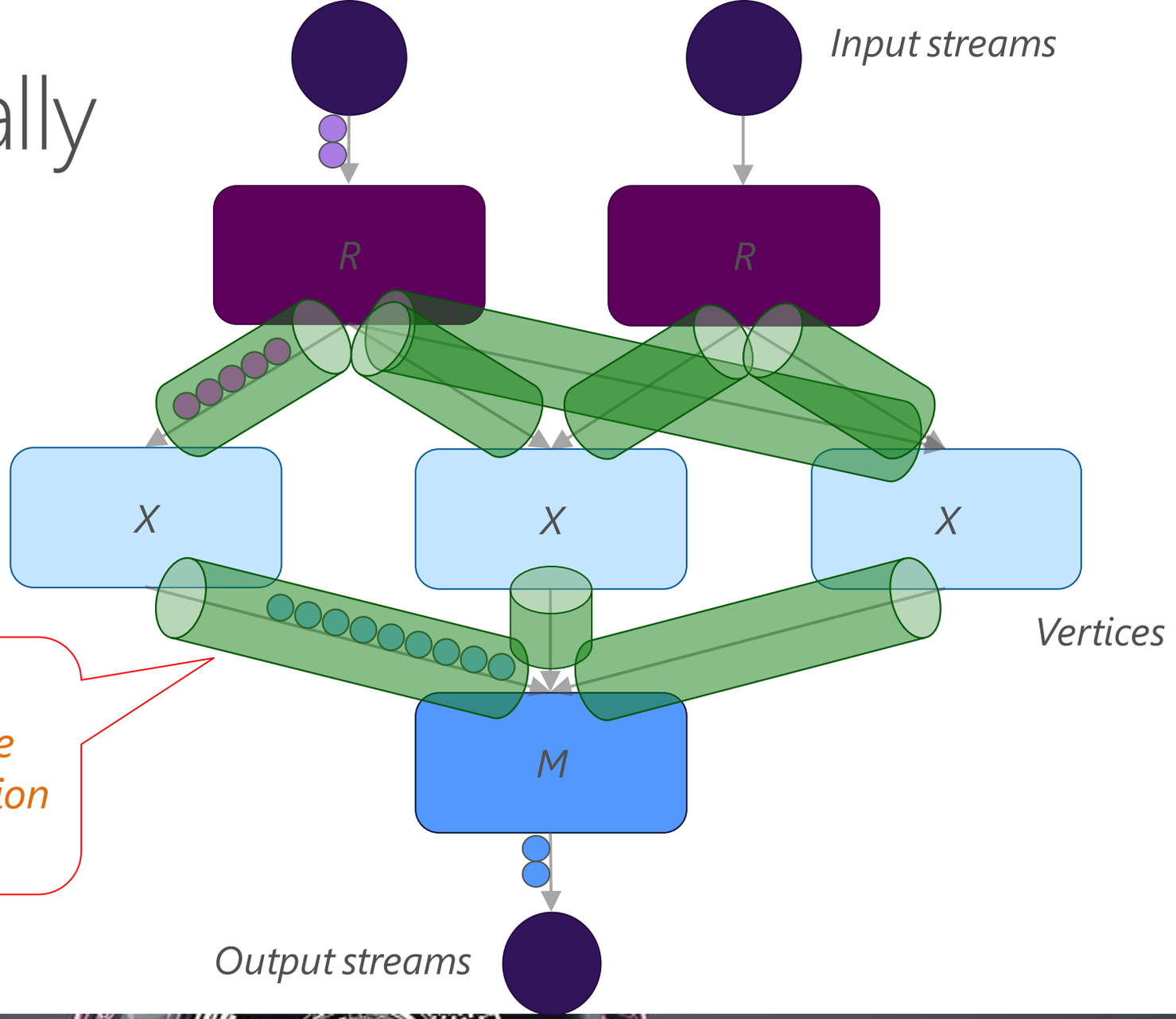
- Continuous input
- Near real-time computation
- Scaling to thousands of nodes
- Fault tolerant
- Strongly consistent

Search data streams

Streaming Data Flow

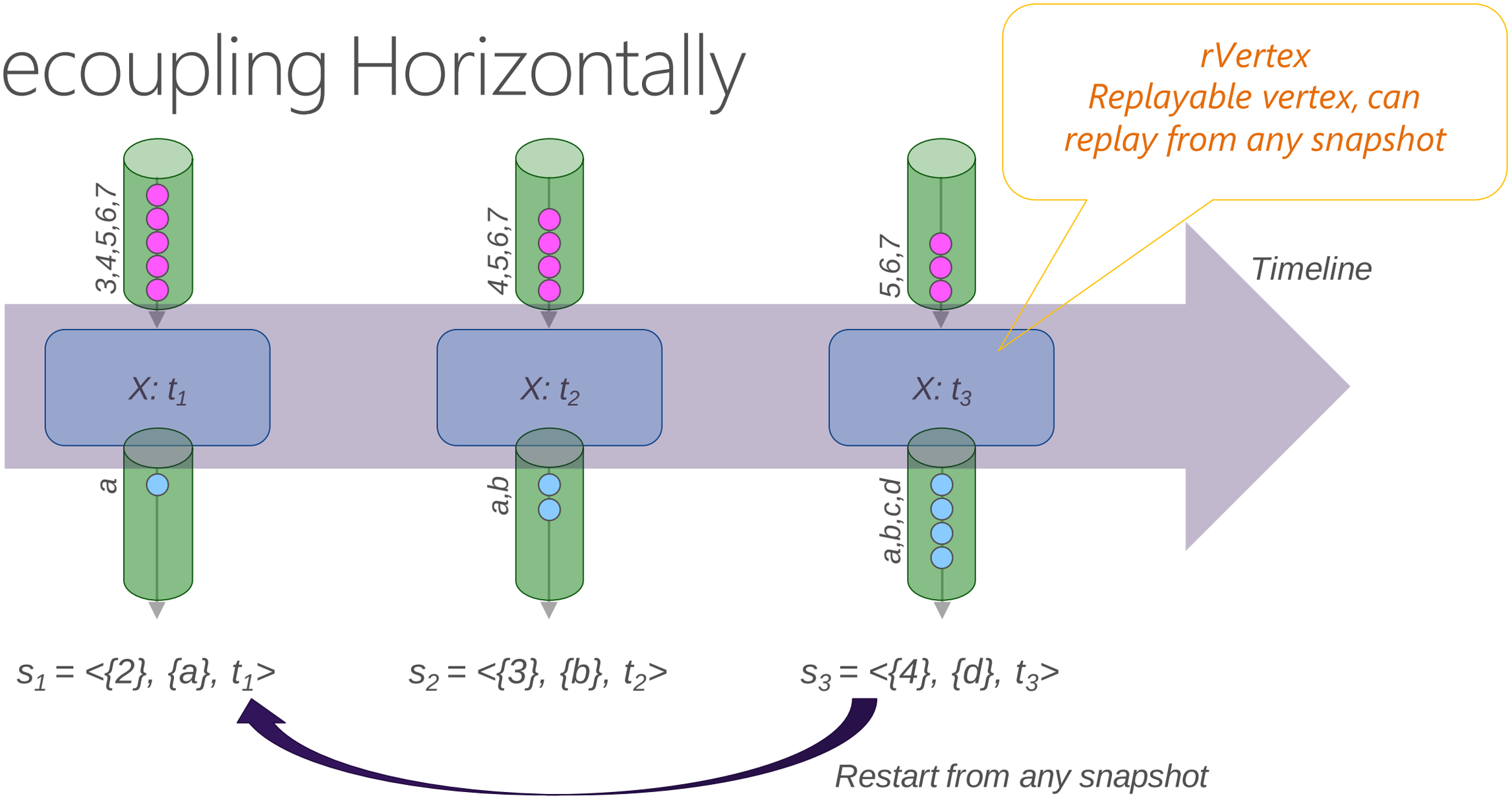


Decoupling Vertically



rStream
Provides the illusion of reliable and asynchronous communication channels

Decoupling Horizontally



Power of Abstraction

- Easy to reason about correctness
- Enabling powerful optimizations seamlessly

  Move reliable persistent writes off the critical path
- Allowing different instantiations throughout life cycle
  - Offline mode to test, profile, and debug individual vertices
  - Optimized implementation when deployed; simple ones for validation
  - Replication based failure recovery
  - Duplicate execution to handle stragglers and planned maintenance

State:
Completion:
Run Time:
Useful PN Hours: 14888:29:40.446
Bonus PN Hours: 43.45%

Runtime Name: scopecep_hcfan_201411
Submitted By: PHX\yuyao
Submit Time: 8/14/2014 8:18:45 PM
Compilation Time: 35 seconds
Queued Time: 1 seconds
Start Time: 8/14/2014 8:19:20 PM
End Time:
Yielded Time:

Cluster: cosmos11-prod-cy2
VC: adCenter.BICore
Priority: 800
Tokens: 482
Allocation(%): 11

Root Process Id: c3b436b7-292f-4a9f-8e5
Root Process Node: cy2sch030020747

Bytes Read: 61,336,252,895,109
Bytes Left: 249,538,700,649
Bytes Written: 60,124,233,285,838

Total Nodes: 2,876
-Completed:
-Running:
-Failed: 35

Job Diagnostics: Diagnose

Alert(s):

Data Skew: 0 shallow issue(s) detected.

Investigate

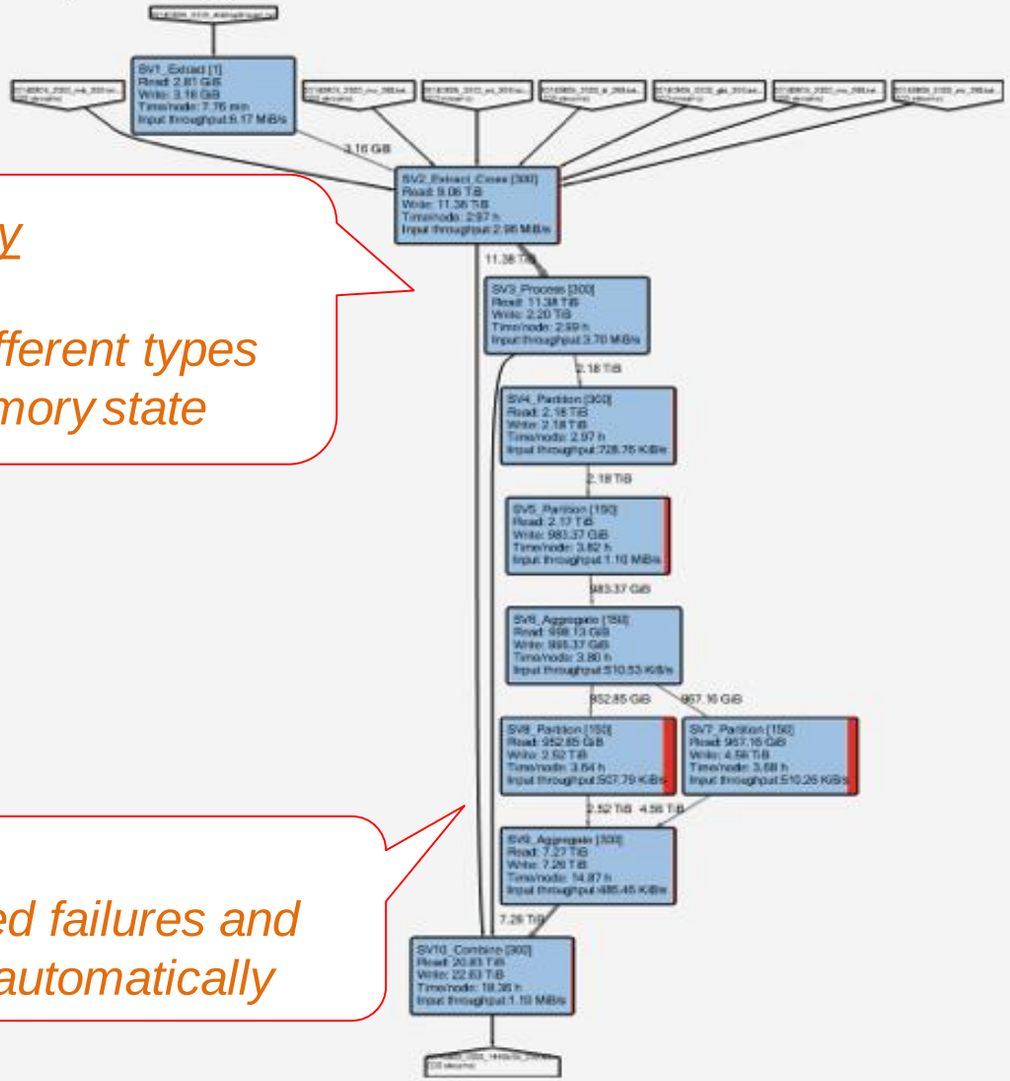
Job Details:

Script | Algebra | VertexDef | Code

| Resources | Debug Stream |

Stage Connection View | Stage Timing View | Vertex RunTime Stats by Stage | Metrics | State History

Display: Progress ▾ Succeeded Failed Running Waiting



High complexity
48 stages
18 joins of 5 different types
21.3 TB in-memory state

Massive scalability
Reads 61TB + Write 61TB
7 billions of input events
6 billions of output events
3000+ long-running tasks

Fault tolerance
Handles both planned failures and unplanned outages automatically

Play

Research and Production: Lessons and Experiences

Research

- Deep insights
- Well founded architecture and methodology
- Simple abstractions
- Fundamental principles

Production

- Keep it simple and operation friendly
- Unexpected *will* happen at scale
- Service mindset: test, validate, deploy, and operate at scale
- No regression, no significant complexity, no unpredictable behavior

Big Data Infrastructure: What's Next

- Convergence of database, systems, programming language, hardware architecture, machine learning and artificial intelligence
- Heterogeneous workloads on heterogeneous hardware: scheduling and resource management
- Continuous, interactive, and rich-structured big data processing

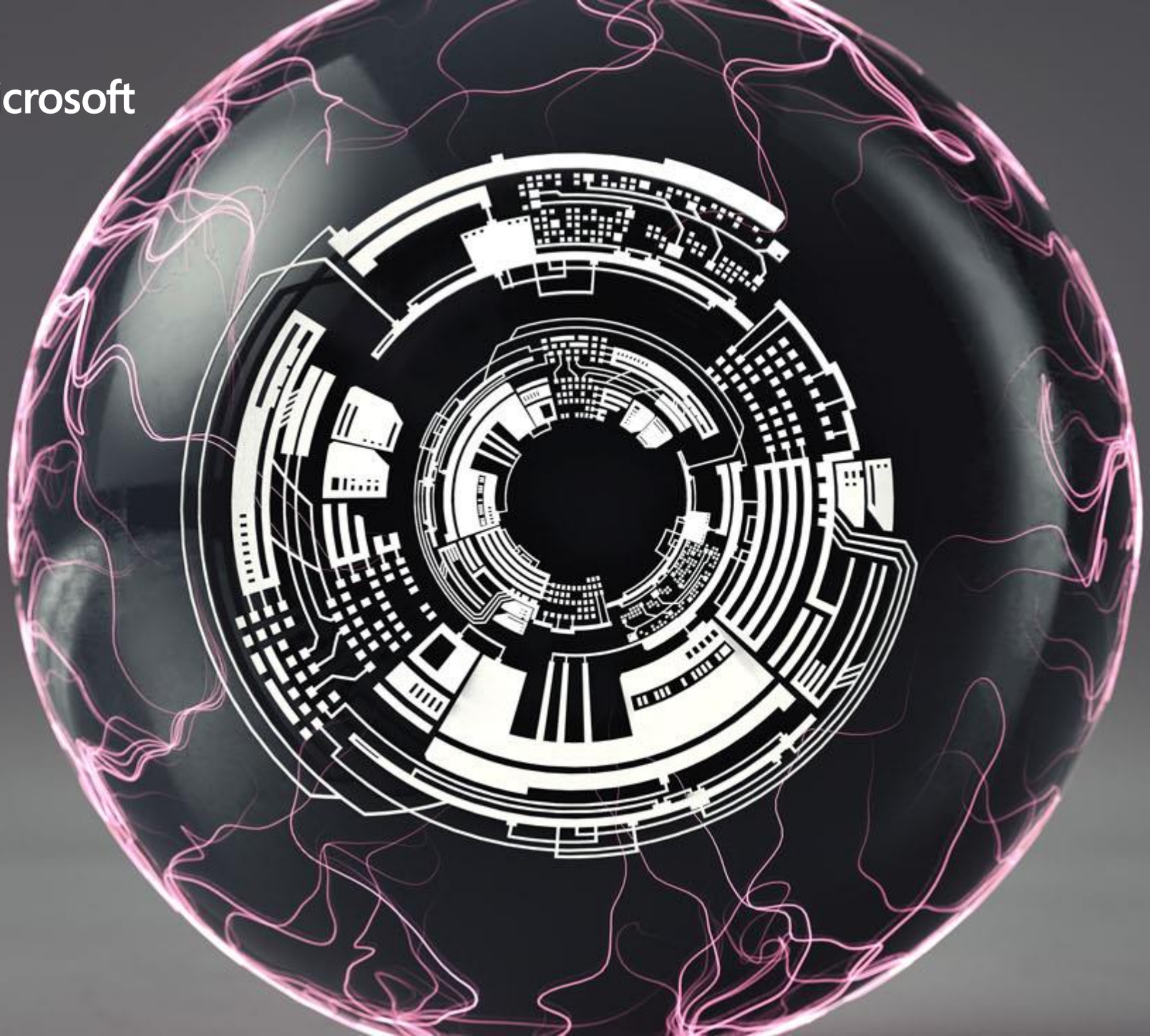
➔ Research and production better together for greater impact

Systems Research Group (MSRA)

Microsoft Big Data Team

Image credits:
http://i.telegraph.co.uk/multimedia/archive/02055/babies-hug_2055065i.jpg

Microsoft



Microsoft Research

Faculty Summit

2016