Benchmarking ETL Workflows

Alkis Simitsis[1], Panos Vassiliadis[2], Umeshwar Dayal[1], Anastasios Karagiannis[2], Vasiliki Tziovara[2]

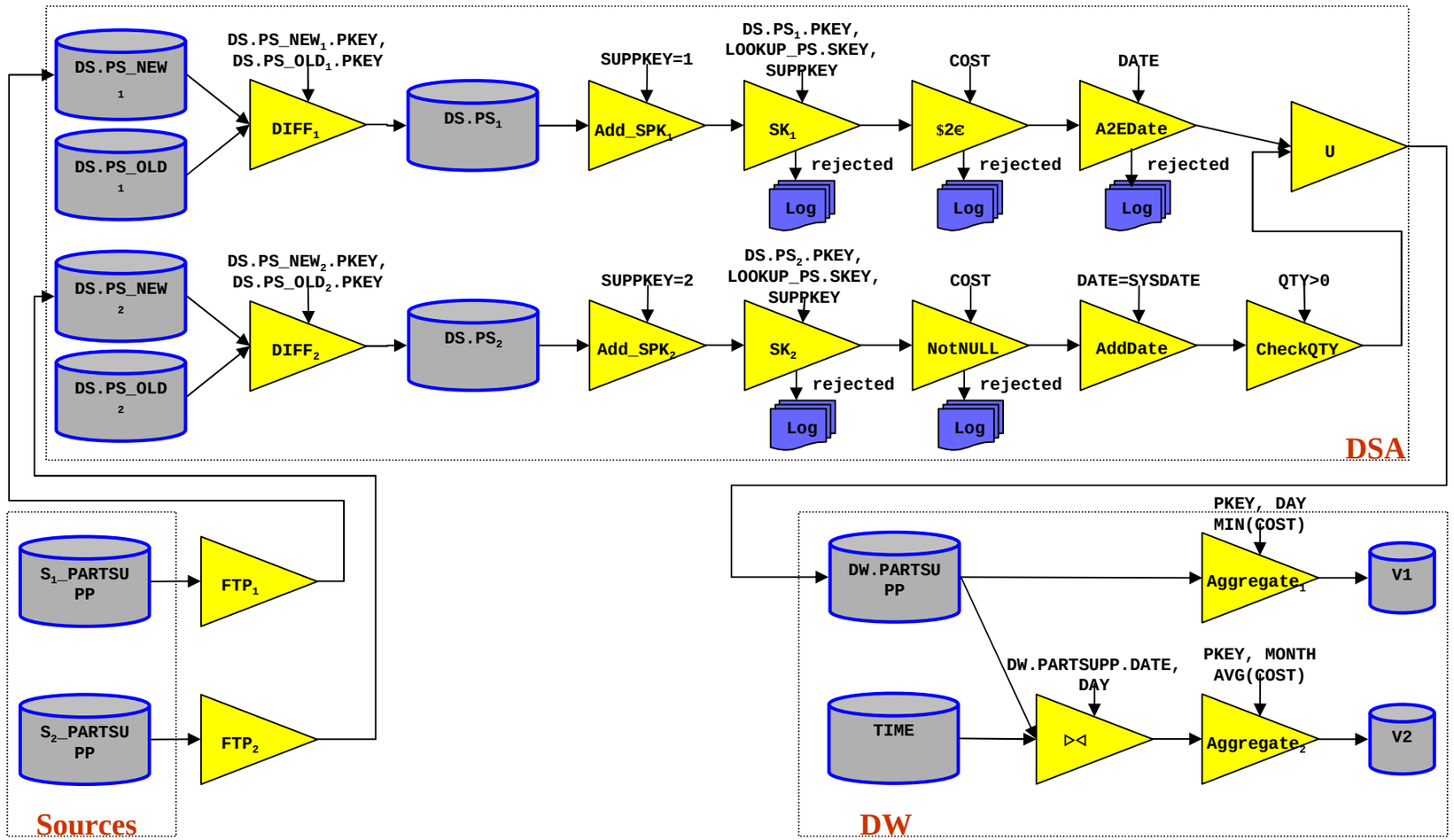
[1] Intelligent Information Management Lab
Hewlett-Packard Labs
Palo Alto, CA, USA

[2] University of Ioannina
Greece

alkis@hp.com

presented by Kevin Wilkinson[1]

ETL workflows



ETL Tools

- Commercial
  - Ab Initio
  - SAP Business Objects
  - IBM WebSphere Information Integration
  - Informatica PowerCenter
  - Microsoft SSIS
  - Oracle Warehouse Builder
  - Pervasive
  - SAS Data Integration Studio
- Open Source
  - Clover
  - Pentaho Kettle
  - Talend

ETL Tools

206 ETL tools, current as of 2003
http://www.dblab.ntua.gr/~asimi/

- Co
  -
  -
  -
  -
  -
  -
  -
  -
  - SA
- Open So
  - Clover
  - Pentaho
  - Talend

1. ActaWorks , Acta Technologies
2. Amadea , ISoft
3. ASC
4. AT
5. Aut
6. Aut
7. Blue
8. Cat
9. CDI
10. Ce
11. Cl
12. Cl
13. CA
14. Cc
15. Cc
16. Cc
17. Cc
18. Cc
19. Cc
20. Cr
21. Cu
22. Cy
23. Dc
24. Dc
25. Dc
26. Data EXTRactor , DogHouse Enterprises
27. Data Flow Manager , Peter's Software
51. DataProF , IT Consultancy Group BV
52. DataPropagator , IBM
53. Dc
54. Dc
55. Dc
56. Dc
57. Dc
58. Dc
59. Dc
60. DE
61. DE
62. DE
63. DE
64. De
65. De
66. DE
67. De
68. DE
69. Di
70. Dc
71. DC
72. DC
73. DT
74. DT
75. eC
76. eIntegration Suite , Taviz Technology
77. Environment Manager , Whitelight Technology
101. iMergence , iMergence Technologies
102. InfluX , Network Software Associates, Inc.
103. Ir
104. Ir
105. Ir
106. Ir
107. Ir
108. Ir
109. Ir
110. Ir
111. Ir
112. IS
113. J
114. K
115. L
116. L
117. M
118. M
119. M
120. M
121. M
122. M
123. M
124. M
125. M
126. MineWorks/400 , Computer Professional Systems
127. MITS , Management Information Tools
151. PL/Loader , Hanlon Consulting
152. PointOut , mSE GmbH
153. Power*Loader Suite , SQL Power Group
154. PowerDesigner WarehouseArchitect , Powersoft
155. PowerMart , Informatica
156. PowerStage , Sybase
157. Rapid Data , Open Universal Software
158. Relational DataBridge , Liant Software Corporation
159. Relational Tools , Princeton Softech
160. ReTarGet , Tominy
161. Rodin , Coglin Mill Pty Ltd.
162. Roll-Up , Ironbridge Software
163. Sagent Solution , Sagent Technology, Inc.
164. SAS/Warehouse Adminstrator , SAS Institute
165. Schemer Advanced , Appligator.com
166. Scribe Integrate , Scribe Software Corporation
167. Scriptoria , Bunker Hill
168. SERdistiller , SER Solutions
169. Signiant , Signiant
170. SIPINA PRO , Diagnos
171. SpeedLoader , Benchmark Consulting
172. SRTransport , Schema Research Corp.
173. StarQuest Data Replicator , StarQuest Software
174. StarTools , StarQuest
175. Stat/Transfer , Circle Systems
176. Strategy , SPSS
177. Sunopsis , Sunopsis
178. SyncSort Unix , Syncsort
179. TableTrans , PPD Informatics
180. Text Agent , Tasc, Inc.
181. TextPipe , Crystal Software Australia
182. TextProc2000 , LVRA
183. Textractor , Textkernel
184. Tilion , Tilion
185. Transporter Fountain , Digital Fountain
186. TransportIT , Computer Associates
187. ViewShark , infoShark
188. Vignette Business Integration Studio , Vignette
189. Visual Warehouse , IBM
190. Volantia , Volantia
191. vTag Web , Connotate Technologies
192. Waha , Beacon Information Technology
193. Warehouse , Taurus Software
194. Warehouse Executive , Ardent Software
195. Warehouse Plus , eNVy Systems
196. Warehouse Workbench , Systemfabrik
197. Web Automation , webMethods
198. Web Data Kit , LOTONtech
199. Web Mining , Blossom Software
200. Web Replicator , Media Consulting
201. WebFOCUS ETL Manager , Information Builders, Inc.
202. WebQL , Caesius Software
203. WhizBang! Extraction Library , WhizBang! Labs
204. Wizport , Turning Point
205. Xentis , GrayMatter Software Corporation
206. XSB , XSB Inc.

LABS hp

Outline

- Motivation
- Goal of the benchmark
  - Effectiveness
  - Efficiency
- Benchmark parameters
  - Experimental parameters
  - Measured effects
- ETL flows
  - Micro-level: activities
  - Macro-level: workflows
- Specific scenarios
- Open issues

Outline

- Motivation
- Goal of the benchmark
  - Effectiveness
  - Efficiency
- Benchmark parameters
  - Experimental parameters
  - Measured effects
- ETL flows
  - Micro-level: activities
  - Macro-level: workflows
- Specific scenarios
- Open issues

Motivation

- An ETL benchmark can be used
  - as a comparison method for
    - ETL tools
    - ETL methods (algorithms)
    - ETL designs
  - for experimenting with ETL workflows
    - for optimizing ETL workflows
      - logical [ICDE05, TKDE05] and physical [DOLAP07] optimization
      - QoX-driven optimization [EDBT09, SIGMOD09]
    - what are the important problem parameters & what are the realistic values for them?
    - what test suites should we use?

Motivation

- Existing standards are insufficient
  - TPC-H
  - TPC-DS
- Practical cases are not publishable

  … and hard to find
- We resort in devising our own ad-hoc test scenarios
  - either through a specific set of scenarios
  - or, through a scenario generator (*will not touch this here*)

Outline

- Motivation
- Goal of the benchmark
  - Effectiveness
  - Efficiency
- Benchmark parameters
  - Experimental parameters
  - Measured effects
- ETL flows
  - Micro-level: activities
  - Macro-level: workflows
- Specific scenarios
- Open issues

Goal of this work

- We are interested in understanding
  - The important parameters to be tuned in an experiment & the appropriate values for them
  - The appropriate measures to be measured during an experiment
  - The fundamental families of activities performed in an ETL scenario
  - The frequent ways with which activities and recordsets interconnect in an ETL scenario

Fundamental goals of any ETL flow

- Effectiveness
  - Quality objectives as
    - performance, recoverability, reliability, freshness, maintainability, scalability, availability, flexibility, robustness, affordability, consistency, traceability, auditability
  - Data should respect both database and business rules
  - Typical questions
    - Q1. Does the workflow execution reach the maximum possible level of data *freshness*, *completeness*, *and consistency in the warehouse within the necessary time* (or resource) constraints?
    - Q2. Is the workflow execution *resilient to occasional failures?*
    - Q3. Is the workflow easily *maintainable?*

Fundamental goals of any ETL flow

- Efficiency
  - Typically ETL processes should run within strict time windows
  - Achieving high performance enables other qualities as well
  - Typical questions
    - Q4. How *fast is the workflow executed?*
    - Q5. What degree of *parallelization is required?*
    - Q6. How much *pipelining does the workflow use?*
    - Q7. What *resource overheads does the workflow incur at the source, intermediate* (staging), and warehouse sites?

Outline

- Motivation
- Goal of the benchmark
  - Effectiveness
  - Efficiency
- Benchmark parameters
  - Experimental parameters
  - Measured effects
- ETL flows
  - Micro-level: activities
  - Macro-level: workflows
- Specific scenarios
- Open issues

Experimental parameters

- Parameters for the measurement of ETL workflows:
  - P1. the *size of the workflow*
  - P2. the *structure of the workflow*
  - P3. the *size of input data originating from the sources,*
  - P4. the *workflow selectivity*
  - P5. the values of *probabilities of failure,*
  - P6. the *latency of updates at the warehouse*
  - P7. the required *completion time*
  - P8. the system *resources (e.g., memory, processing power)*
  - *P9. the "ETL workload" and* the *number of instances of the workflows that should run concurrently*

Measures

- Q1. Measures for data freshness and data consistency
  - % data that violate business rules / are not present at the DW
- Q2. Measures for the resilience to failures
  - MTBF, MTTR, #rec_points, resumption type, #replicas, ETL uptime
- Q3. Measures for maintainability (qualitative objective)
  - Flow length, complexity, modularity, coupling
- Q4. Measures for the speed of the overall process
  - Throughput of workflow execution: regular, w/ failures, avg latency per tuple in regular execution
- Q5. Measures for partitioning parallelism
  - Partition type, number/length/data_volume of branches, #partitions,
- Q6. Measures for pipelining parallelization
  - CPU/mem util for flows/operators, #blocking operators, length of the largest and smaller paths containing pipelining operations
- Q7. Measured Overheads
  - Memory consumed at the sources/DW, elapsed time for OLTP/OLAP transactions (w/ or w/o failures)

Outline

- Motivation
- Goal of the benchmark
  - Effectiveness
  - Efficiency
- Benchmark parameters
  - Experimental parameters
  - Measured effects
- ETL flows
  - Micro-level: activities
  - Macro-level: workflows
- Specific scenarios
- Open issues

Micro-macro view of ETL flows

- Micro-level
  - Inside the workflow
  - A "taxonomy" for ETL activities
- Macro-level
  - Infinite possibilities of connecting nodes (activities and recordsets)
  - A set of "design patterns" as abstractions of how frequently encountered ETL graphs look like

Micro level

- Problem
  - derive a set of fundamental classes, where frequently encountered activities can be classified
- Why a taxonomy of ETL activities?
  - Impossible to predict any possible script / algorithm / operator
  - No algebra for ETL available right now
- Not necessary only for the benchmark, useful for other tasks (e.g., optimization, statistics, etc.)

	Transformation Category*	SQL Server Information Services SSIS [7]	DataStage [2]	Oracle Warehouse Builder [9]
Transformation and Cleansing	**Row-level**: Function that can be applied locally to a single row	– Character Map – Copy Column – Data Conversion – Derived Column – Script Component – OLE DB Command – Other filters (not null, selections, etc.)	– Transformer (A generic representative of a broad range of functions: date and time, logical, mathematical, null handling, number, raw, string, utility, type conversion/casting, routing.) – Remove duplicates – Modify (drop/keeps columns or change their types)	– Deduplicator (distinct) – Filter – Sequence – Constant – Table function (it is applied on a set of rows for increasing the performance) – Data Cleansing Operators (Name and Address, Match-Merge) – Other SQL transformations (Character, Date, Number, XML, etc.)
	Routers: Locally decide, for each row, which of the many outputs it should be sent to	– Conditional Split – Multicast	– Copy – Filter – Switch	– Splitter
	Unary Grouper: Transform a set of rows to a single row	– Aggregate – Pivot/Unpivot	– Aggregator – Make/Split subrecord – Combine/Promote records – Make/Split vector	– Aggregator – Pivot/Unpivot
	Unary Holistic: Perform a transformation to the entire data set (blocking)	– Sort – Percentage Sampling – Row Sampling	– Sort (sequential, parallel, total)	– Sorter
	Binary or N-ary: Combine many inputs into one output	Union-like: – Union All – Merge Join-like: – Merge Join (MJ) – Lookup (SKJ) – Import Column (NLJ)	Union-like: – Funnel (continuous, sort, sequence) Join-like: – Join – Merge – Lookup Diff-like: – Change capture/apply – Difference (record-by-record) – Compare (column-by-column)	Union-like: – Set (union, union all, intersect, minus) Join-like: – Joiner – Key Lookup (SKJ)
Extr.		– Import Column Transformation	– Compress/Expand – Column import	– Merge – Import
Load		– Export Column – Slowly Changing Dimension	– Compress/Expand – Column import/export	– Merge – Export – Slowly Changing Dimension

* All ETL tools provide a set of physical operations that facilitate either the extraction or the loading phase. Such operations include: extraction from hashed/sequential files, delimited/fixed width/multi-format flat files, file set, ftp, lookup, external sort, compress/uncompress, and so on.

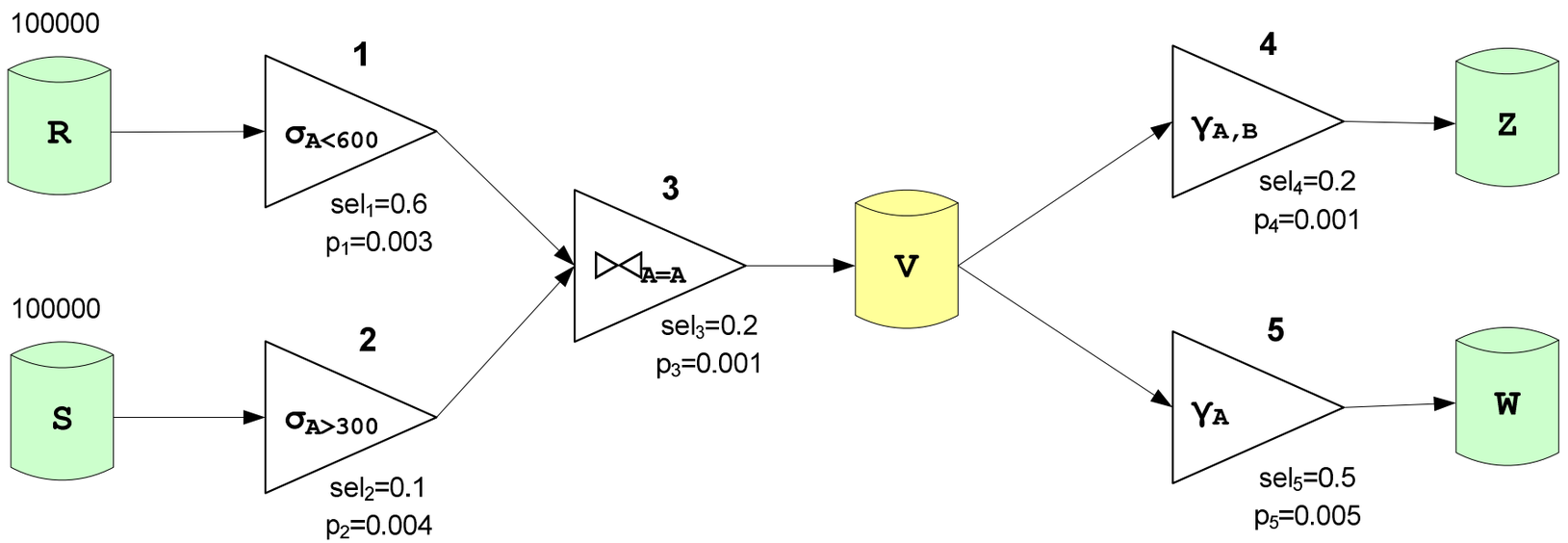
Macro level

- Even harder!
- How to derive a set of typical structural patterns for an ETL scenario?
  - Top down: delve to the fundamental constituents of such a scenario
  - Bottom up: explore scenarios and try to abstract common parts
- We did a little bit of both, and derived a fundamental pattern of structure

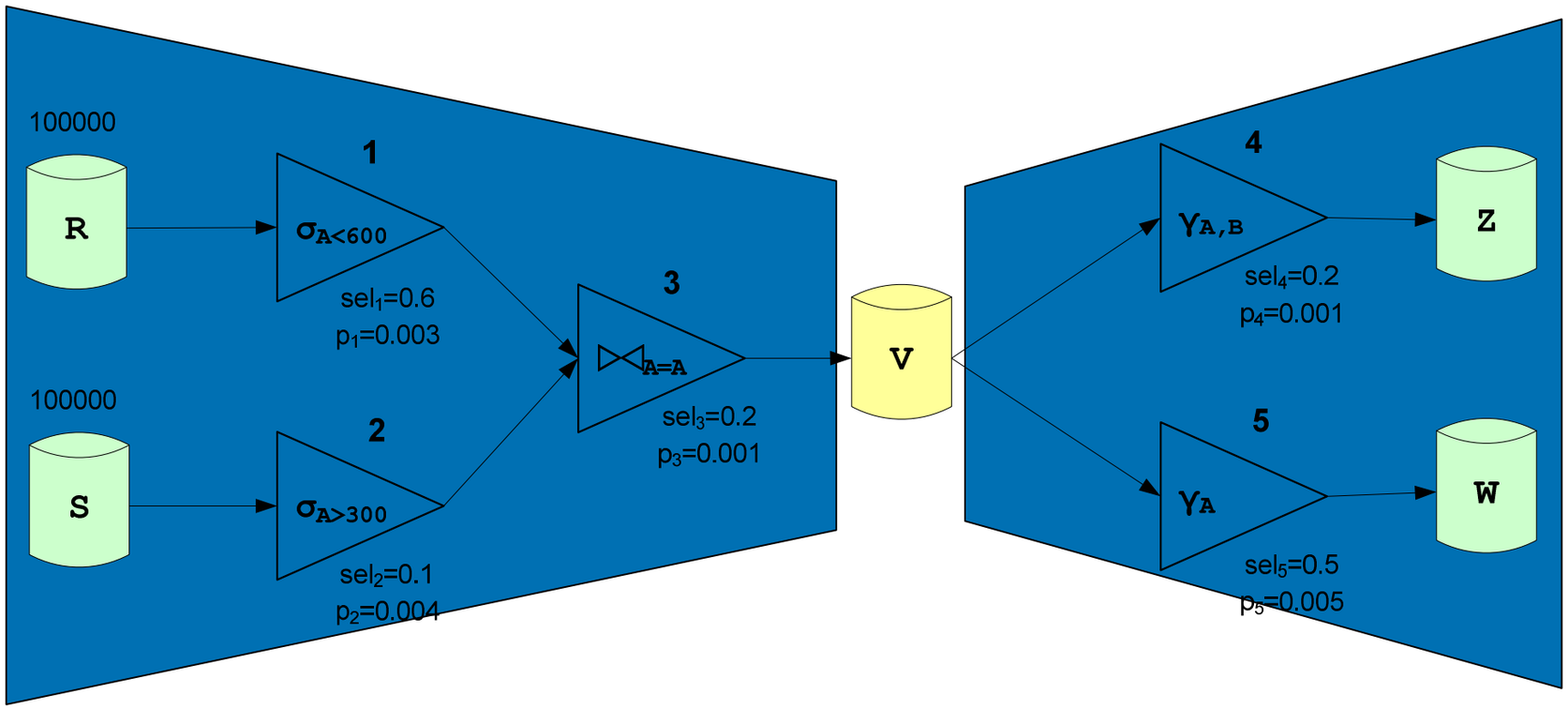
Butterflies to the rescue!

- A butterfly is an ETL workflow that consists of three distinct components:
  - Body
    - a central, detailed point of persistence (e.g., fact or dimension table) that is populated with the data produced by the left wing
  - Left wing
    - sources, activities, intermediate results
    - performs extraction, cleaning and transformation + loads the data to the body
  - Right wing
    - materialized views, reports, spreadsheets, as well as the activities that populate them, to support reporting and analysis

Butterflies to the rescue!



Butterflies to the rescue!



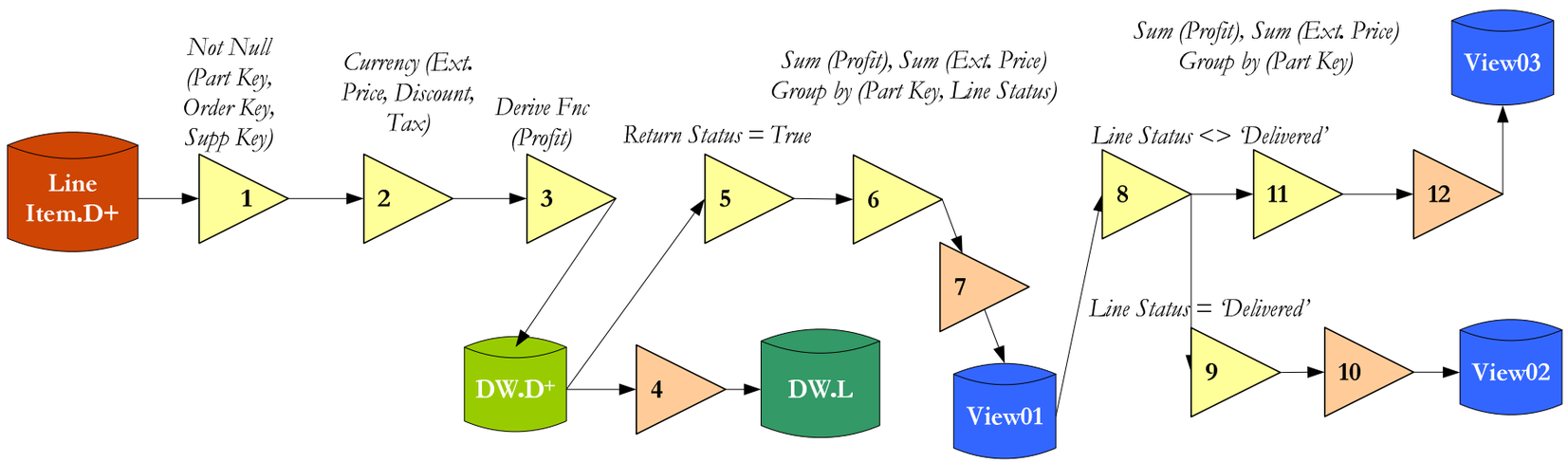
Outline

- Motivation
- Goal of the benchmark
  - Effectiveness
  - Efficiency
- Benchmark parameters
  - Experimental parameters
  - Measured effects
- ETL flows
  - Micro-level: activities
  - Macro-level: workflows
- Specific scenarios
- Open issues

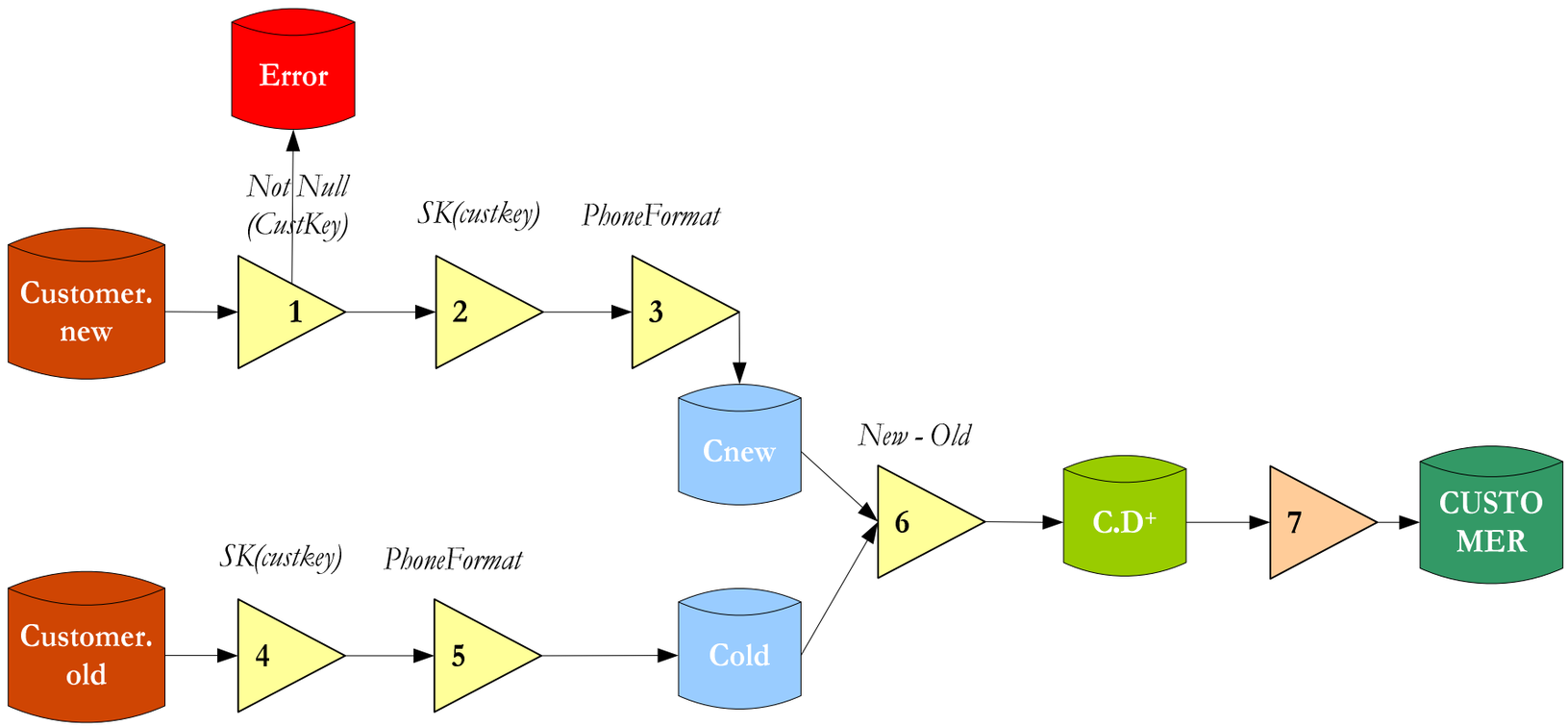
Butterflies to the rescue!

- Butterflies constitute a fundamental pattern of reference
  - Line
  - Balanced butterfly
- Left-winged variants (heavy of the ETL part)
  - Primary flow
  - Wishbone
  - Tree
- Right-winged variants (heavy on the “reporting” part)
  - Fork
- Irregular variants

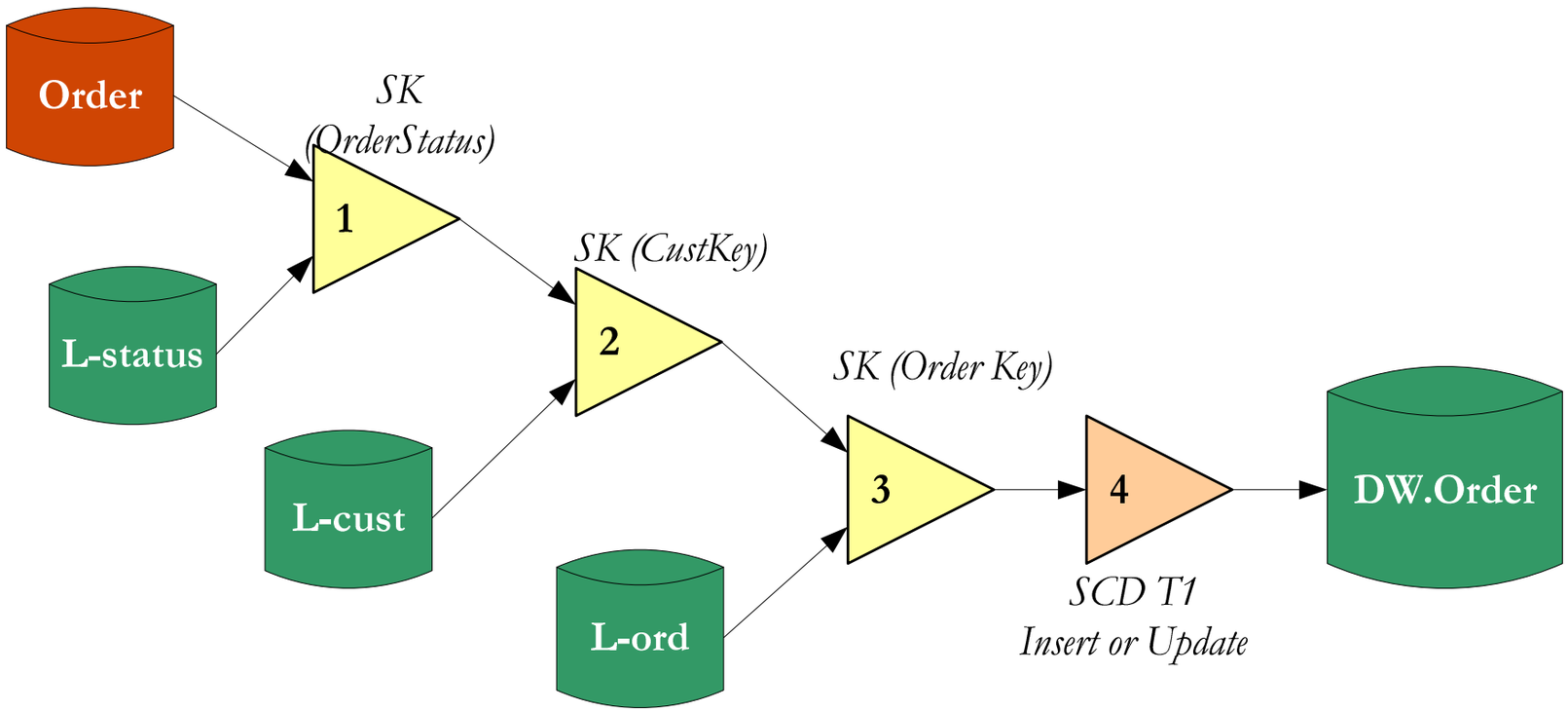
Line


Line Item.D+
Not Null (Part Key, Order Key, Supp Key)
1
Currency (Ext. Price, Discount, Tax)
2
Derive Fnc (Profit)
3
DW.D+
4
Return Status = True
5
Sum (Profit), Sum (Ext. Price) Group by (Part Key, Line Status)
6
DW.L
7
View01
Line Status <> 'Delivered'
8
Sum (Profit), Sum (Ext. Price) Group by (Part Key)
11
12
View03
Line Status = 'Delivered'
9
10
View02

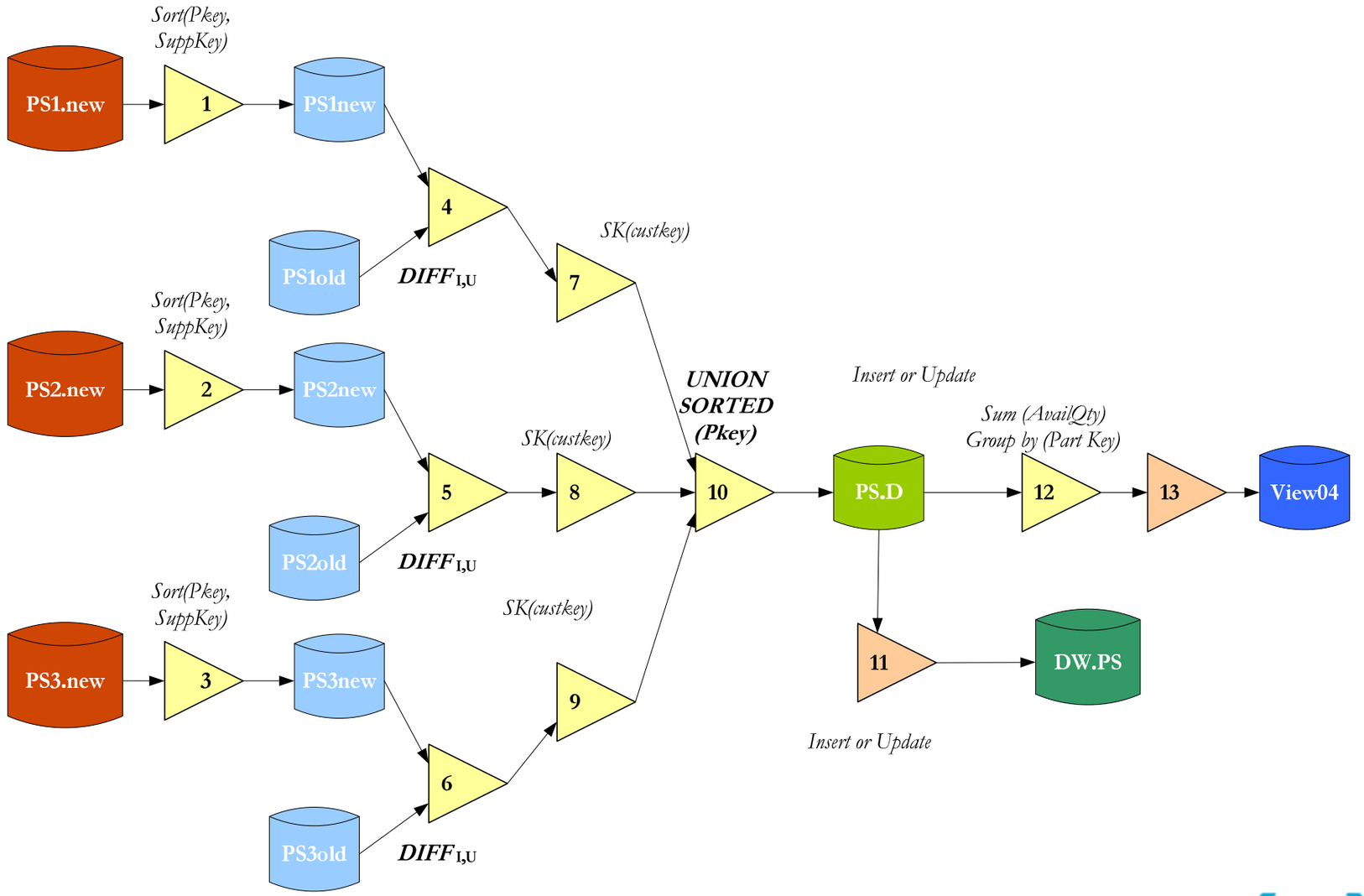
Wishbone


Error
Not Null
(CustKey)
SK(custkey)
PhoneFormat
Customer.
new
1
2
3
Cnew
New - Old
6
C.D+
7
CUSTO
MER
SK(custkey)
PhoneFormat
Customer.
old
4
5
Cold

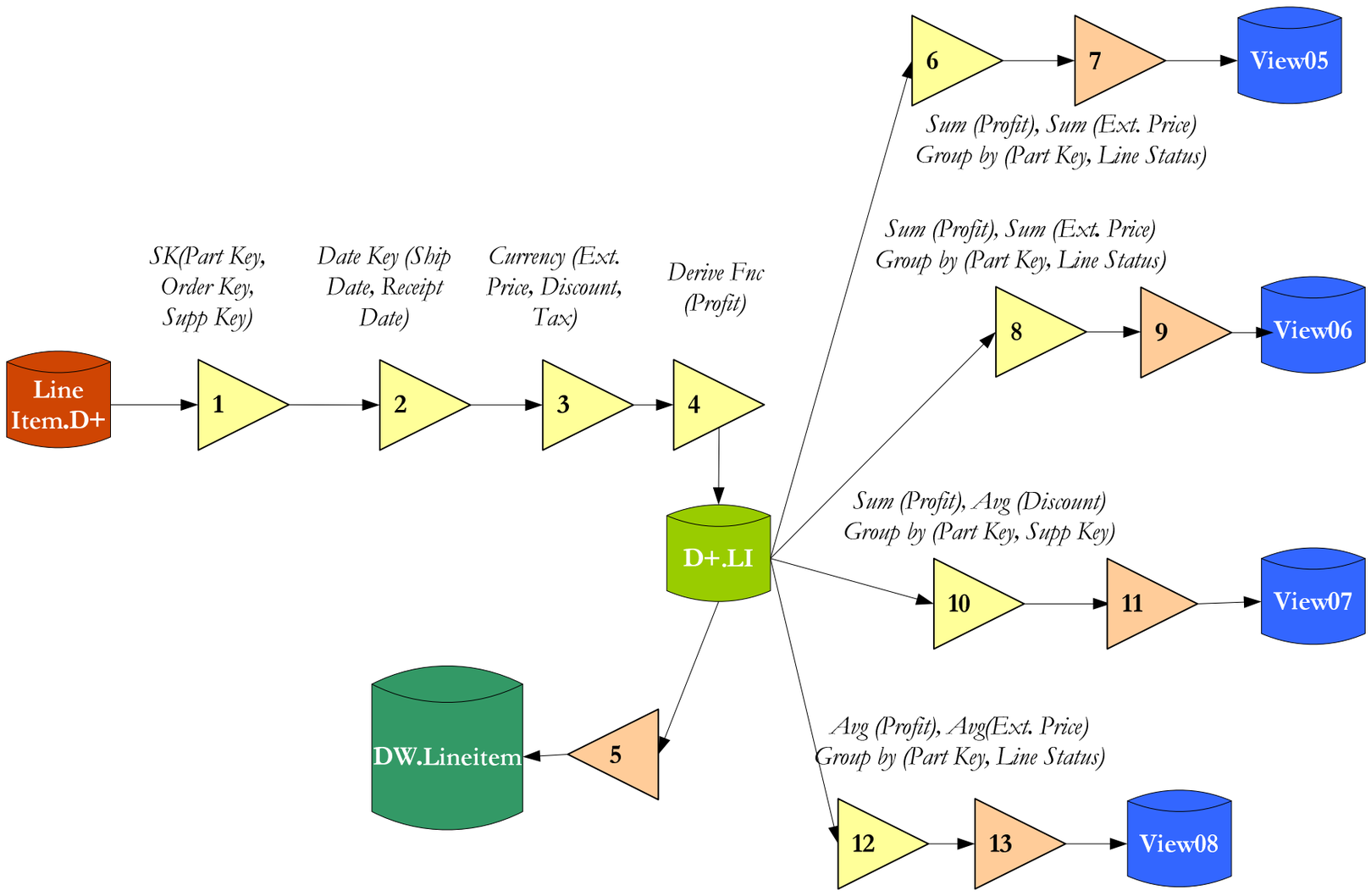
Primary Flow



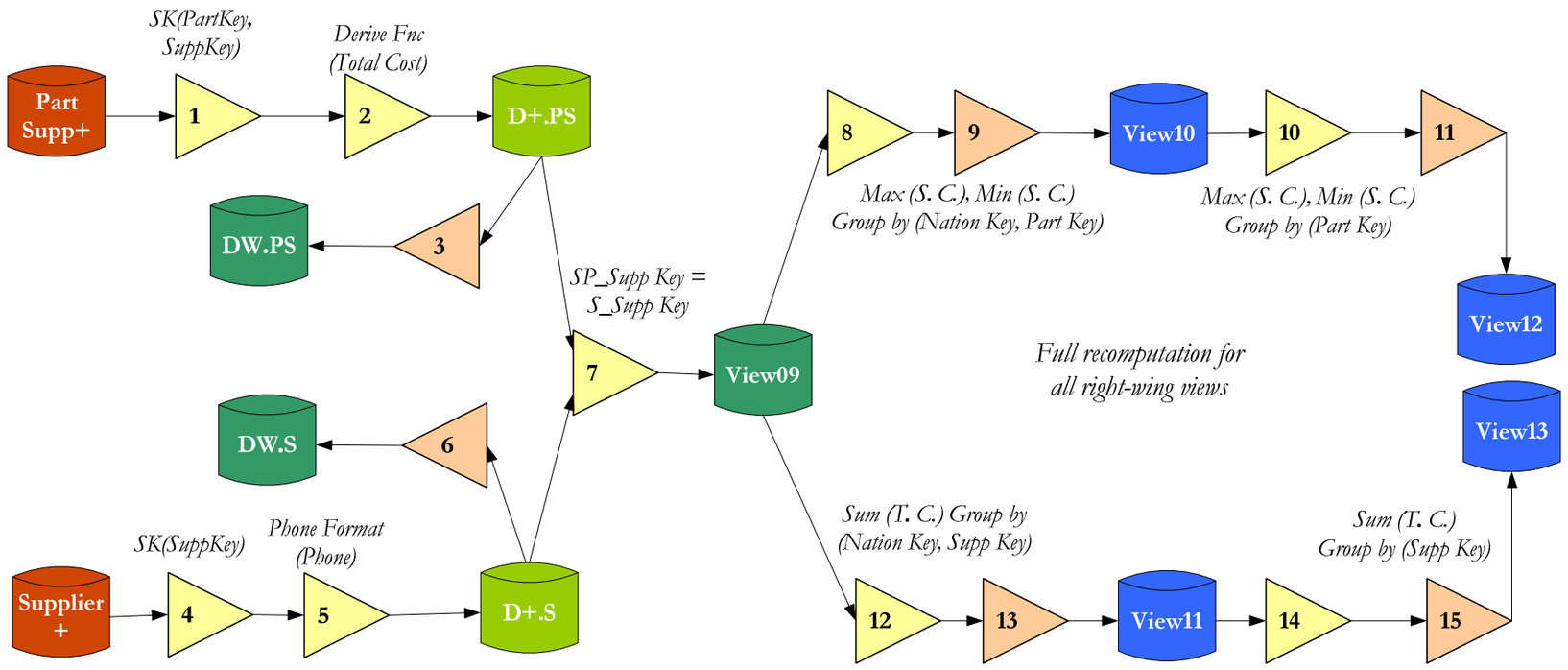
Tree


Sort(Pkey, SuppKey)
PS1.new
1
PS1new
PS1old
4
DIFF I,U
SK(custkey)
7
Sort(Pkey, SuppKey)
PS2.new
2
PS2new
PS2old
5
DIFF I,U
SK(custkey)
8
Sort(Pkey, SuppKey)
PS3.new
3
PS3new
PS3old
6
DIFF I,U
SK(custkey)
9
UNION SORTED (Pkey)
10
Insert or Update
PS.D
Sum (AvailQty) Group by (Part Key)
12
13
View04
11
DW.PS
Insert or Update

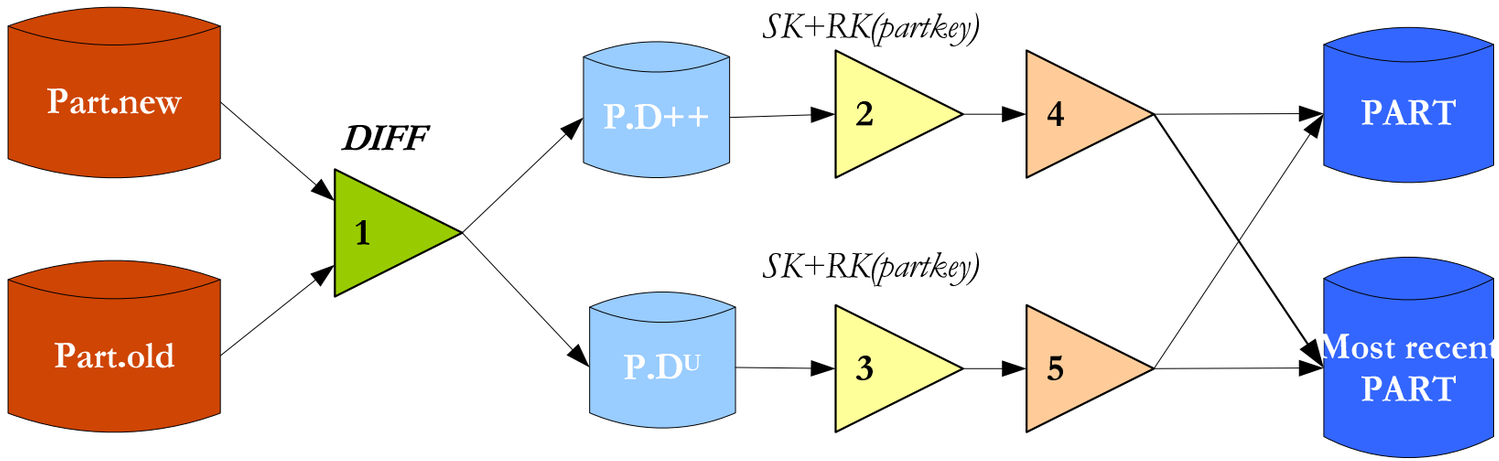
Fork


Line Item.D+
1
SK(Part Key, Order Key, Supp Key)
2
Date Key (Ship Date, Receipt Date)
3
Currency (Ext. Price, Discount, Tax)
4
Derive Fnc (Profit)
D+.LI
5
DW.Lineitem
6
7
View05
Sum (Profit), Sum (Ext. Price)
Group by (Part Key, Line Status)
8
9
View06
Sum (Profit), Sum (Ext. Price)
Group by (Part Key, Line Status)
10
11
View07
Sum (Profit), Avg (Discount)
Group by (Part Key, Supp Key)
12
13
View08
Avg (Profit), Avg(Ext. Price)
Group by (Part Key, Line Status)

Balanced Butterfly



Balanced Butterfly
Slowly Changing Dimension of Type II



Outline

- Motivation
- Goal of the benchmark
  - Effectiveness
  - Efficiency
- Benchmark parameters
  - Experimental parameters
  - Measured effects
- ETL flows
  - Micro-level: activities
  - Macro-level: workflows
- Specific scenarios
- Open issues

Open issues

- Data sizes
  - the numbers given by TPC-H can be a valid point of reference for data warehouse contents
  - Important: fraction of source data over the warehouse contents. Values in the range 0.01 to 0.7?
- Selectivity of the left wing of a butterfly
  - Values between 0.5 and 1.2?
- Failure rates
  - Range of 10^{-4} and 10^{-2}?
- Workflow size
  - Although we provide scenarios of small scale, medium–size and large-size scenarios are also needed

Open issues

- Nature of data
  - not only relational
  - also: XML, unstructured data, spatial data, multimedia, …
- Active vs. off-line modus operandi
- Auxiliary structures and processes
  - e.g., indexes, backup & maintenance scenarios, etc.
- Parallelism and Partitioning

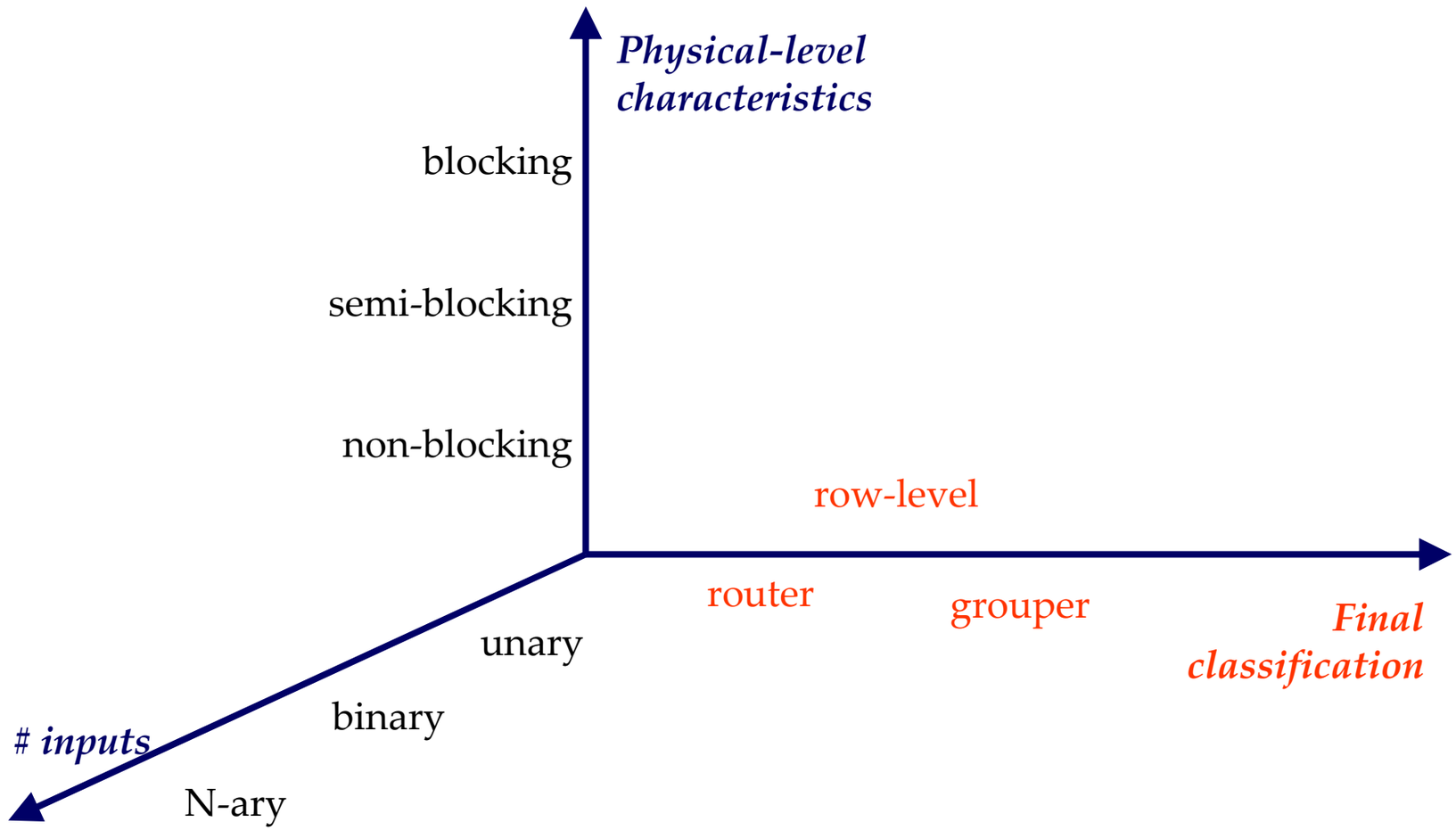
Conclusions

- We need a commonly agreed benchmark that realistically reflects real-world ETL scenarios
- We have provided
  - A list of parameters and metrics
  - A taxonomy for ETL activities (micro level)
  - A set of design patterns: *butterflies* (macro level)
- Future tasks
  - study more real-world scenarios for identifying
    - workflow complexity
    - workflow variants of different scale
    - frequencies of typically encountered ETL operations


Thank you!
All pictures are imported from MS Clipart

Auxiliary slides

Micro level


Physical-level characteristics
blocking
semi-blocking
non-blocking
row-level
router
grouper
Final classification
unary
binary
N-ary
inputs

Data Warehouse:

PART (rkey s_partkey, name, mfgr, brand, type, size, container, comment)

SUPPLIER (s_suppkey, name, address, nationkey, phone, acctbal, comment, totalcost)

PARTSUPP (s_partkey, s_suppkey, availqty, supplycost, comment)

CUSTOMER (s_custkey, name, address, nationkey, phone, acctball, mktsegment, comment)

ORDER (s_orderkey, custkey, orderstatus, totalprice, orderdate, orderpriority, clerk, shippriority, comment)

LINEITEM (s_orderkey, partkey, suppkey, linenumber, quantity, extendedprice, discount, tax, returnflag, linestatus, shipdate, commitdate, receiptdate, shipinstruct, shipmode, comment, profit)

Storage House:

PART (partkey, name, mfgr, brand, type, size, container, comment)

SUPPLIER (suppkey, name, address, nationkey, phone, acctbal, comment)

PARTSUPP (partkey, suppkey, availqty, supplycost, comment)

Sales Point:

CUSTOMER (custkey, name, address, nationkey, phone, acctball, mktsegment, comment)

ORDER (orderkey, custkey, orderstatus, totalprice, orderdate, orderpriority, clerk, shippriority, comment)

LINEITEM (orderkey, partkey, suppkey, linenumber, quantity, extendedprice, discount, tax, returnflag, linestatus, shipdate, commitdate, receiptdate, shipinstruct, shipmode, comment)

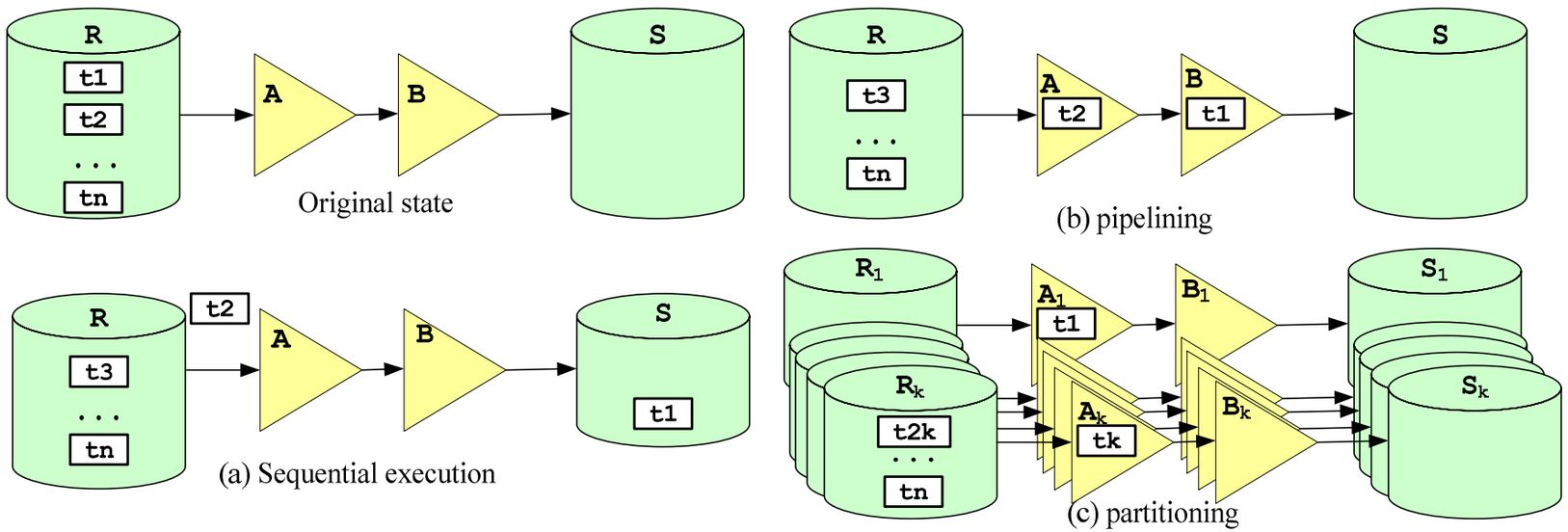
Statistics per pattern

	Filters	Functions	Routers	Aggr	Holistic f.	Joins	Diff	Unions	Load Body	Load Views
Line	1+1	2+0	0+1	0+3					INCR	INCR
Wishbone	1+0	4+0				1+0			INCR	-
Pr. Flow						3+0			I/U	-
Tree				0+1	1+0	1+0		1+0	I/U	I/U
Fork		3+0		0+4					INCR	INCR
BB(1)		4+0		0+4		1+0			INCR	FULL
BB(2)		0+2					1		-	I/U
	2+1	13+2	0+1	0+12	1+0	6+0	1	1+0		

Legend:

- N+M (left wing + right wing)
- INCR: incremental maintenance
- I/U: insert and/or update
- FULL: full recomputation

Partitioning & parallelism



Measures

- Q1. Measures for data freshness and data consistency
  - The objective is to have data respect both database and business rules
  - Concrete measures are:
    - (M1.1) Percentage of data that violate business rules
    - (M1.2) Percentage of data that should be present at their appropriate warehouse targets, but they are not

Measures

- Q2. Measures for the resilience to failures
  - Test the capability of a workflow to successfully compensate within the specified time constraints
  - Concrete measures are:
    - (M2.1) Percentage of successfully resumed workflow executions
    - (M2.2) MTBF, the mean time between failures
    - (M2.3) MTTR, mean time to repair
    - (M2.4) Number of recovery points used
    - (M2.5) Resumption type: synchronous or asynchronous
    - (M2.6) Number of replicated processes (for replication)
    - (M2.7) Uptime of ETL process

Measures

- Q3. Measures for maintainability (qualitative objective)
  - It captures the effort needed after a change has been occurred either at the SLA's or the underlying systems
  - Concrete measures are:
    - (M3.1) Length of the workflow (i.e., the length of its longest path)
    - (M3.2) Complexity of the workflow refers to the amount of relationships that combine its components
    - (M3.3) Modularity (or cohesion) refers to the extent to which the workflow components perform exactly one job
    - (M3.4) Coupling captures the amount of relationship among different recordsets or activities (i.e., workflow components)

Measures

- Q4. Measures for the speed of the overall process
  - The objective is to perform the ETL process as fast as possible
  - Concrete measures are:
    - (M4.1) Throughput of regular workflow execution (this may also be measured as total completion time)
    - (M4.2) Throughput of workflow execution including a specific percentage of failures and their resumption
    - (M4.3) Average latency per tuple in regular execution

Measures

- Q5. Measures for partitioning parallelism
  - (M5.1) Partition type (e.g., round-robin, hash-based, follow-database-partitioning, and so on)
  - (M5.2) Number and length of workflow parts that use partitioning
  - (M5.3) Number of partitions
  - (M5.4) Data volume in each partition (it is related to partition type too)
- Q6. Measures for pipelining parallelization
  - (M6.1) CPU and memory utilization for pipelining flows or for individual operation run in such flows
  - (M6.2) Min/Max/Avg length of the largest and smaller paths (or subgraphs) containing pipelining operations
  - (M6.3) Min/Max/Avg number of blocking operations

Measures

- Q7. Measured Overheads
  - The overheads at the source and DW are measured in terms of consumed memory and latency w.r.t. regular operation
  - Concrete measures are:
    - (M7.1) Min/Max/Avg timeline of memory consumed at the sources
    - (M7.2) Time needed to complete a set of OLTP transactions in the presence (vs. absence) of ETL software at the sources (normal mode)
    - (M7.3) The same as 7.2, but with source failures (recovery mode)
    - (M7.4) Min/Max/Avg/ timeline of memory consumed at the DW
    - (M7.5) (active warehousing) Time needed to complete a set of OLAP queries in the presence (vs. absence) of ETL software at the DW (normal mode)
    - (M7.6) The same as M7.5, but with DW failures (recovery mode)