Optimization of Analytic Data Flows for Next Generation Business Intelligence Applications

Umeshwar Dayal, Kevin Wilkinson, Alkis Simitsis, Malu Castellanos, Lupita Paz
HP Labs
Palo Alto, CA, USA
umeshwar.dayal@hp.com

OUTLINE

– Next Generation Business Intelligence

– Analytic Data Flows

– Challenges in Analytic Data Flow Design and Optimization

– QoX Framework

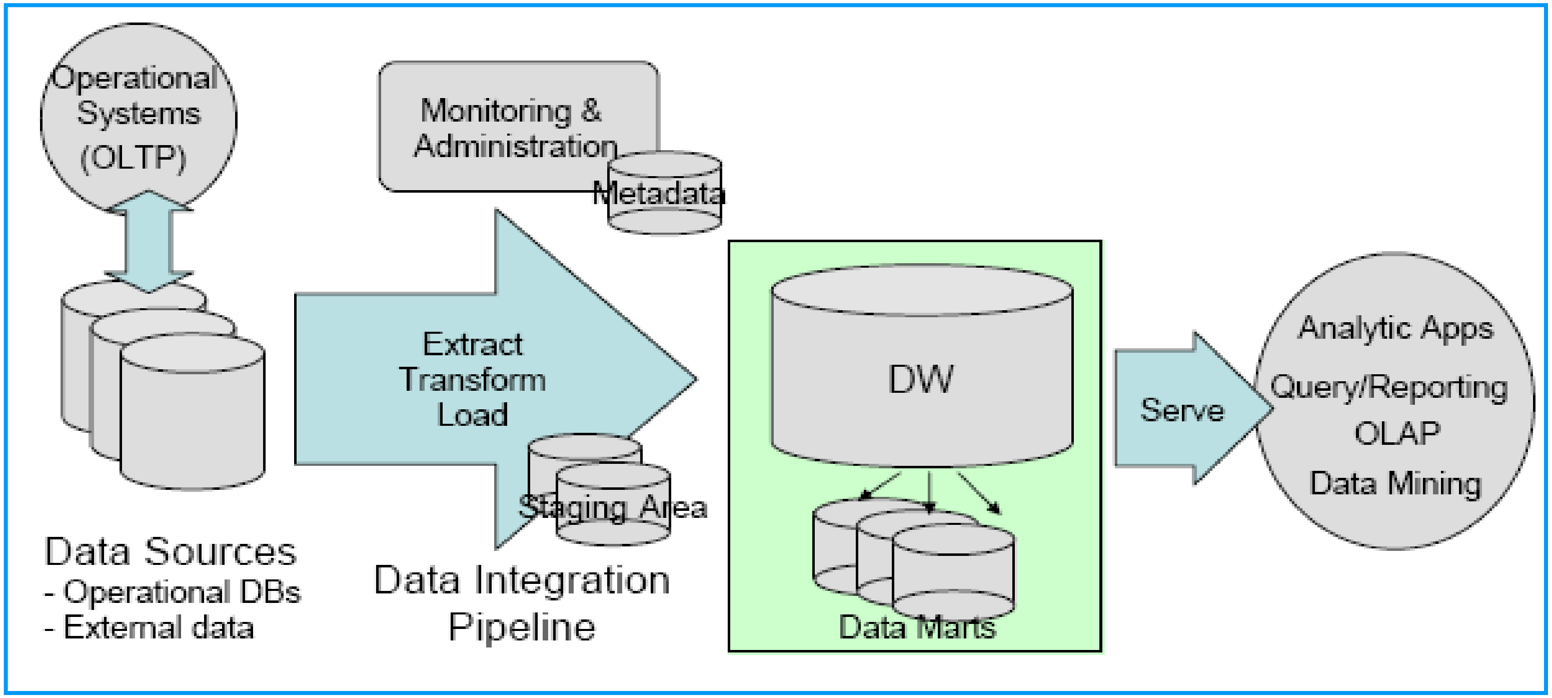
– QoX-driven Optimization of Data Integration Flows

– Micro-benchmarks

– Open Problems

– Summary

TRADITIONAL BUSINESS INTELLIGENCE ARCHITECTURE



BI: Technologies, tools, and practices for collecting, integrating, analyzing, and presenting large volumes of information to enable better decision making

- Focus on enterprise OLTP data sources
- Batch ETL (Extract-Transform-Load) pipeline
- Front-end analytics for strategic decision making

EMERGENCE OF AN INFLECTION POINT…

- Sensors, real time events, unstructured data and web contents have the promise of transforming the way we manage our customers, resources, environments, health,…

…The shift to the **web** makes available unprecedented quantities of structured and unstructured databases about human activities…

…and **sensing technologies** are making available great quantities of data…that provide unprecedented scope and resolution…

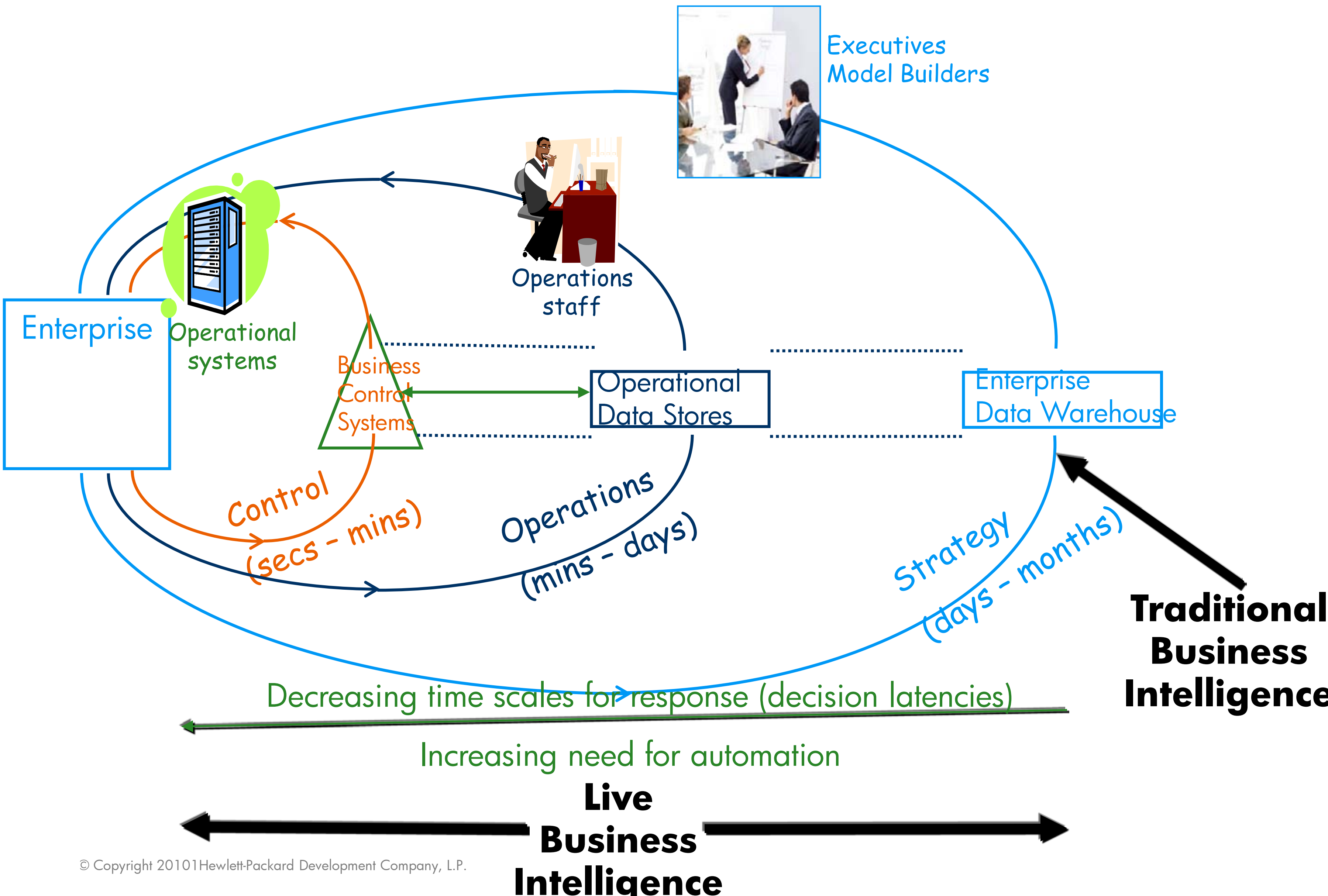
…the *availability of **rich streams of data***, …create(s) an inflection point .. for generating insights and guiding decision making…

…To date, we have only scratched the surface of the potential for learning from these large-scale data sets..

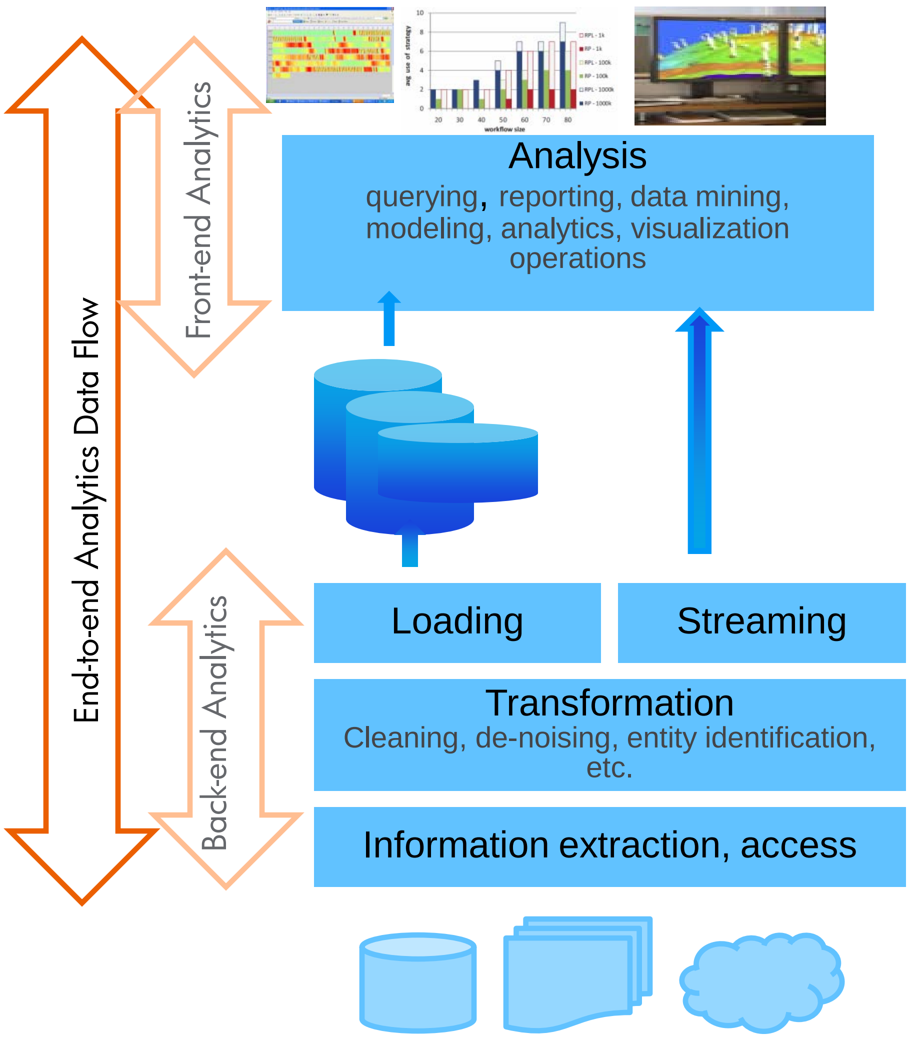
- "From Data to Knowledge to Action – A Global Enabler for the 21st Century",
Computing Community Consortium, August 2010

Analytics, and specifically technologies that enable real-time analysis of large data volumes, social media analytics, and mobile BI will continue to dominate the enterprise agenda - Computer World, Jan 2011 -

Live BI: Decisions *at* any time scale



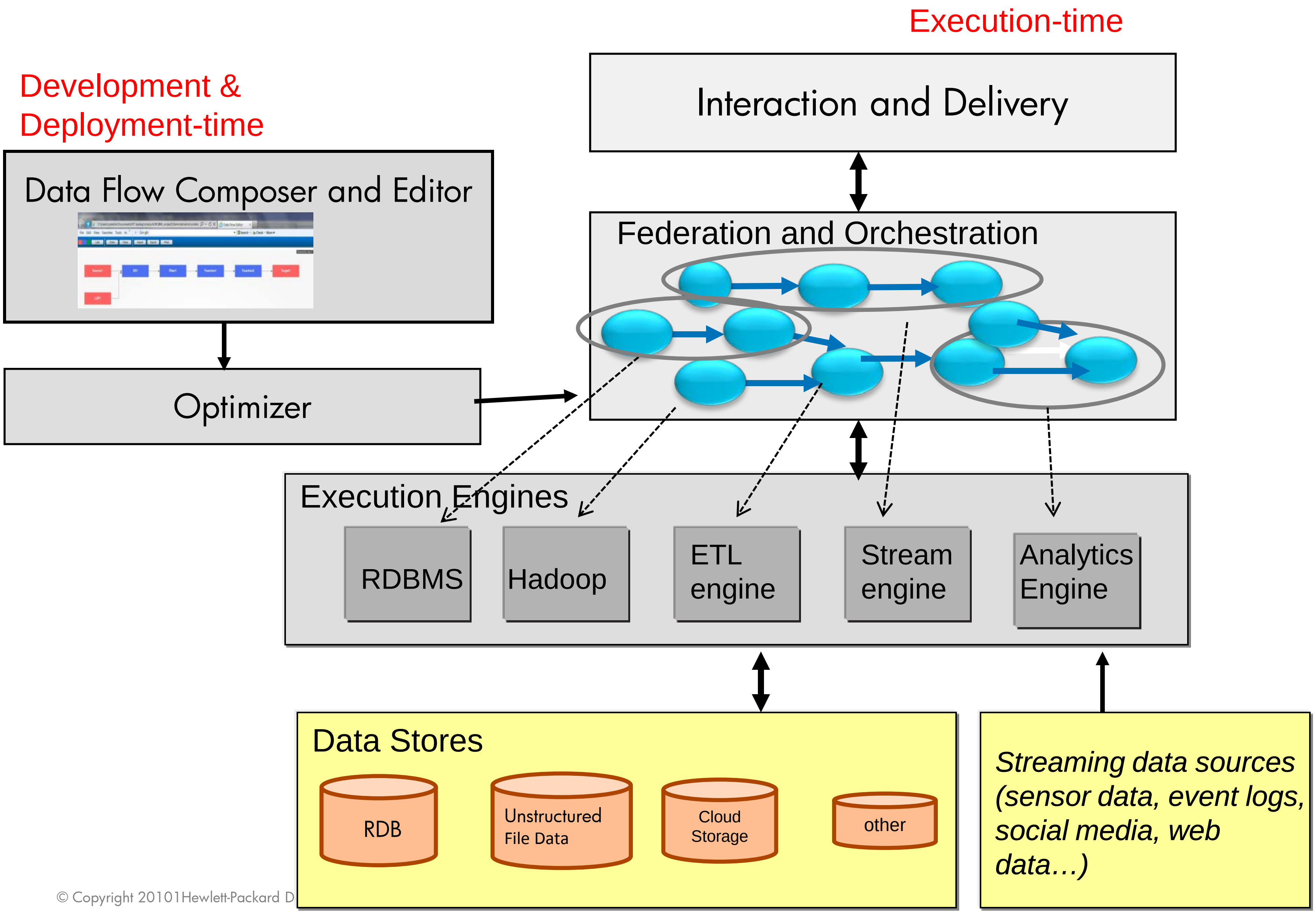
ANALYTICS DATA FLOWS



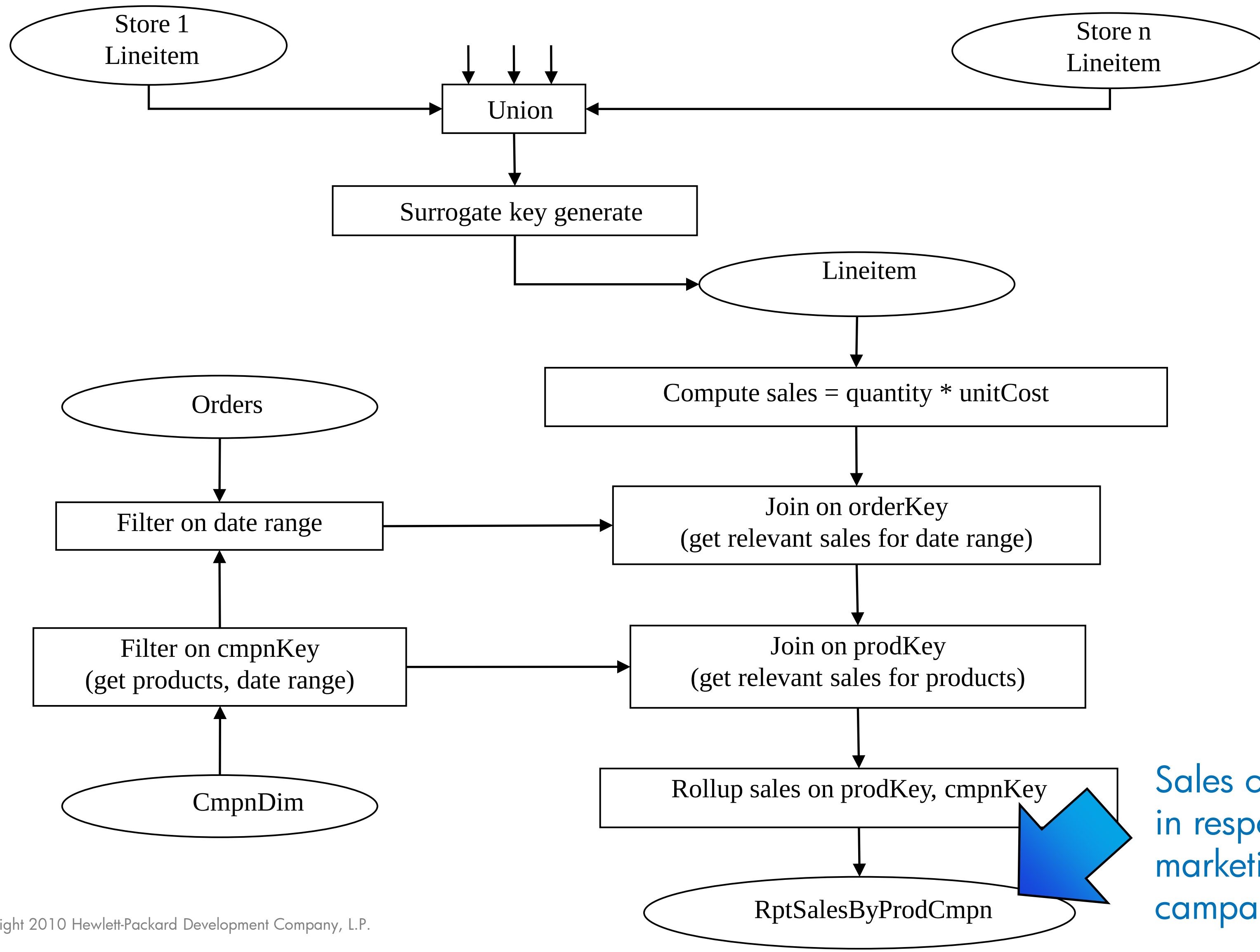
Challenges

- Long development cycles
- Little support for optimization or automation
- Possibly many objectives: Correctness, Fault-tolerance, Scalability, Maintainability, Cost
- Many types of data sources: Structured/ Unstructured, Stored/ Streaming, Internal/ External
- Complex flows, involving ETL, stream processing, queries, analytics, visualization operations
- Possibly executed on multiple engines

NEXT-GENERATION: LIVE BI

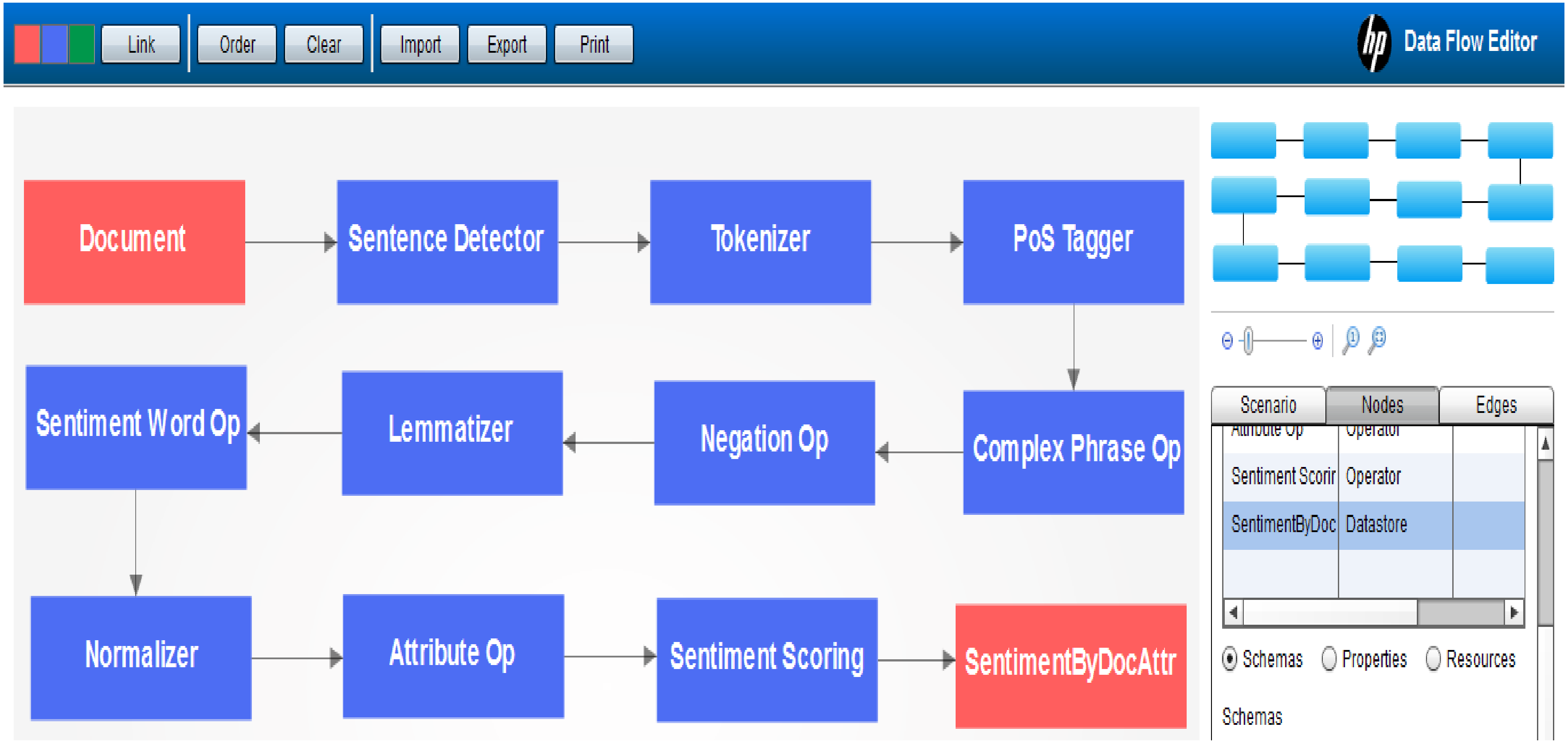


EXAMPLE FLOW 1: STRUCTURED DATA -- TPC-H SOURCES

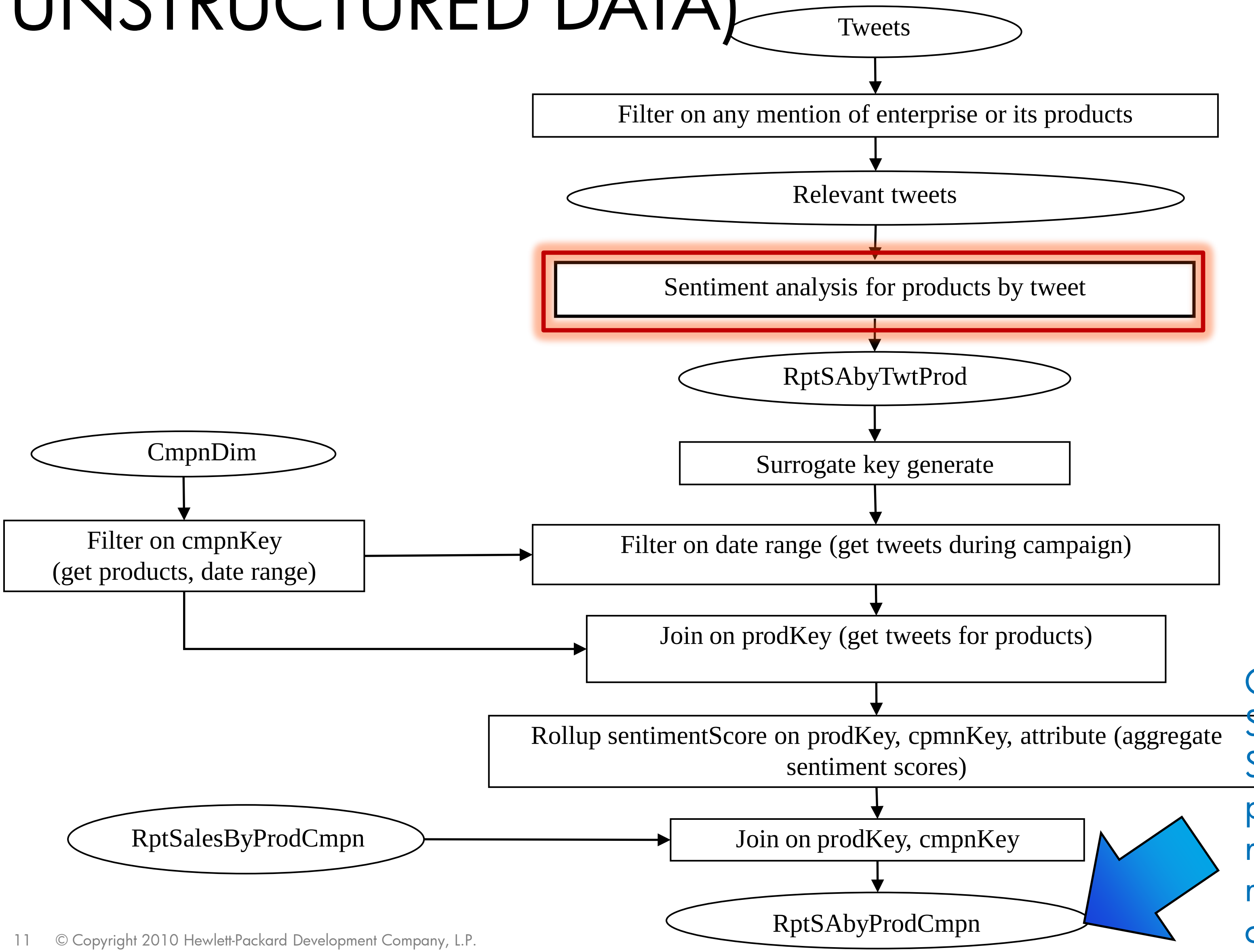


Sales of products in response to marketing campaign

EXAMPLE FLOW 2: UNSTRUCTURED DATA SENTIMENT ANALYSIS

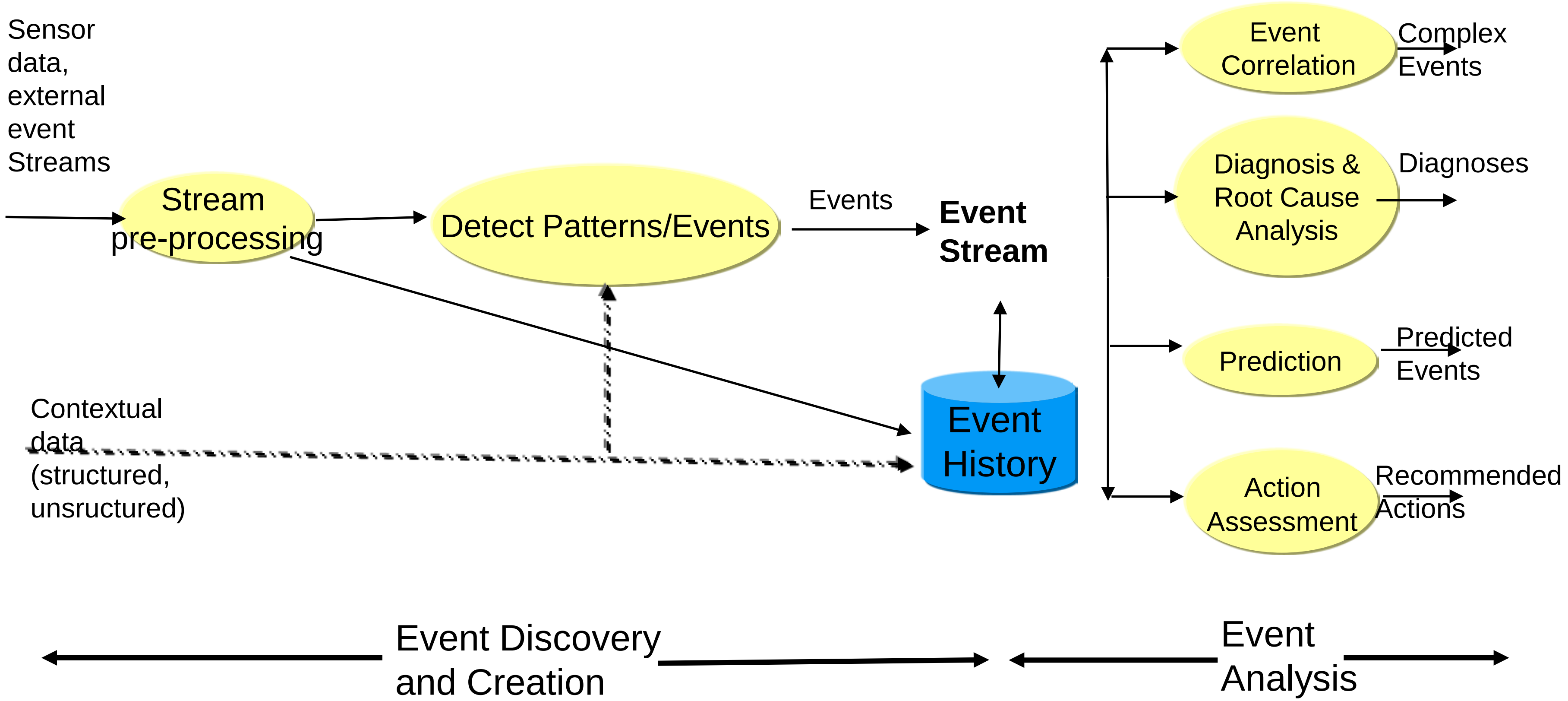


EXAMPLE FLOW 3 (STRUCTURED & UNSTRUCTURED DATA)



Correlate Sales and Sentiments for products in response to marketing campaign

EXAMPLE FLOW 4: EVENT AND STREAM ANALYTICS



FROM DESIGN TO EXECUTION

[Dayal, et al., EDBT'09]
[Wilkinson, Simitsis ER'10]

QoX Objectives

- Performance
- Fault-tolerance
- Maintainability
- Freshness
- Scalability
- Availability
- Cost

Flow design editor

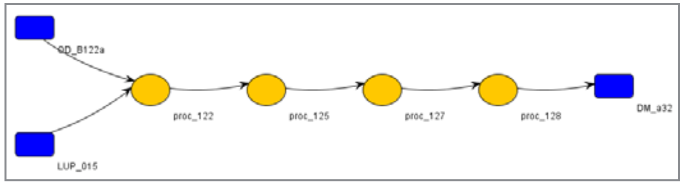
xLM

Optimizer

xLM

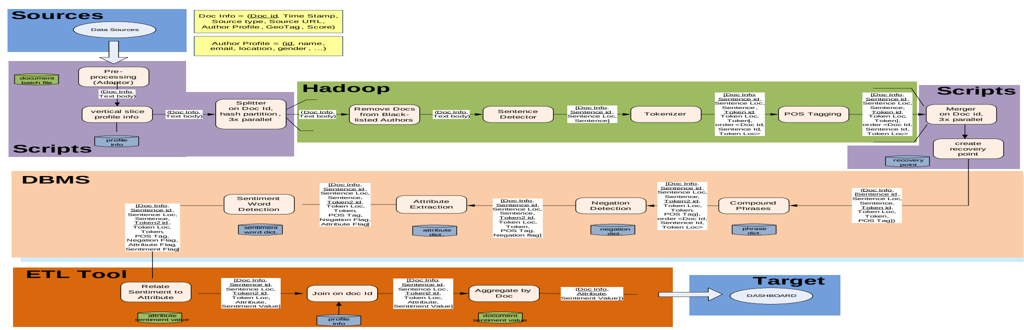
Engine-specific code-gen

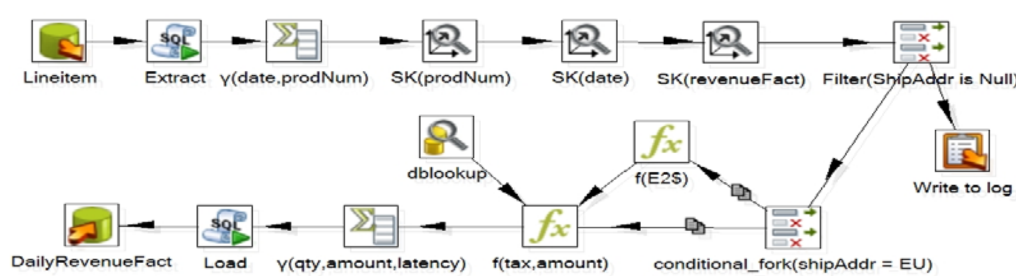
tool-specific execution instructions



Logical Data Flow Graph

Optimized Physical Data Flow Graph





Quality objectives

– functional correctness

- the optimization of the initial flow does not affect the flow semantics or correctness

– performance

- measures the flow completion in terms of time and resources used

– freshness

- the latency between the occurrence of an event at a source system and the reflection of that event at the target system

– reliability

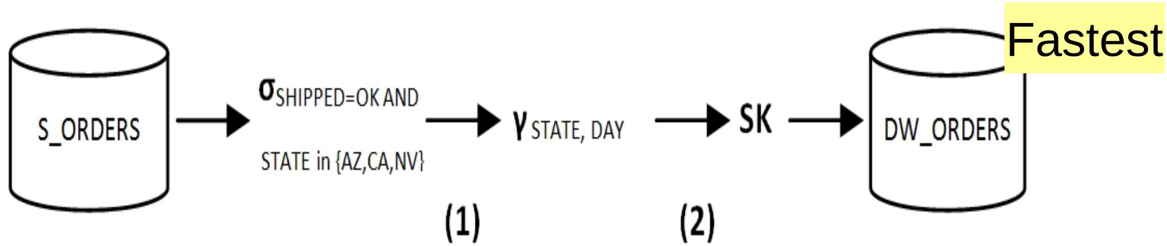
- the ability of flow to complete successfully within the specified time window despite failures

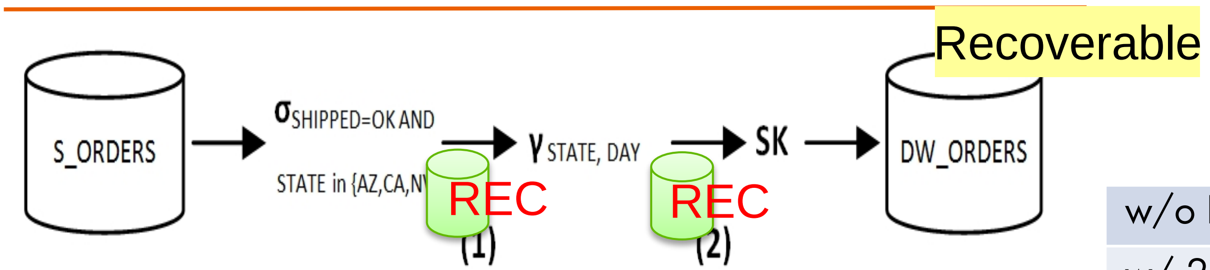
14

QUALITY OBJECTIVES

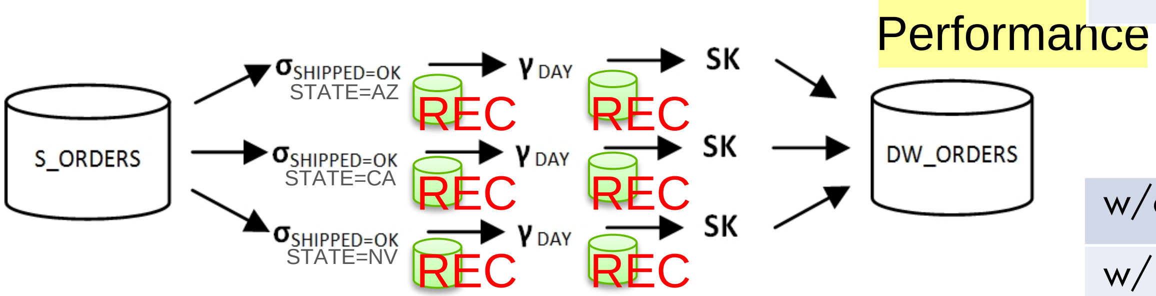
- Performance: usually throughput, total execution time
- Reliability: probability that a flow will complete correctly (~ mean time between failures)
- Recoverability: ability to restore the flow after a failure (~mean time to recover)
- Scalability: ability to process higher volumes of data or higher data rates
- Freshness: latency between source data and data in the DW
- Consistency: accuracy (correctness) of the data in the DW
- Availability: probability that the data is available for queries
- Maintainability: ability to operate at specified service levels
- Flexibility: ability to respond to evolving requirements
- Affordability: cost of integration project
- Traceability: ability to track lineage (provenance) of data
- Auditability: ability to satisfy legal/regulatory compliance requirements

TRADEOFFS AMONG QoX OBJECTIVES

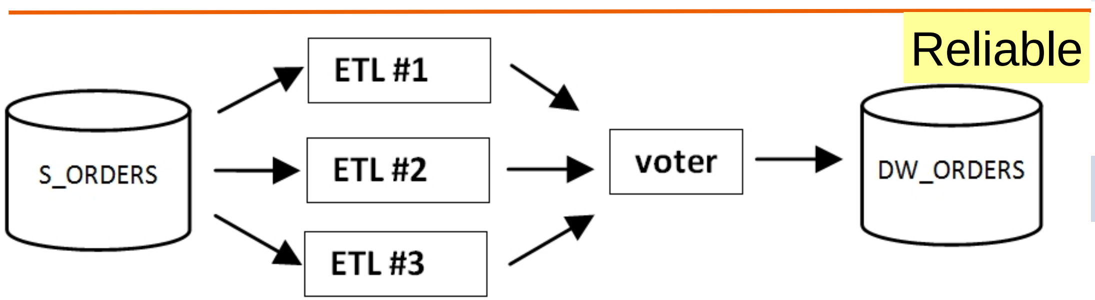




	Perf	Rec	Rel	Cost
w/o RPs	+	-	-	∅
w/ 2 RPs	-	++	-	+



	Perf	Rec	Rel	Cost
w/o RPs	++	-	- -	++
w/ 6 RPs	+	++	- -	++



	Perf	Rec	Rel	Cost
w/o RPs	+	n/a	++	+++

QoX Optimization

[Simitsis, et al., ICDE'10
Simitsis, et al., SIGMOD'09]

– Logical Optimization:
  - Flow restructuring
  - Parallelization, partitioning
  - Collapsing chains
  - Reordering operators
  - Inserting recovery (persistence) points
  - Replication

– Physical optimization
  - Implementations of operators
  - Materialize vs. Virtualize
  - Choice of execution engines
  - Data shipping vs.Function shipping

Optimization for performance and fault-tolerance

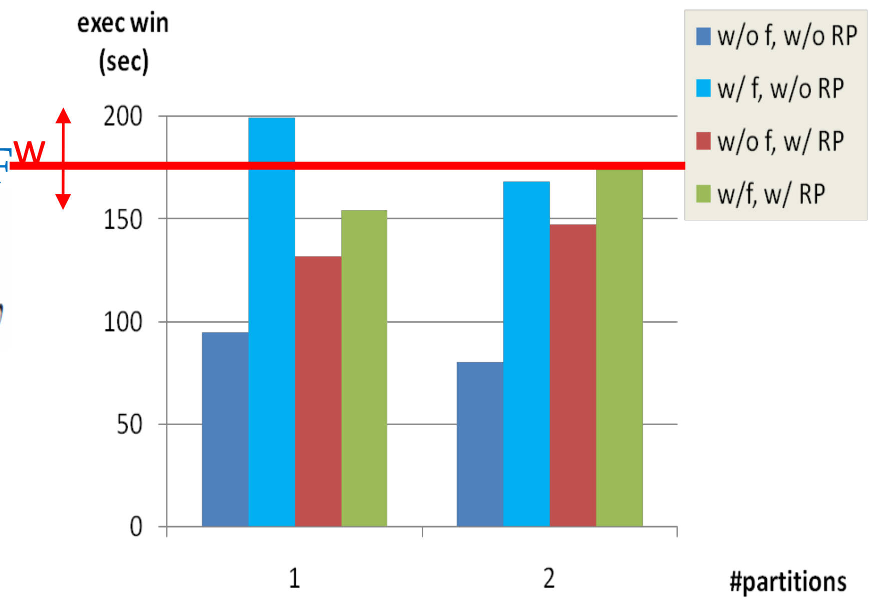
Given a data flow F, objective function OF

$$\mathrm{OF}(F, n, k, w) : \ minimize\ c_{T(F)}, \quad where\ time(F(n,k)) \ < \ w$$

n: input recordset size

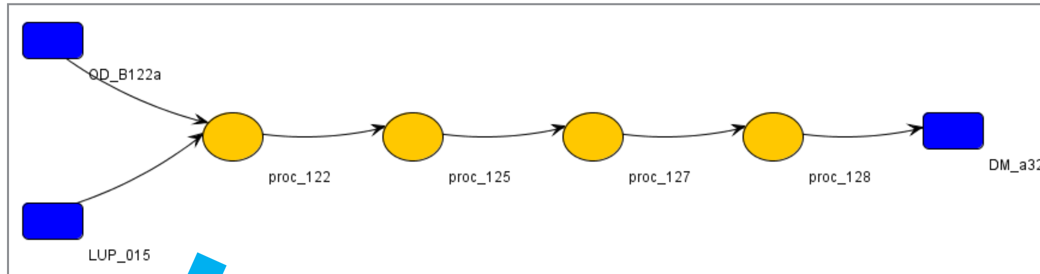
w: execution time window

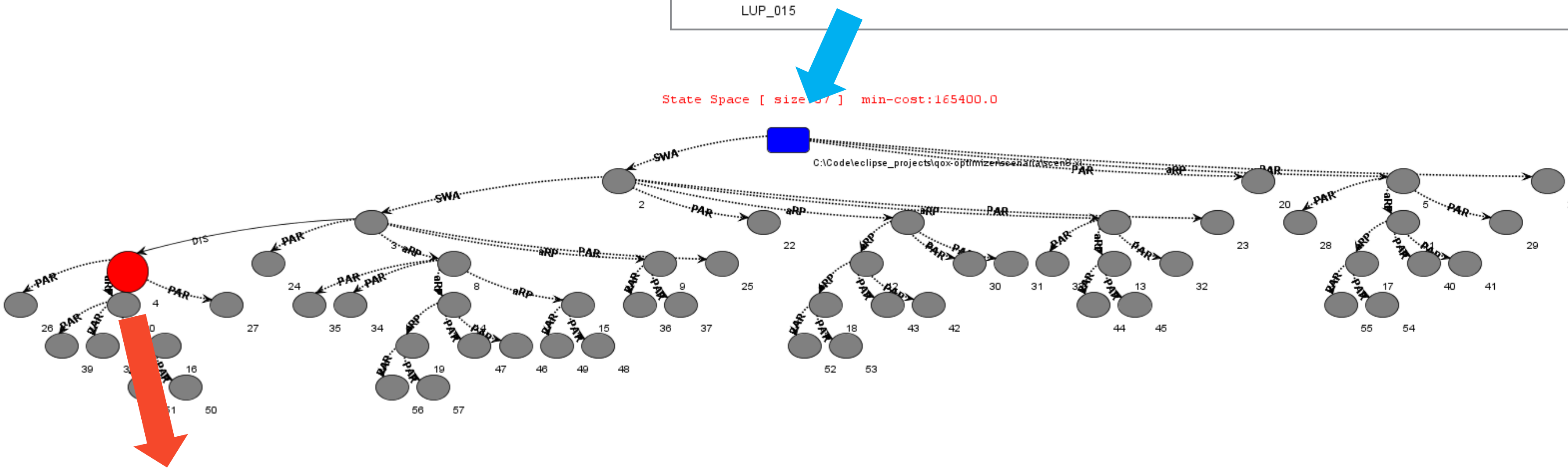
k: number of faults to be tolerated



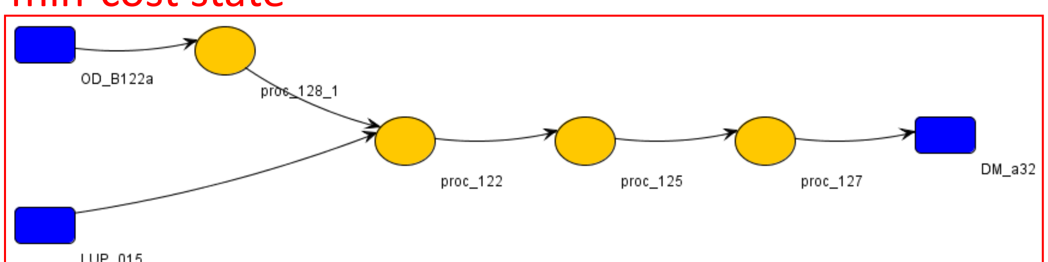
STATE SPACE SEARCH

initial state

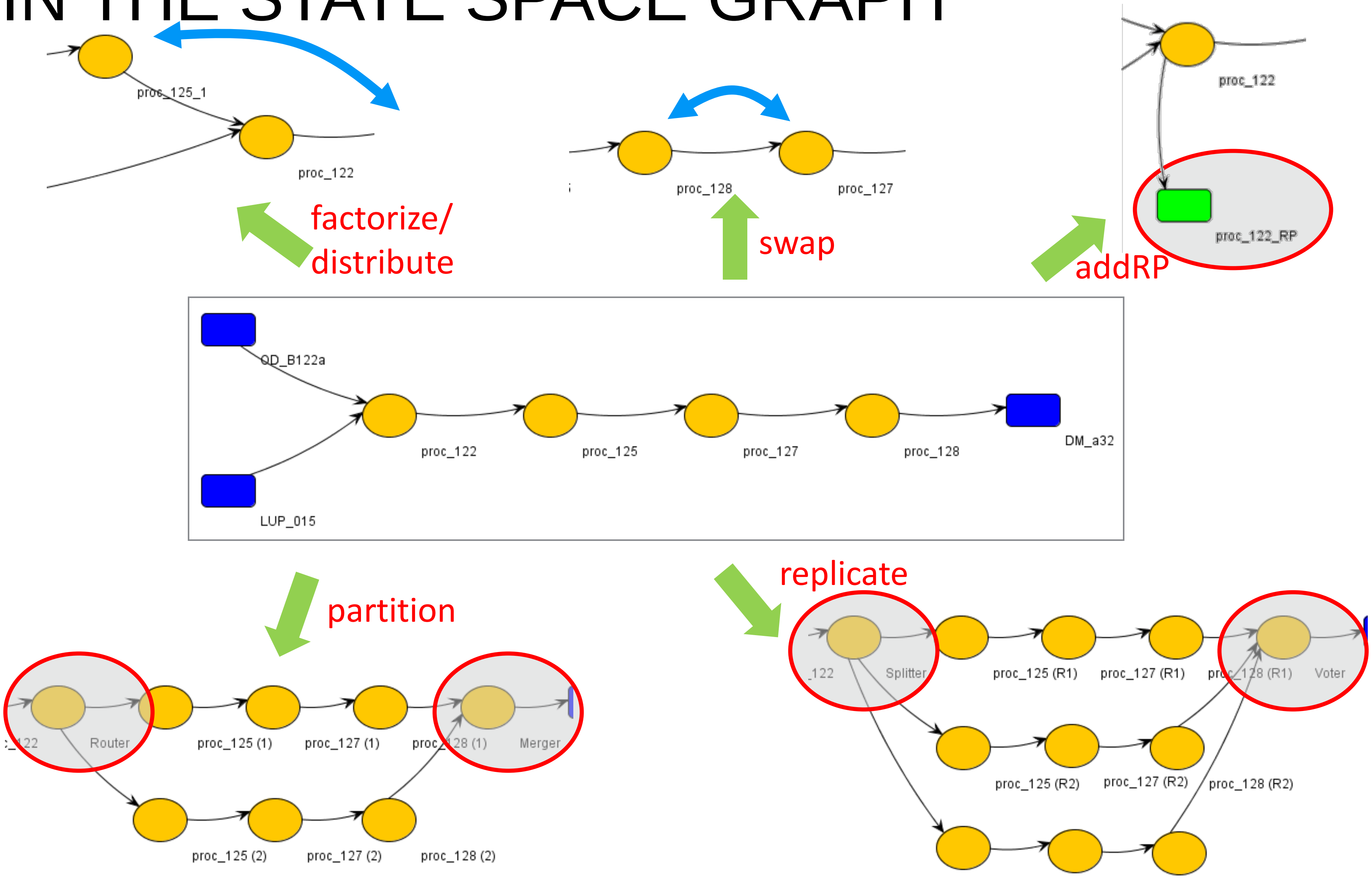




min-cost state



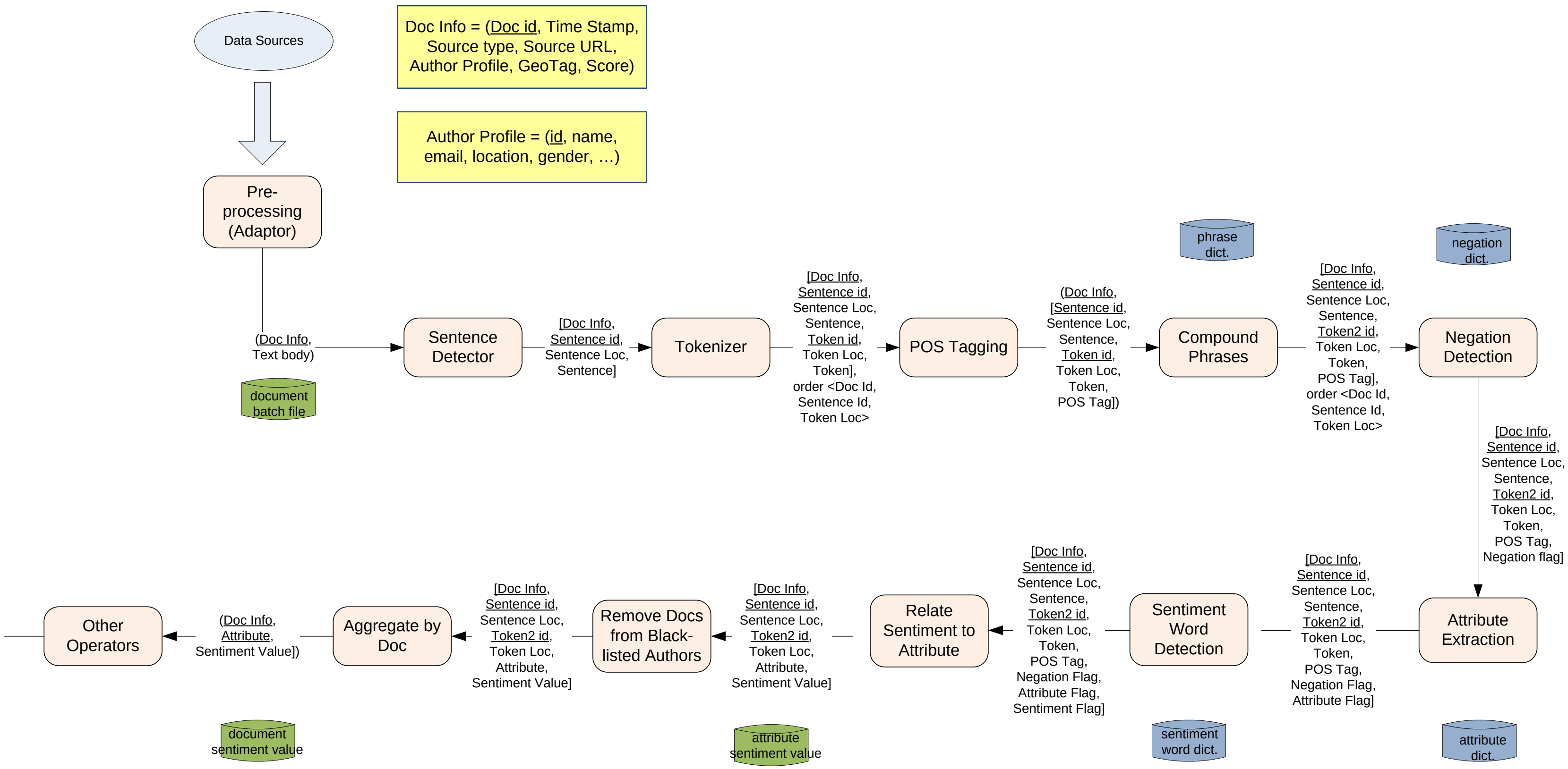
OPTIMIZATION TACTICS: TRANSITIONS IN THE STATE SPACE GRAPH



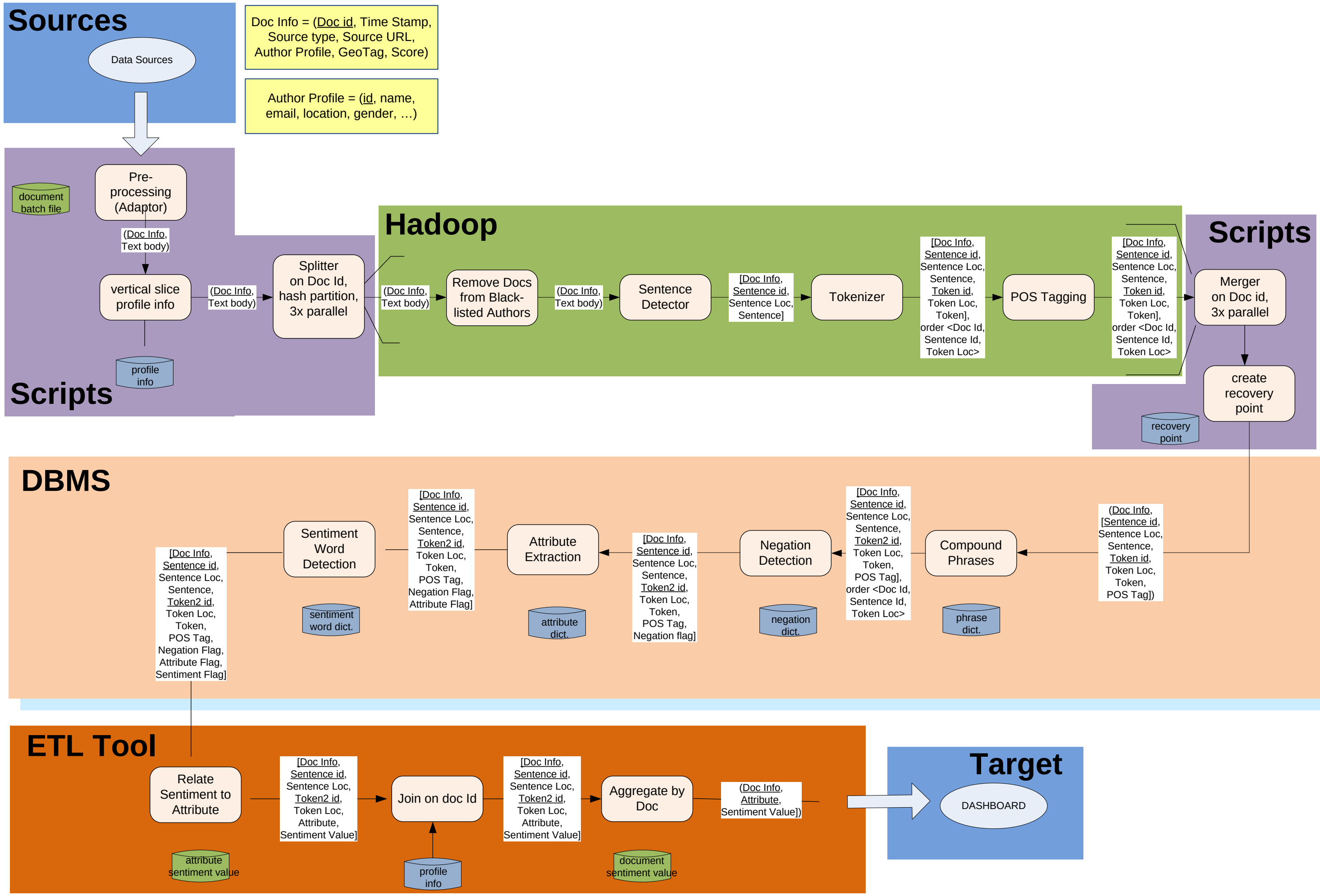
OPTIMIZATION FOR MULTI-ENGINE EXECUTION

- Characterize the execution of operations on different execution engines
  - Implement micro-benchmarks for individual operators
- Learn cost functions to plug into optimization objective
  - Interpolate from micro-benchmarks
  - Compose costs for individual operations into cost of flow
  - Account for resource contention
- Extend transitions considered by the optimizer to include implementation and engine choices
- Extend heuristics for state space search

Example: Sentiment Analysis Flow – Logical Model



Example: Sentiment Analysis Flow – Physical Model



MICRO-BENCHMARKS FOR CONVENTIONAL (ETL, RELATIONAL) OPERATORS

- Experiments on sort, join, surrogate-key, etc.
- Example: Sort
  - os-sort
    - home-grown distributed sort implementation
  - hd-sort
    - Hadoop-based sort (used Pig Latin)
  - pdb-sort
    - used a commercial, parallel DBMS
- Dataset
  - TPC-H lineitem

SF	1	10	100
Size (GBs)	0.76	7.3	75
Rows ($x10^6$)	6.1	59.9	600

Sort (Comparison)

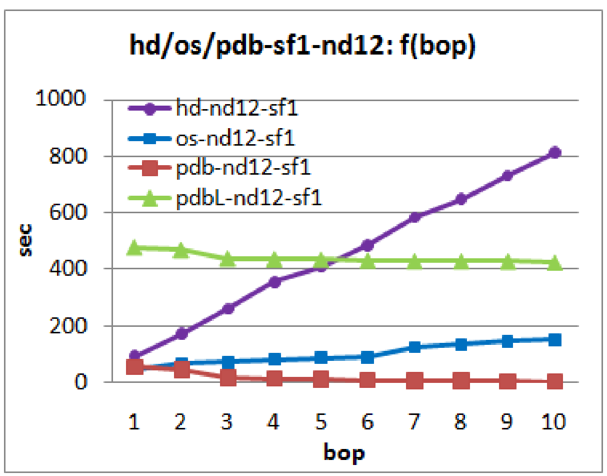
– Hadoop vs. Custom script vs. pDB (with and without loading)

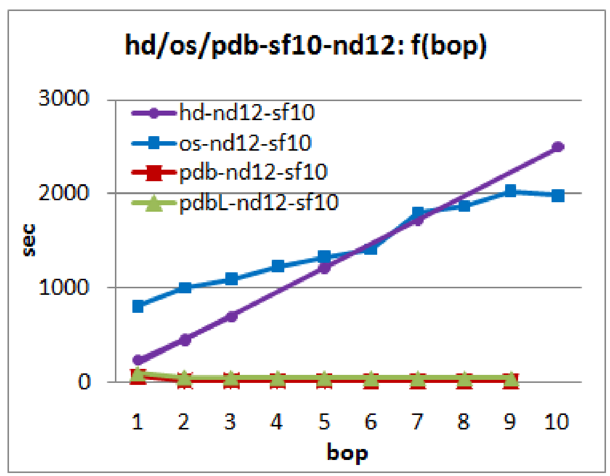
<u>Pig script (hd-sort):</u>

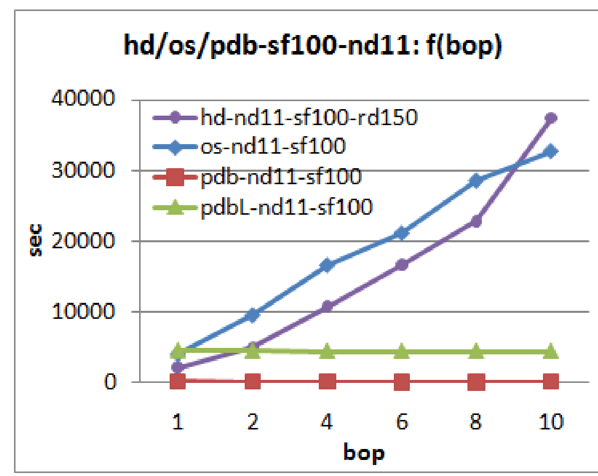
sf$sf = load 'lineitem.tbl.sf$sf'
        using PigStorage('|') as (f1,f2,...,f17);
ord1_$nd = order sf$sf by f1 parallel $nd;
...
ord10_$nd = order ord9_$nd by f10 parallel $nd;
store ord$op_$nd into 'res_sfsf_Ordop-$nd.dat'
        using PigStorage('|');

<u>C and shell scripts (os-sort):</u>

split file: “filesize/#nodes” (leftovers go to chunk #1)
transfer data
sort chunks on nodes
transfer back and merge (hierarchical)
(to be fair with hd-sort and etl-sort, avoid pipelining)

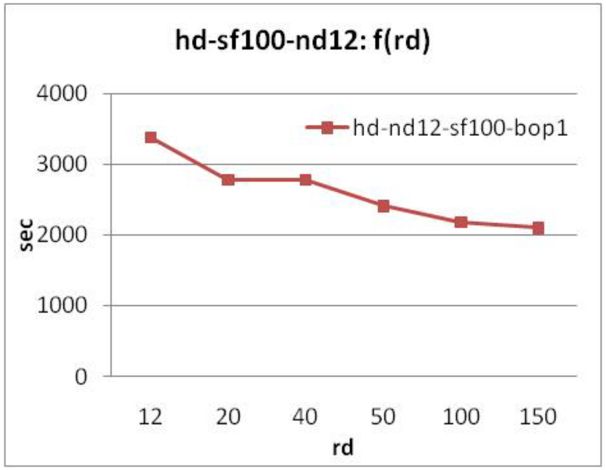




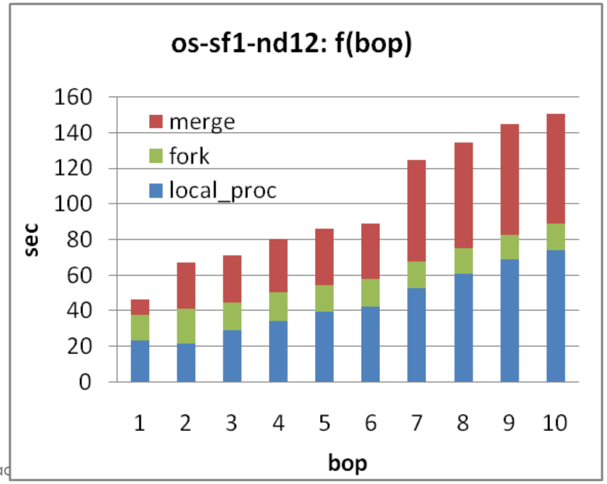


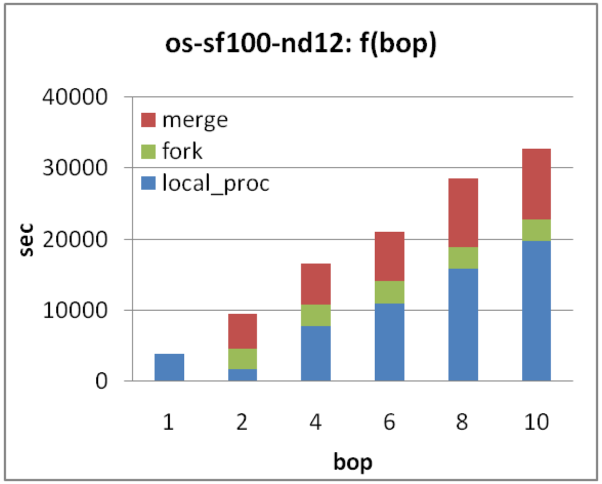
(TPC-H data SF:1,10,100)

Sort (Tuning)



- Hadoop
  - e.g., #reducers
- Shell scripts
  - e.g., analyze sort internal phases



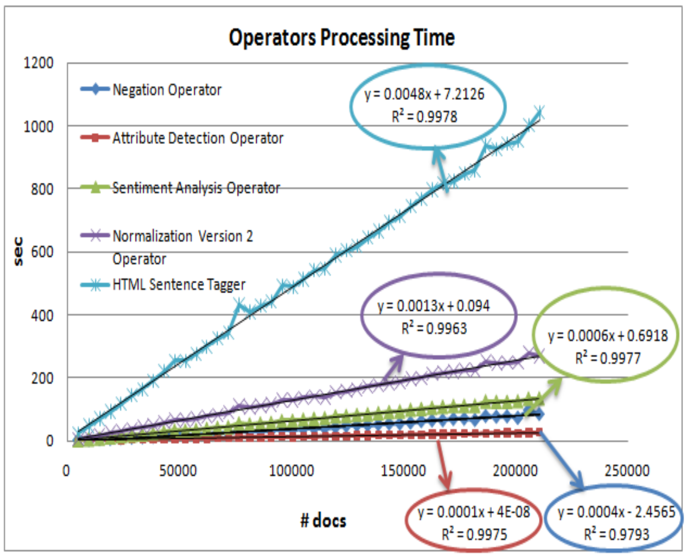


MICRO-BENCHMARKS FOR UNCONVENTIONAL OPERATORS

- Experiments on Sentiment Analysis operators
  - individual operators
  - the whole sentiment analysis flow treated as a single operator
- Implementations
  - Single node
  - Distributed, scripts
  - Distributed, Hadoop
- Datasets
  - Small- to medium-sized document corpus: up to 200,000 documents
  - Large corpus: ~30 million documents

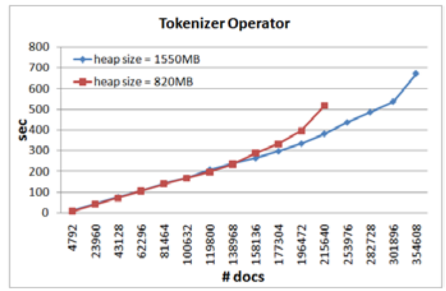
MICRO-BENCHMARKS FOR UNCONVENTIONAL OPERATORS

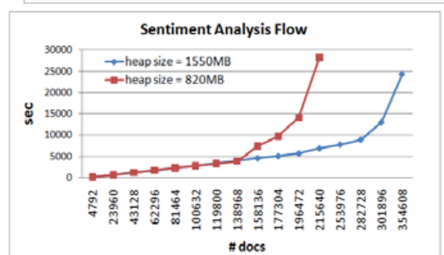
(SENTIMENT ANALYSIS)

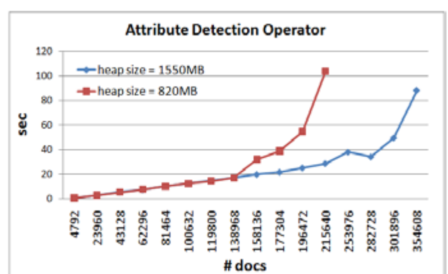


Single node, small to medium-sized corpus

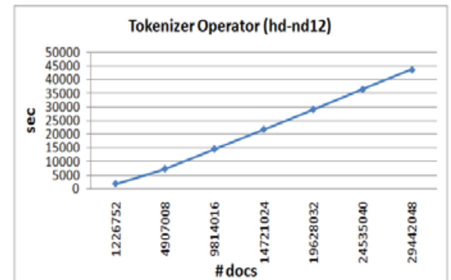
Use linear regression to learn interpolation function

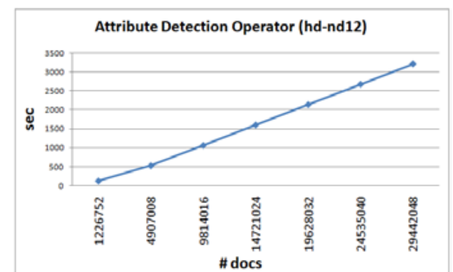


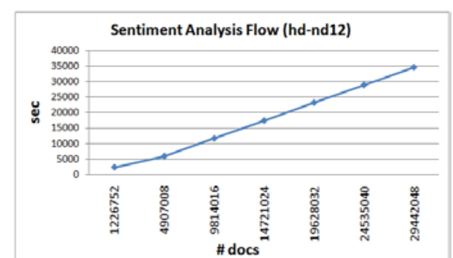




Single node, large corpus; effect of limited memory



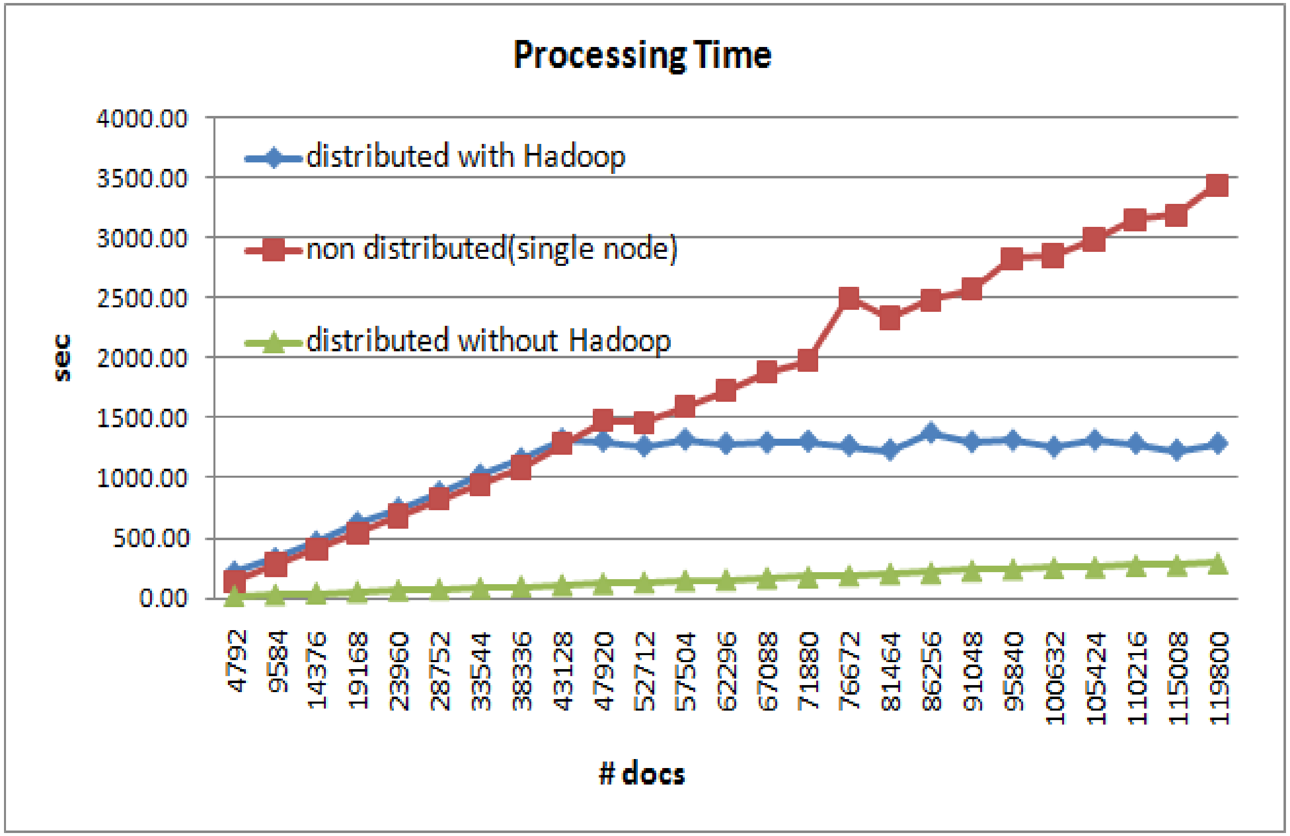




12-node Hadoop cluster

SENTIMENT ANALYSIS (COMPARISON)

– Single node vs. Distributed with Hadoop vs. Distributed without Hadoop



OPEN PROBLEMS: FUTURE WORK

– Micro-benchmarks of other operators: ETL, relational, content analytics, front-end analytics, stream processing

– Cost functions: interpolation, composition

– Benchmarks and objective functions for other QoX measures

– Optimization strategies for physical optimization

  • Assign segments of the flow to execution engines

  • Where, when, and what to materialize

SUMMARY

– Design of data flows for BI applications is a very expensive, largely manual activity: little support for automation, optimization

– Situation gets worse with next gen BI: fuse back-end and front-end operations into end-to-end analytic data flows

– Many different types of sources, more complex operators, multiple execution engines

– Layered methodology: QoX-driven optimization allows trading off different quality objectives

– Need benchmarks and micro-benchmarks to drive the design and optimization

– Many challenges remain

Thank You