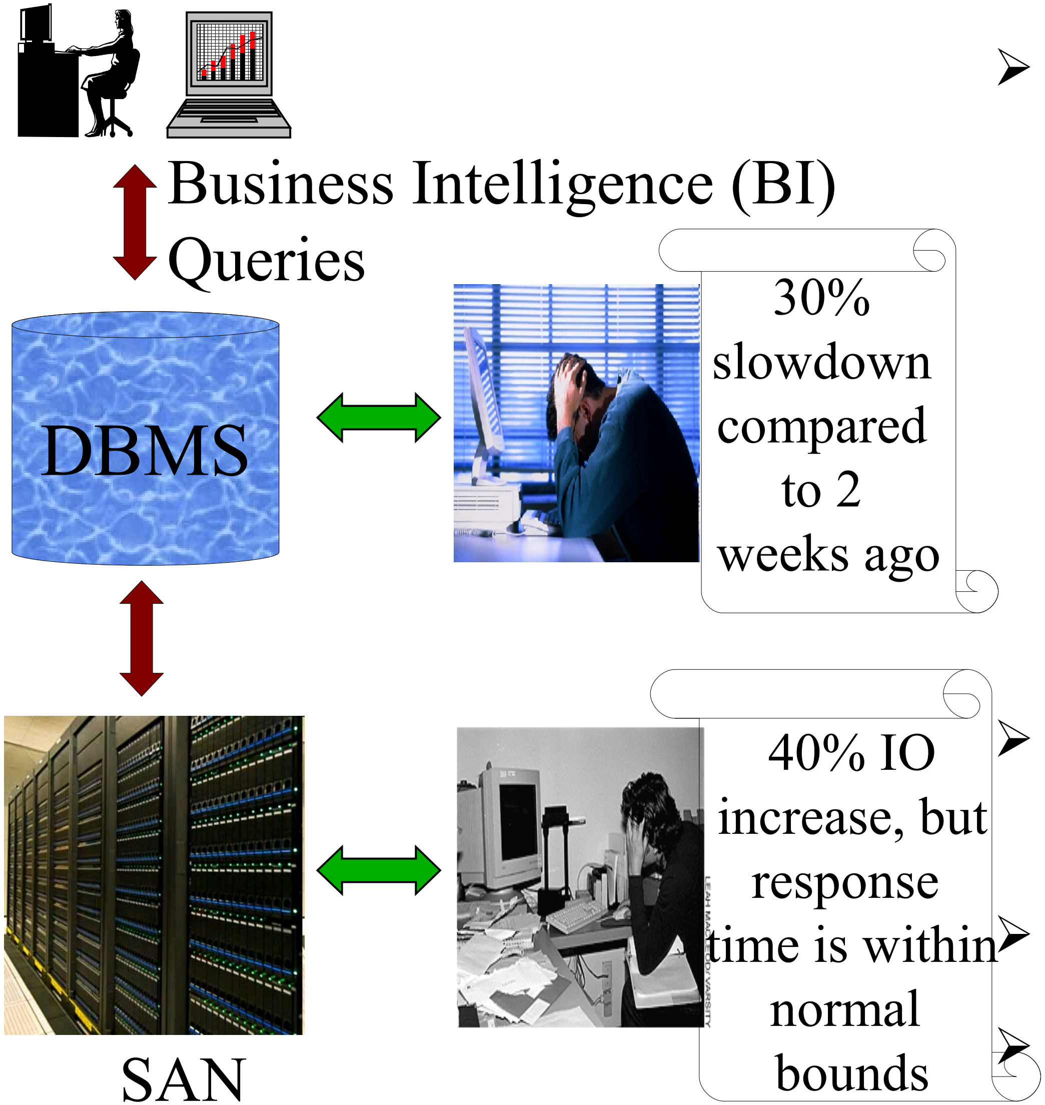
DIADS: Addressing the "My-Problem-or-Yours" Syndrome with Integrated SAN and Database Diagnosis

Nedyalko Borisov

Duke University

Shivnath Babu,	Duke
Sandeep Uttamchandani,	IBM
Ramani Routray,	IBM
Aameek Singh,	IBM

Current State

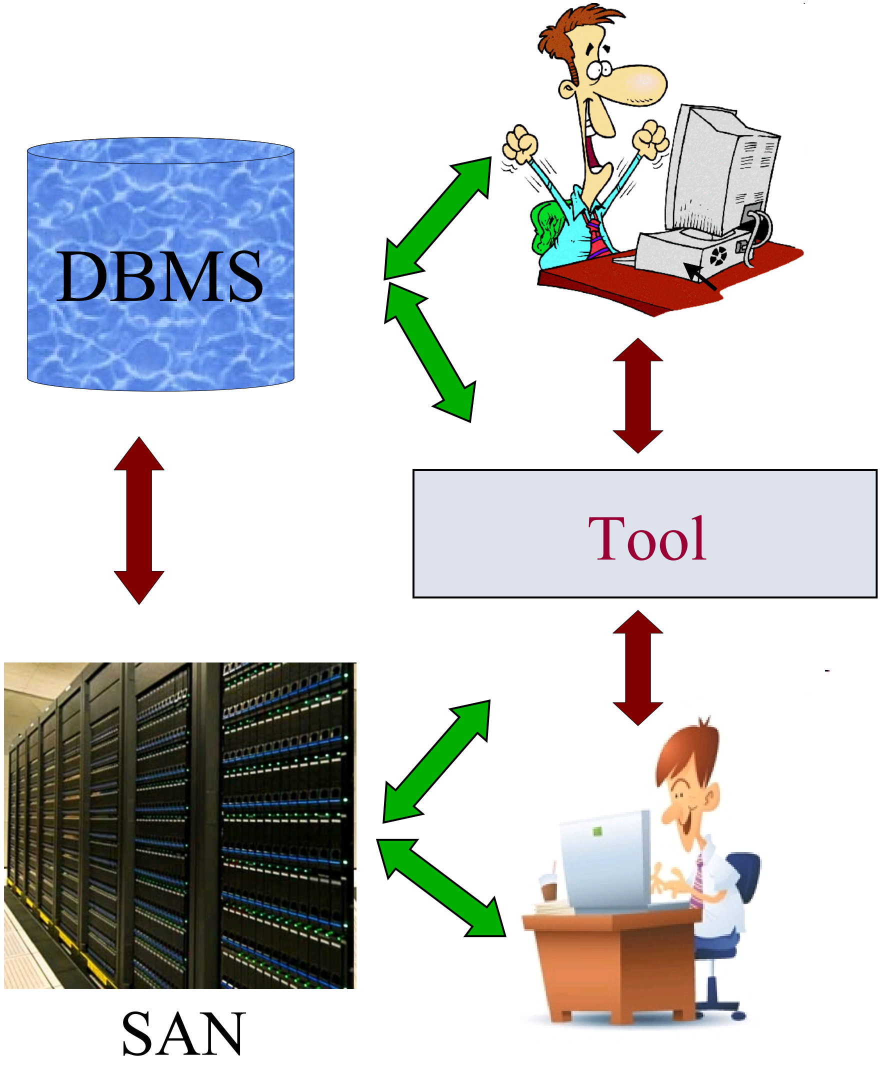


30% slowdown compared to 2 weeks ago

40% IO increase, but response time is within normal bounds

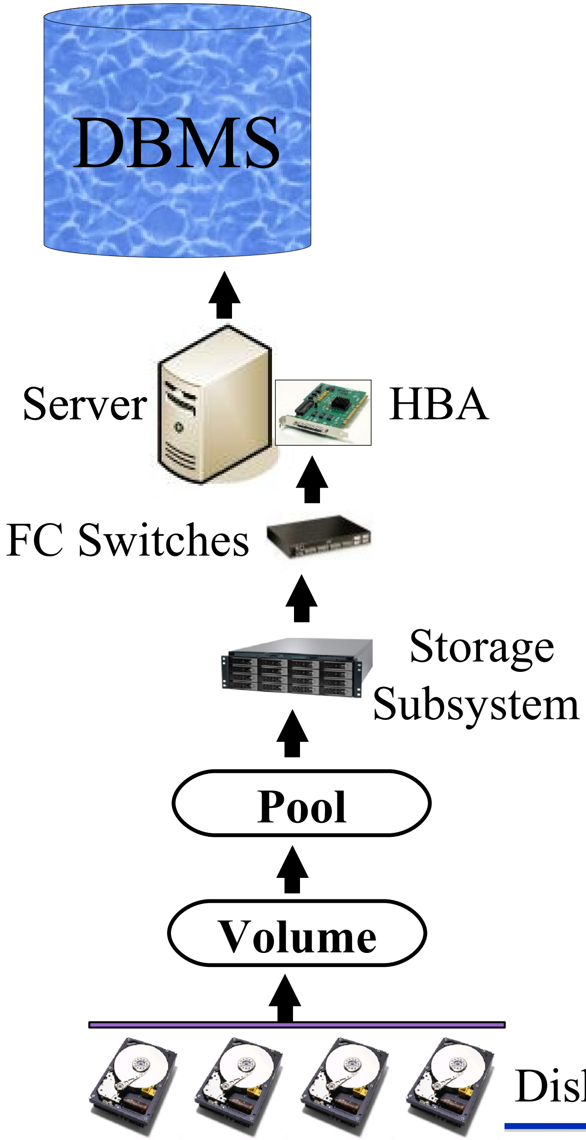
- Databases (DBMSs) and SANs have separate admin teams
  - Each team has limited visibility into full system
- Database admin (DBA) opens problem ticket
- SAN admin responds
- To and fro may continue

What is the Natural Solution?



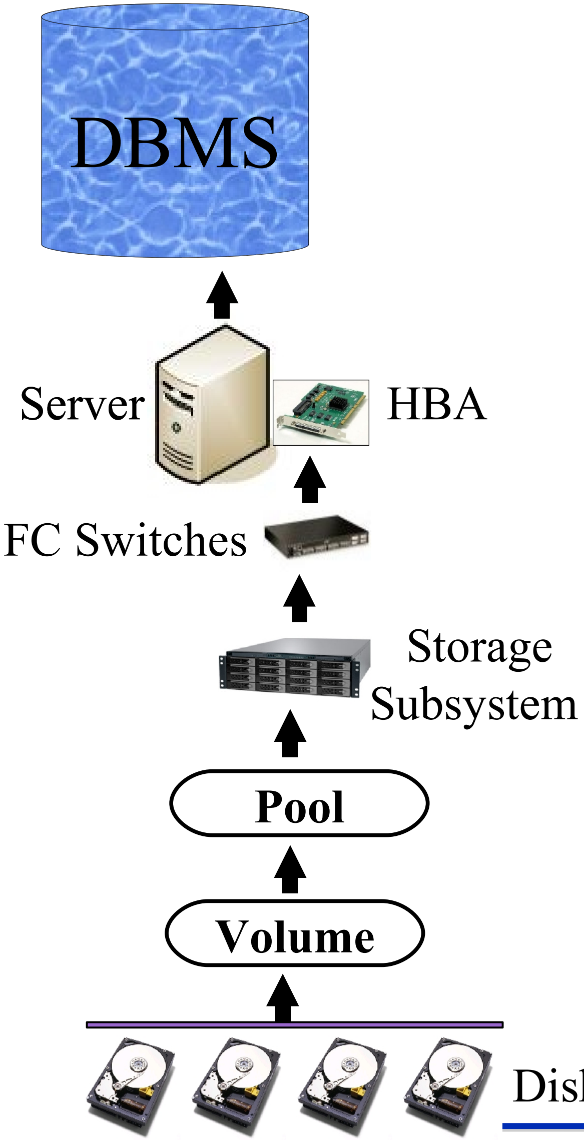
- Separate admin teams do not have holistic view of query execution
- Easy if we have low-level tracing
  - May be infeasible
  - May have high overhead

Our Solution: DIADS



- DBMS level and SAN level monitoring tools - e.g., Hyperic HQ, TPC
- Need to integrate these separate pieces of data to create a holistic view of query execution
- DIADS: DIAgnosis for Databases and SANs
  - Inputs
    - Poorly performing query
    - Monitoring data from DBMS
    - Monitoring data from SAN

Our Solution: DIADS



- DBMS level and SAN level monitoring tools - e.g., Hyperic HQ, TPC
- Need to integrate these separate pieces of data to create a holistic view of query execution
- DIADS: DIAgnosis for Databases and SANs
  - Outputs
    - Root cause of query's poor performance (ideal)
    - Localization of problem

Contributions of DIADS

Feature	Novelty
- Annotated Plan Graph (APG) across DBMS and SAN	- Holistic view of query execution - Generated from commonly-available monitoring data
- Diagnosis workflow	- Careful combination of machine-learning (ML) techniques and expert knowledge (EK) - Deals with flood of monitoring data (ML) - Deals with noisy monitoring data in real systems (ML + EK) - Deals with fault propagation (EK) - Incorporates checks and balances

Roadmap

- Motivation
- Running Example
- Workflow
- Evaluation
- Conclusions & Future work

Running Example

- Report-generation query (TPC-H Query 2) is running periodically

```
SELECT s_acctbal, s_name, n_name, p_partkey, p_mfgr,
s_address, s_phone, s_comment
FROM part,  supplier,  partsupp,  nation,  region
WHERE  p_partkey = ps_partkey
        AND s_suppkey = ps_suppkey AND p_size = 28
        AND p_type like '%COPPER' AND s_nationkey = n_nationkey
        AND n_regionkey = r_regionkey   AND r_name = 'AMERICA'
        AND ps_supplycost = (
                SELECT  min(ps_supplycost)
                FROM     partsupp, supplier, nation, region
                WHERE p_partkey = ps_partkey
                        AND s_suppkey = ps_suppkey
                        AND s_nationkey = n_nationkey
                        AND n_regionkey = r_regionkey
                        AND r_name = 'AMERICA' )
ORDER BY s_acctbal desc, n_name, s_name, p_partkey;
```

SAN (Mis)configuration Issue

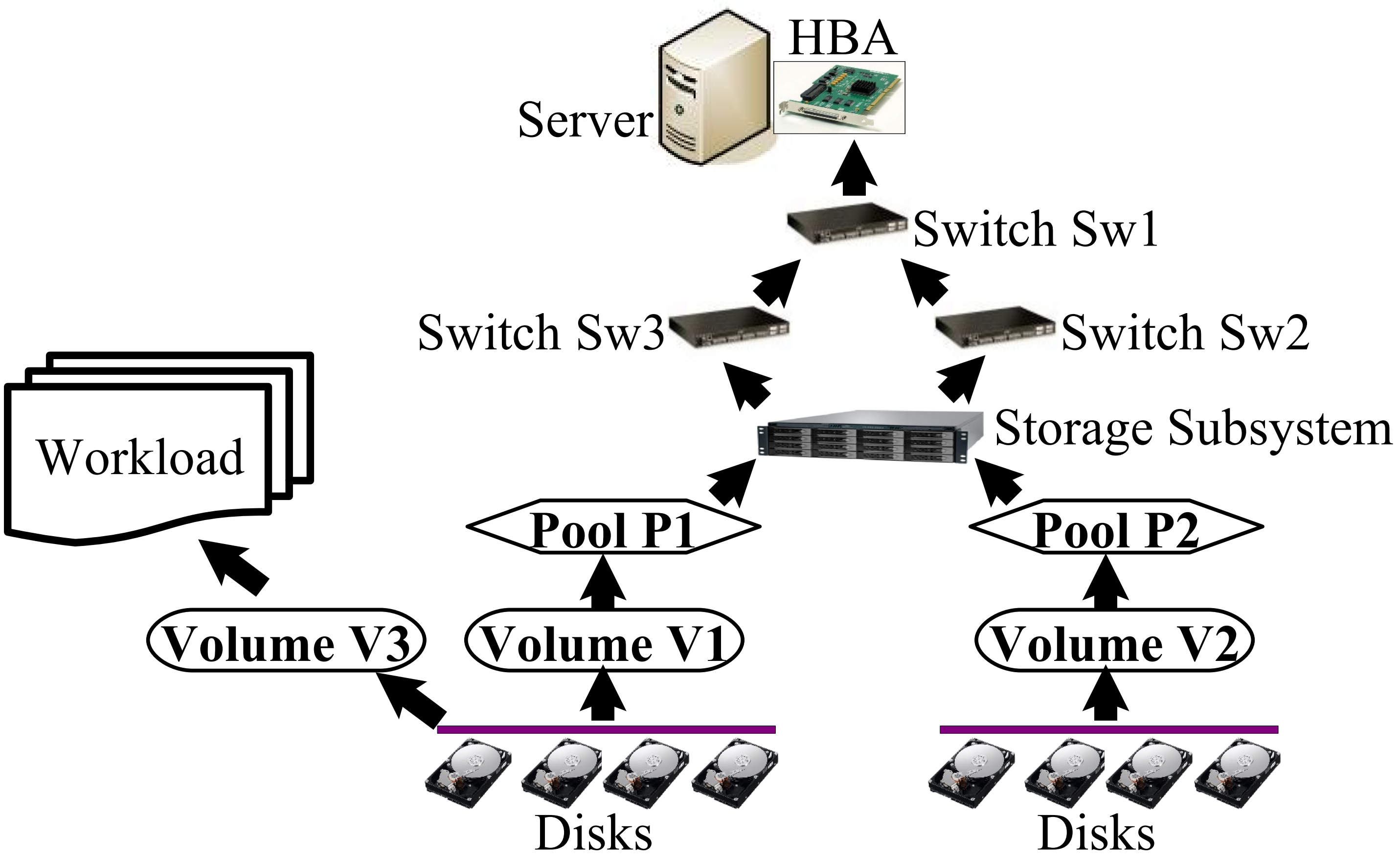


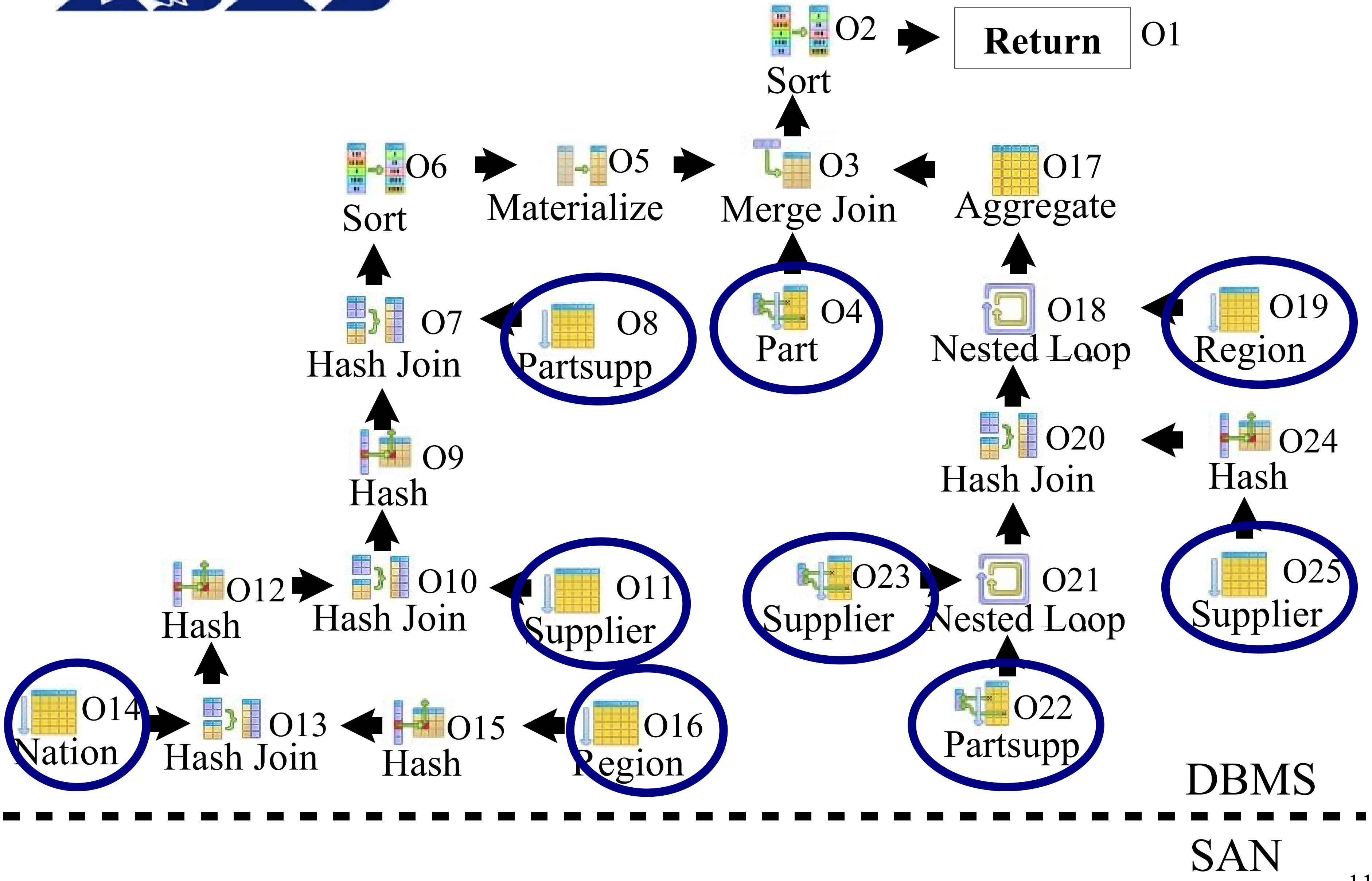
Table: Partsupp

Tables: Region, Nation, Supplier, Part

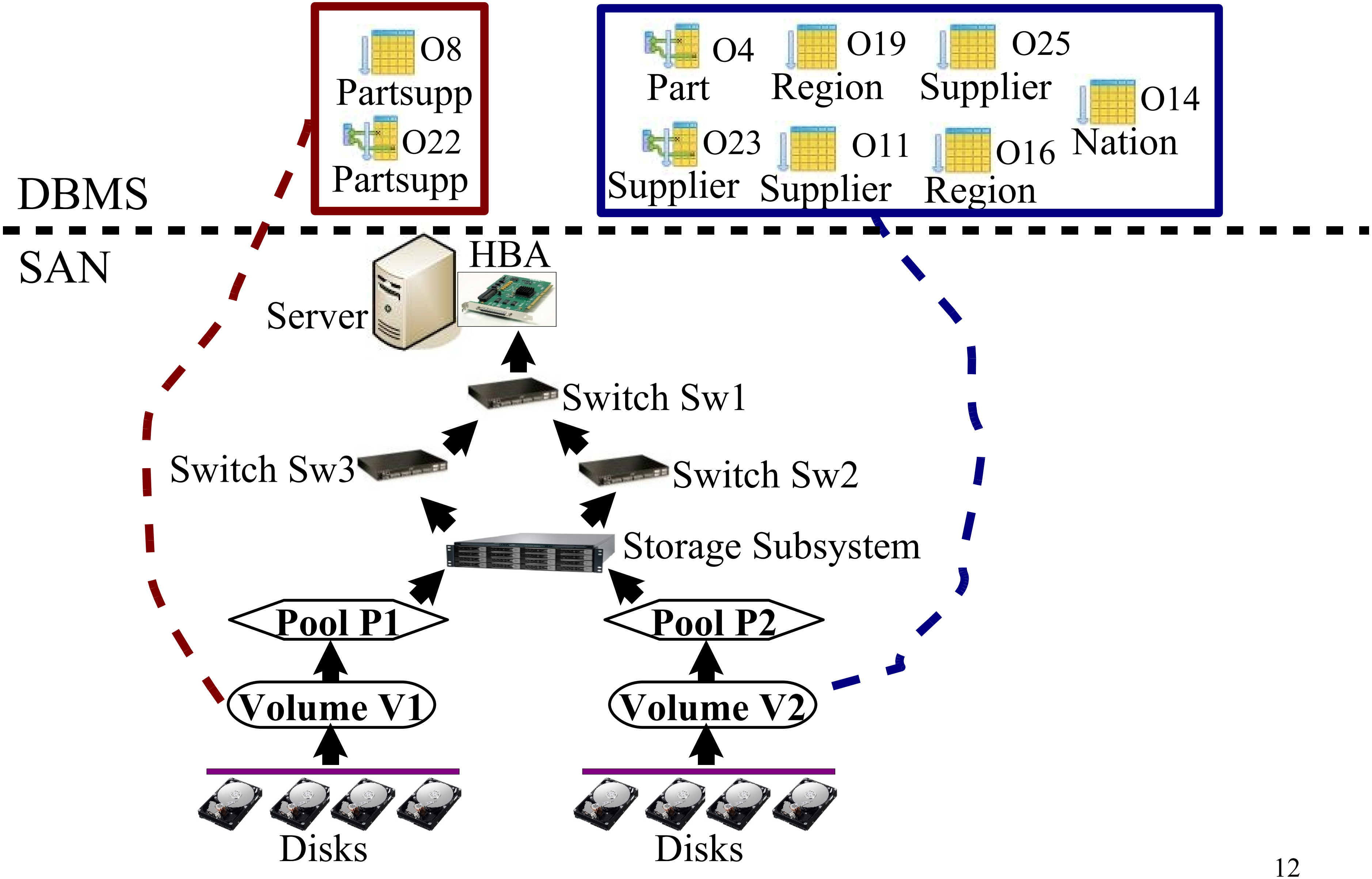
Running Example (Cont.)

- Observations
  - 15.2 minutes
  - 15.1 minutes
  - 14.9 minutes
  - 15.2 minutes
  - 33.1 minutes
  - 31.3 minutes
- Diagnose the cause for the slowdown

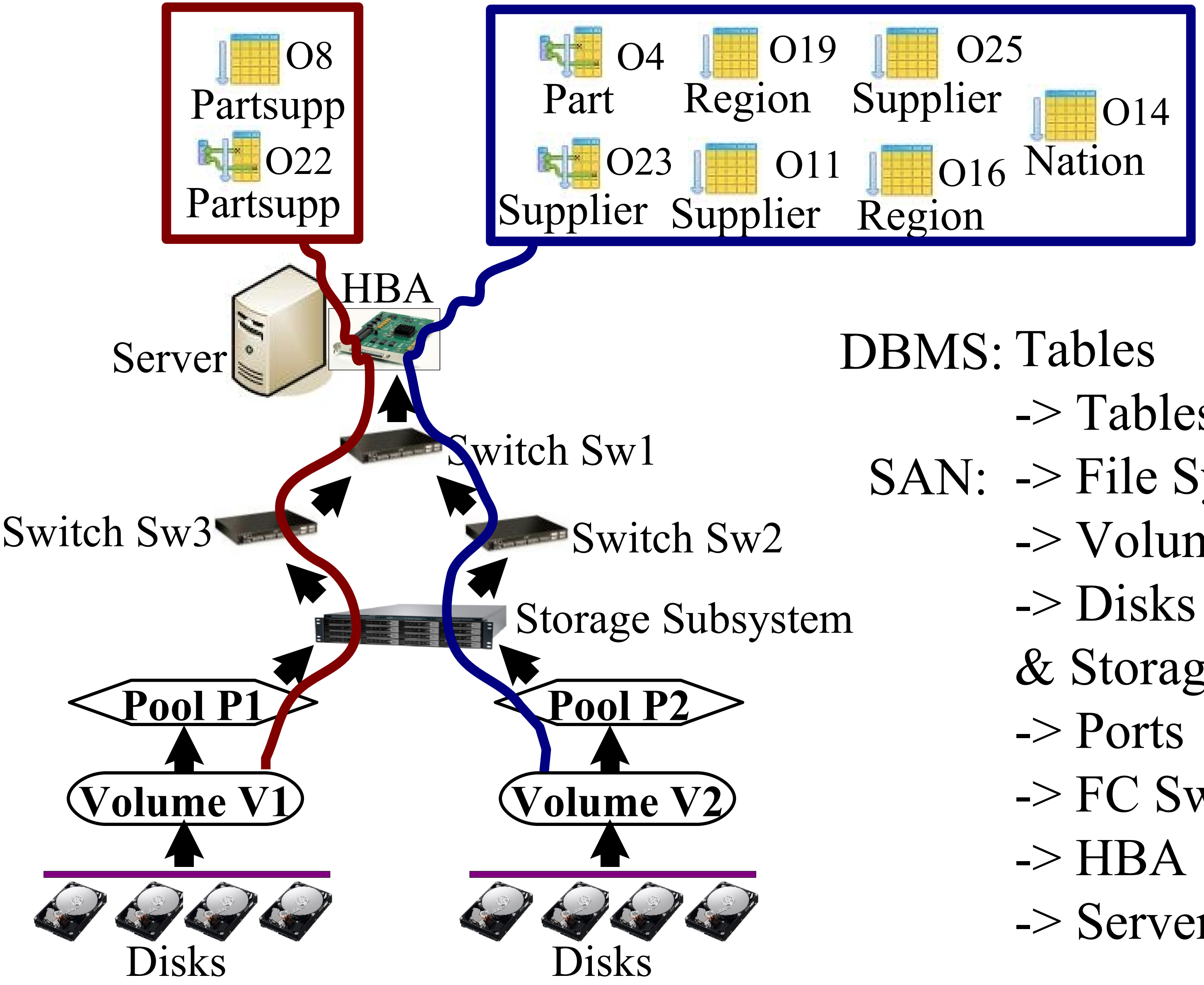
Query Plan Execution



Running Example of APG

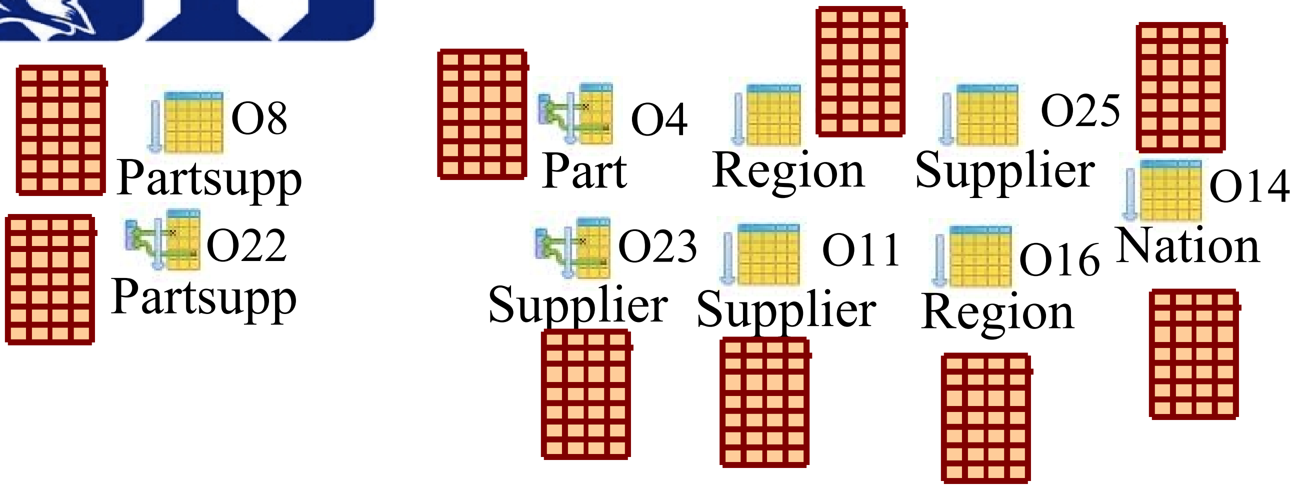


APG Dependency Paths



DBMS: Tables
-> Tablespaces
SAN: -> File System
-> Volumes
-> Disks & Pools
& Storage Subsystem
-> Ports
-> FC Switches
-> HBA
-> Server

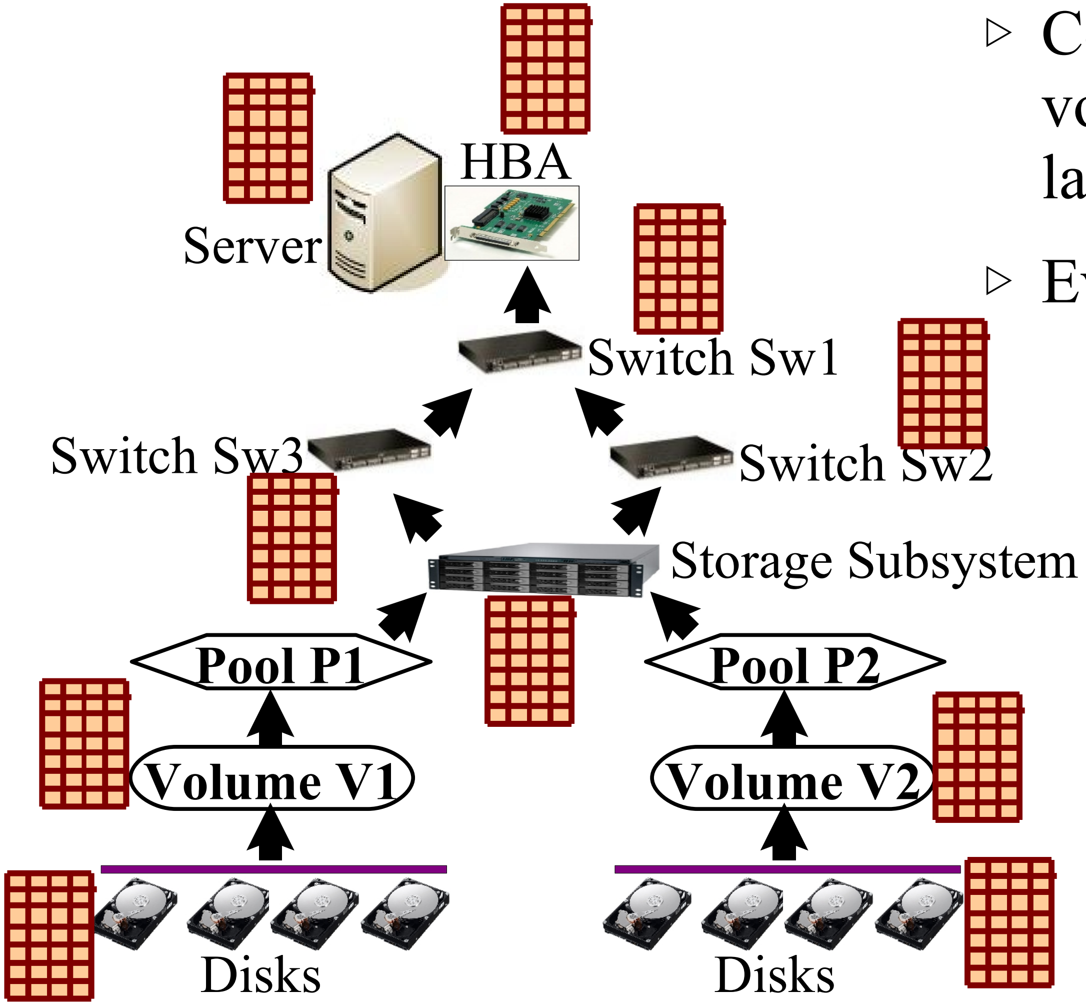
APG Annotations



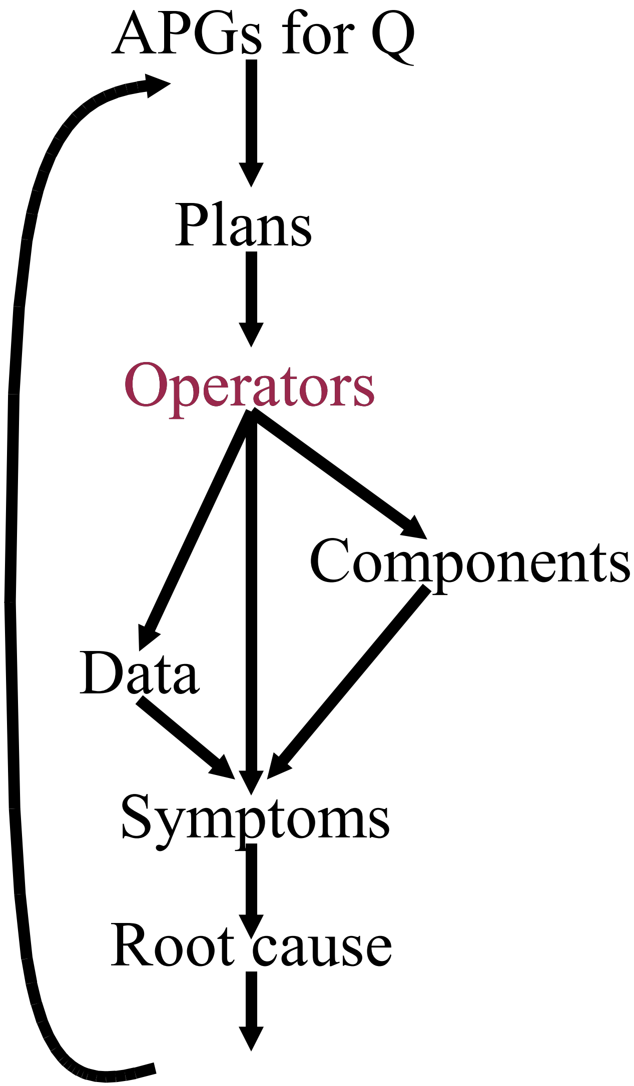
- Monitoring data
  - DBMS
    - Plan-level data (e.g., running time of operator, # of records)
    - DBMS-level data (e.g., hits in the buffer pool, event logs)

APG Annotations

- Monitoring data
  - SAN
    - Component-level data (e.g., for volumes - #reads, #writes, latency, bytes transfered)
    - Event logs



Workflow



APGs for Q

Plans

Operators

Components

Data

Symptoms

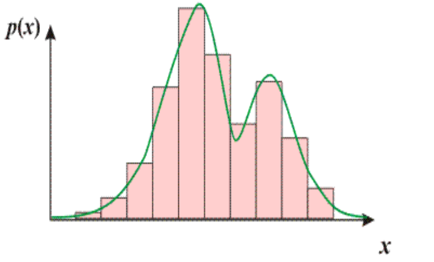
Root cause

Admin identifies run instances when query Q ran fine and when it did not

Correlate with change in operator costs

Module Correlated Operators

- Which operators have a change in running time that explains change in running time of the entire plan?
- Anomaly Score computed with Kernel Density Estimation (KDE)



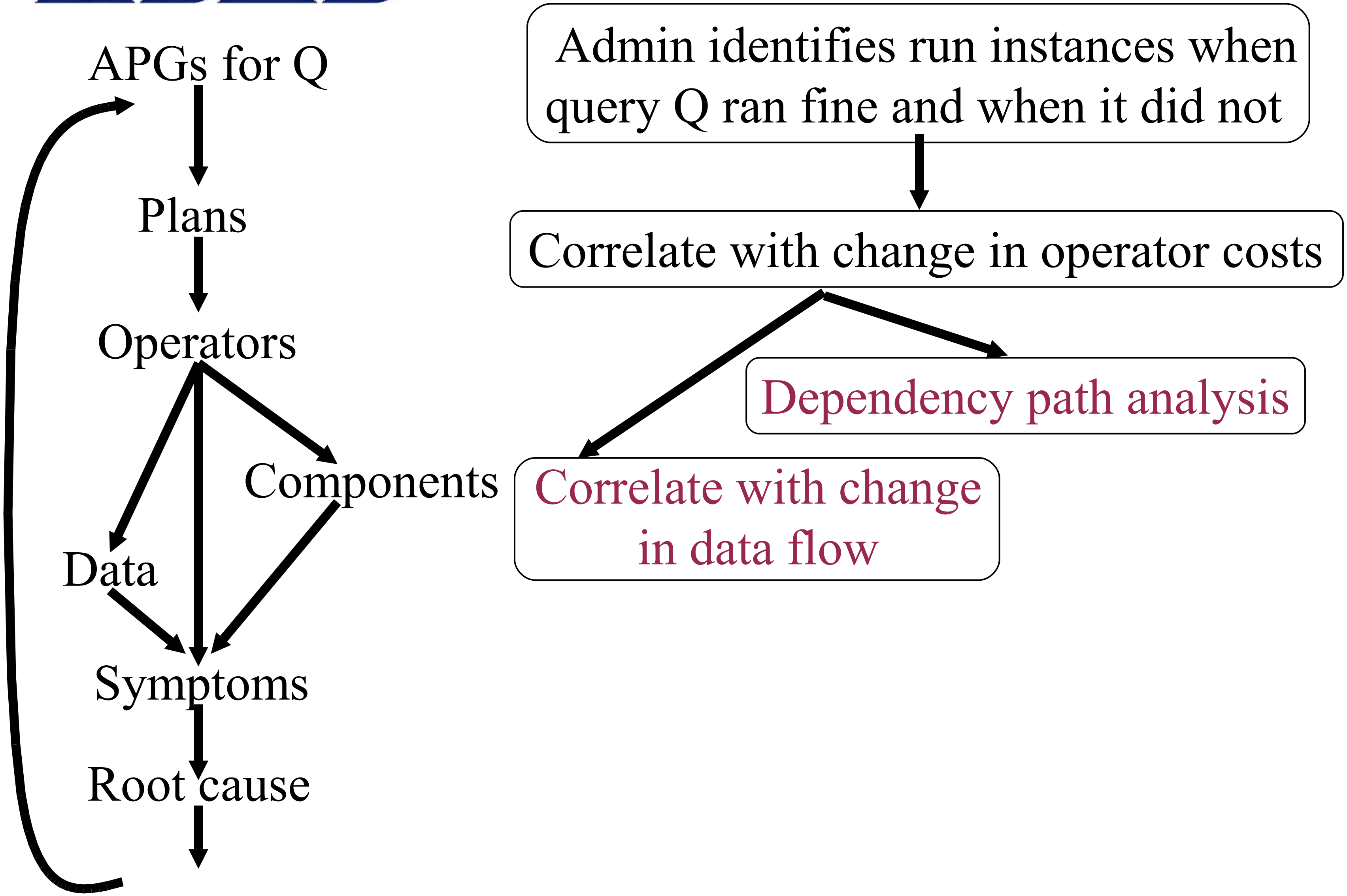
	Anomaly Score
O8	1.0
O4	0.965
O22	1.0

Running times (seconds)

	O16	O14	O11	O8	O4	O25	O23	O22	O19	**Plan**
APG #1	1	2	43	377	277	1	44	24	1	**911**
APG #2	1	1	44	382	281	1	39	22	2	**920**
APG #3	2	2	43	380	272	1	38	26	1	**905**
APG #4	2	1	43	628	401	1	51	45	1	**1903**
APG #5	1	1	45	596	390	1	40	51	2	**1880**

KDE picture borrowed from Internet

Workflow



APGs for Q

Plans

Operators

Components

Data

Symptoms

Root cause

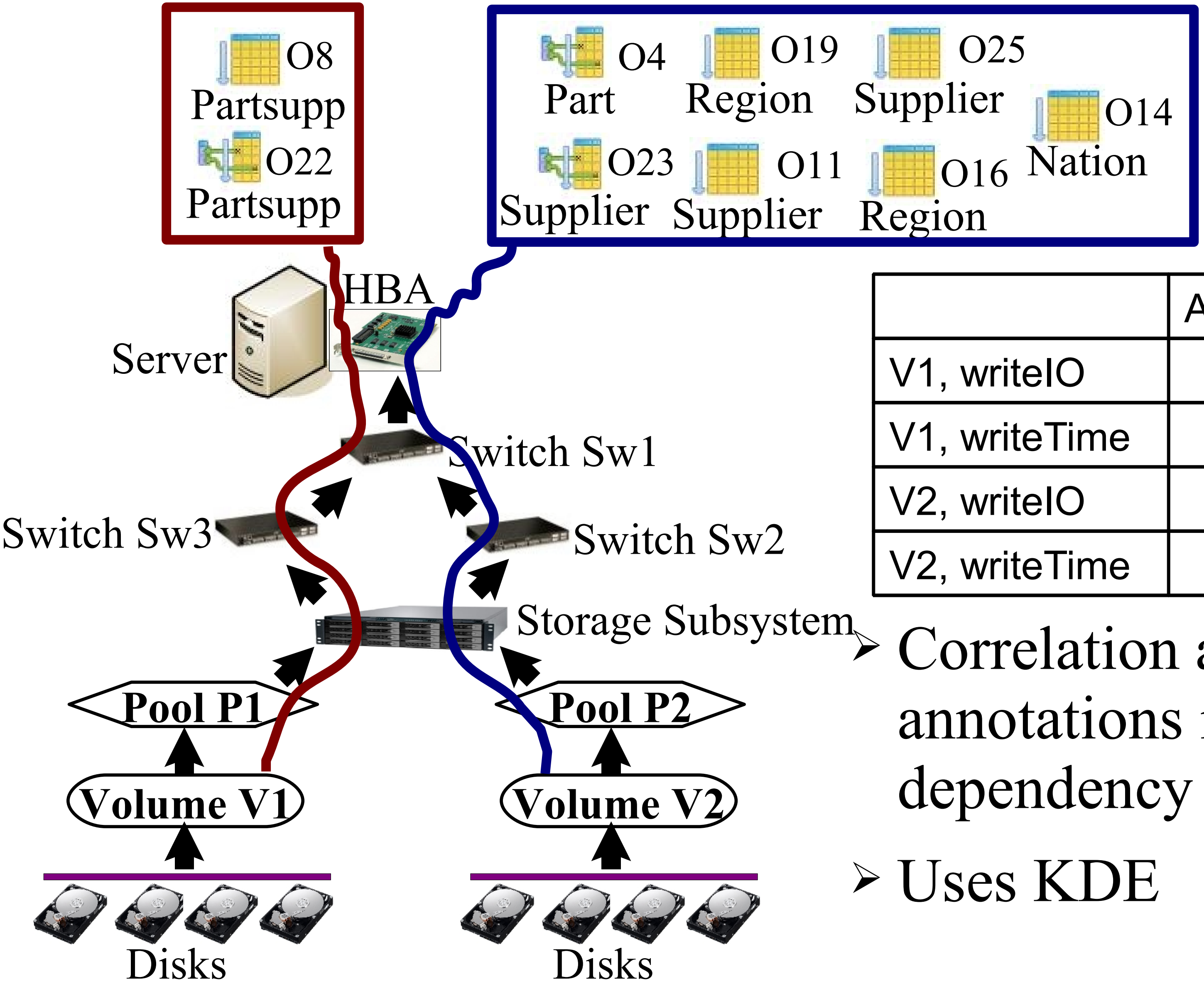
Admin identifies run instances when query Q ran fine and when it did not

Correlate with change in operator costs

Dependency path analysis

Correlate with change in data flow

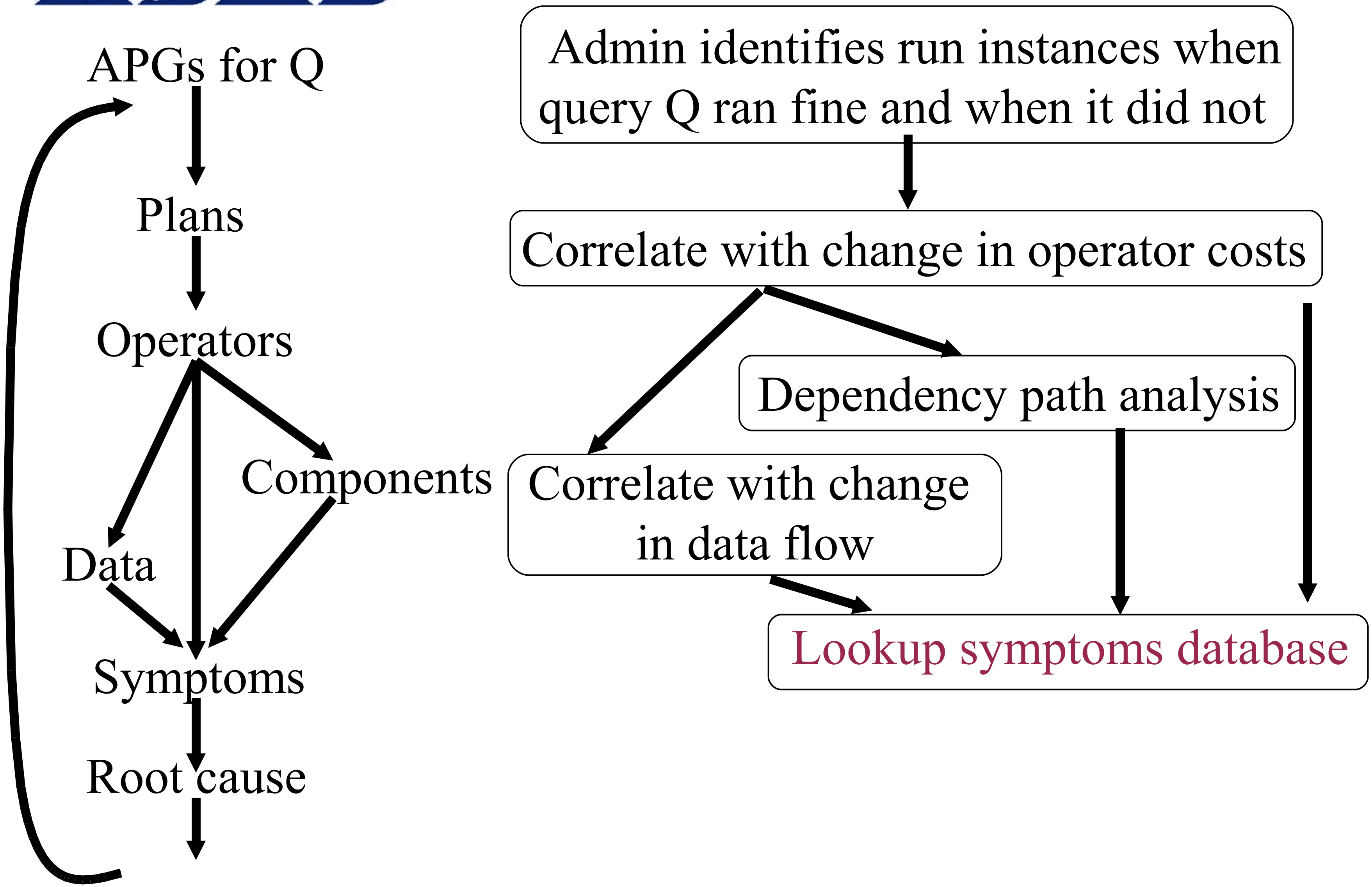
Module Dependency Analysis



	Anomaly Score
V1, writeIO	0.894
V1, writeTime	0.823
V2, writeIO	0.063
V2, writeTime	0.479

- Correlation analysis of annotations in each dependency path
- Uses KDE

Workflow



APGs for Q

Plans

Operators

Components

Data

Symptoms

Root cause

Admin identifies run instances when query Q ran fine and when it did not

Correlate with change in operator costs

Dependency path analysis

Correlate with change in data flow

Lookup symptoms database

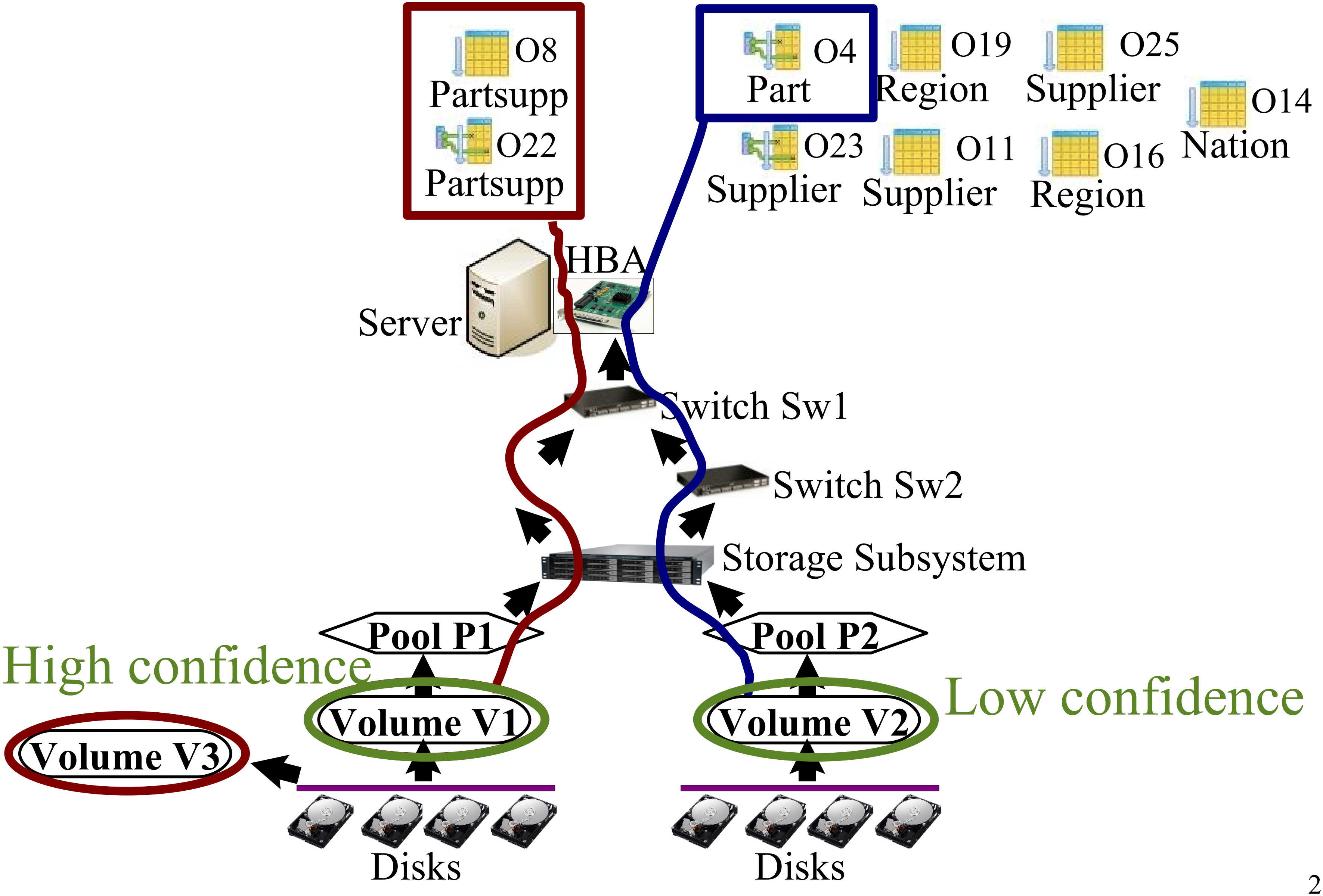
Module Symptom Database

- Mapping from symptoms to root causes
  - Handling event (fault) propagation
- Machine learning is not enough. Need to incorporate expert knowledge about DBMS and SAN systems
- Many implementation choices
  - Codebook (ex: EMC)
  - Rules (ex: Oracle)
  - Bayesian networks

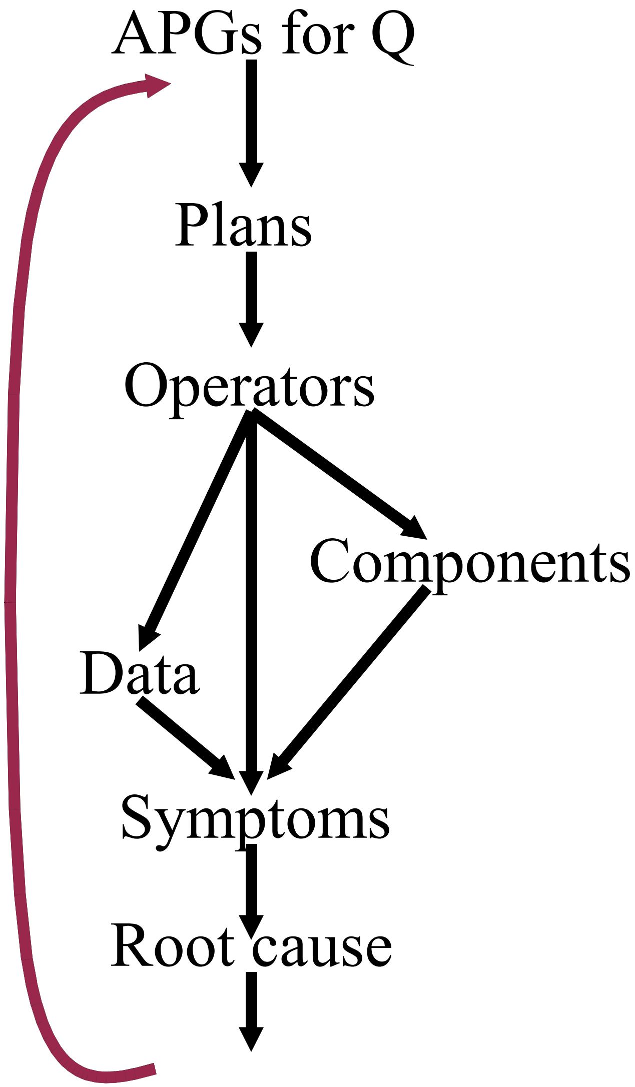
Our Impl. of Symptom Database

Challenges	Our Solution
➢ How are symptoms expressed?	➢ Language for expressing complex symptoms ‣ Intuitive built-in patterns ‣ Temporal patterns
➢ How is database populated and maintained?	➢ Currently, by administrators; Working on partial automation
➢ How to prevent database bloat?	➢ Parameterized symptoms and root causes
➢ What about missing/extra symptoms due to noise?	➢ Support for partial matching with confidence score

Module Symptom Database



Workflow



APGs for Q

Plans

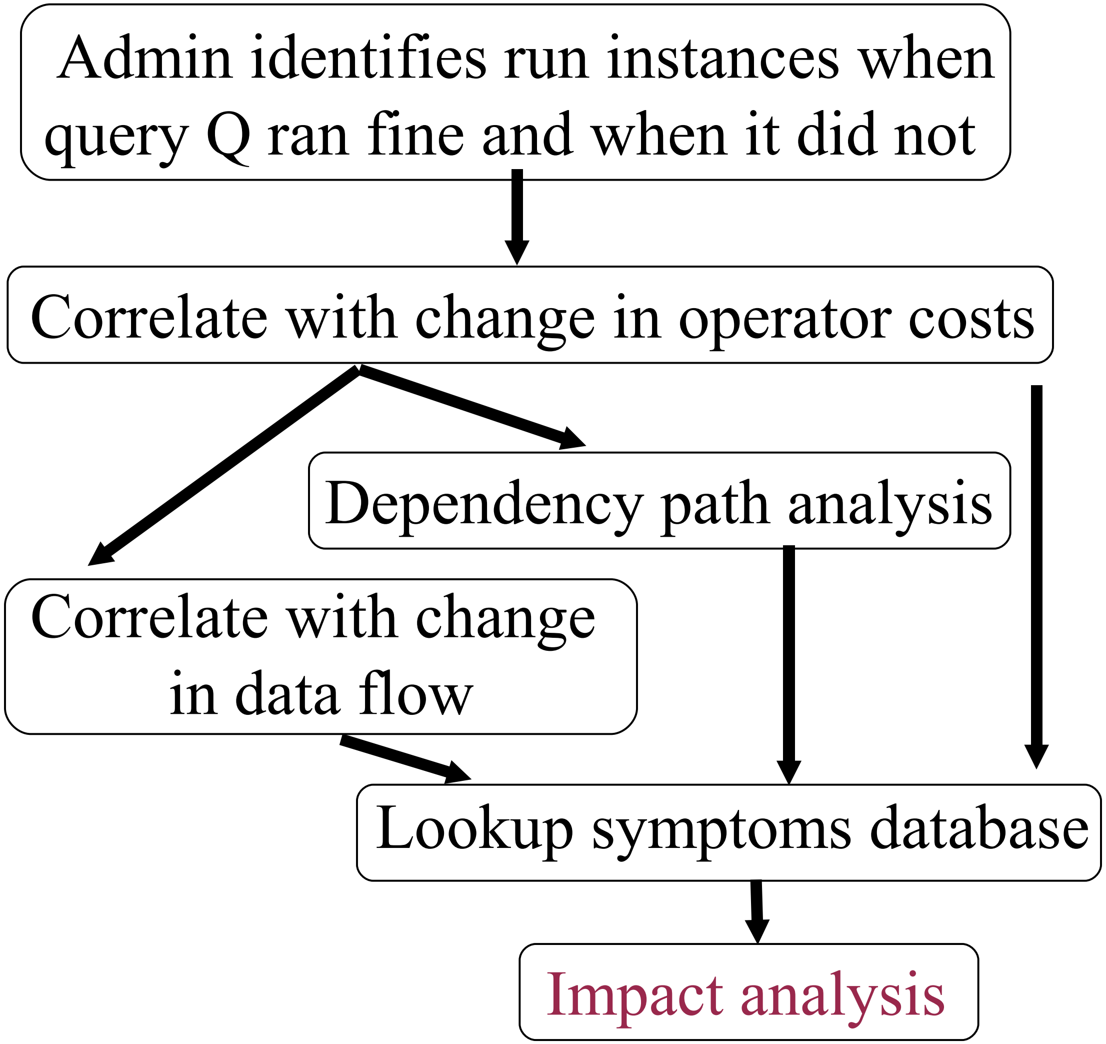
Operators

Components

Data

Symptoms

Root cause



Admin identifies run instances when query Q ran fine and when it did not

Correlate with change in operator costs

Dependency path analysis

Correlate with change in data flow

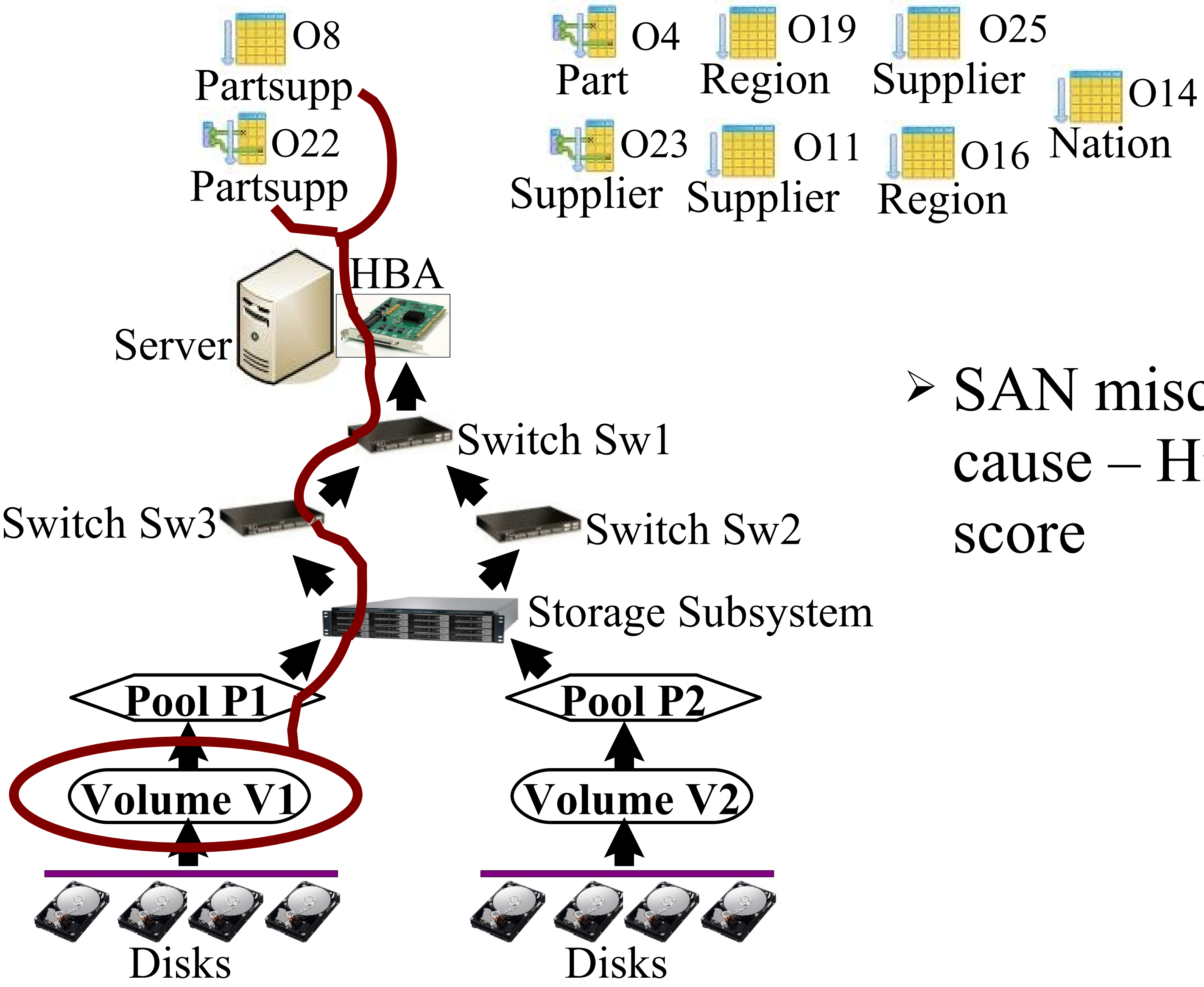
Lookup symptoms database

Impact analysis

Module Impact Analysis

- What fraction of the slowdown does this root cause explain?
  - Impact score (0-100%)
- Uses
  - Separating high-impact causes from others
  - Safeguard against false positives
  - Identifying presence of false negatives
- Suite of techniques to compute impact score
  - Reverse dependency analysis: Bottom-up traversal of the correlated dependency paths
  - Use of models (DBMS cost models, SAN device models)

Reverse Dependency Analysis

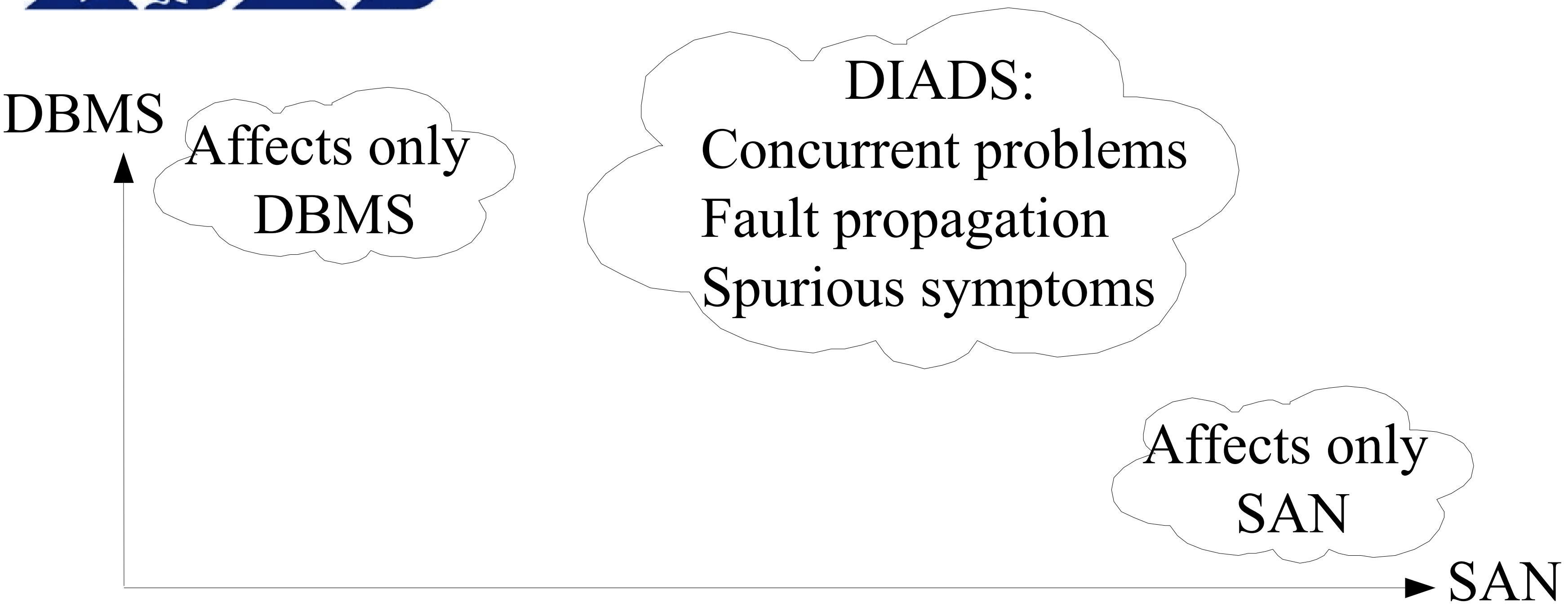


- SAN misconfiguration cause – High Impact score

Roadmap

- Motivation
- Running Example
- Workflow
- Evaluation
- Conclusions & Future work

Evaluation Methodology



DBMS

Affects only DBMS

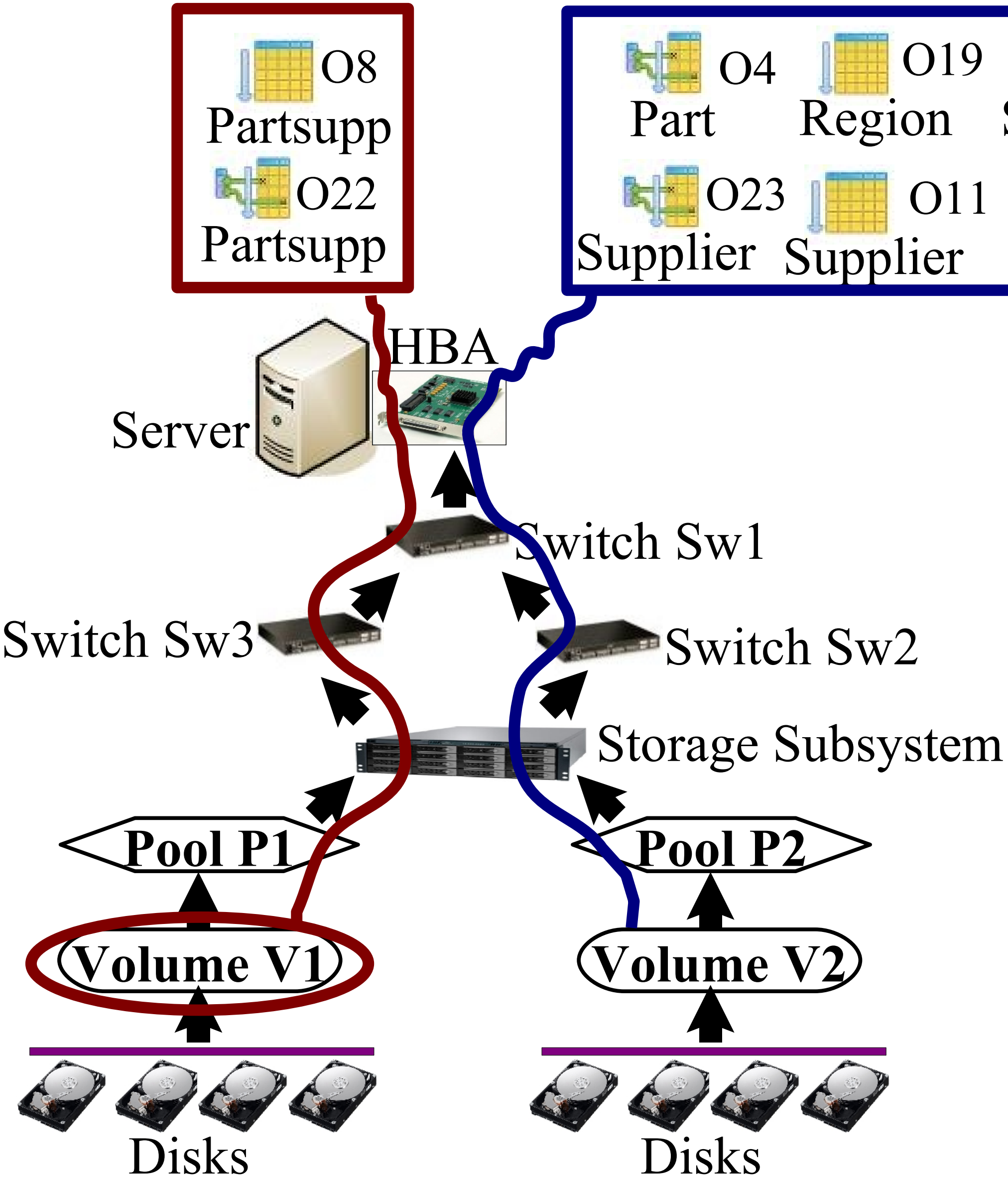
DIADS:
Concurrent problems
Fault propagation
Spurious symptoms

Affects only SAN

SAN

- Testbed
  - TPC-H Queries
  - PostgreSQL
  - IBM DS6000 storage manager
  - On production system

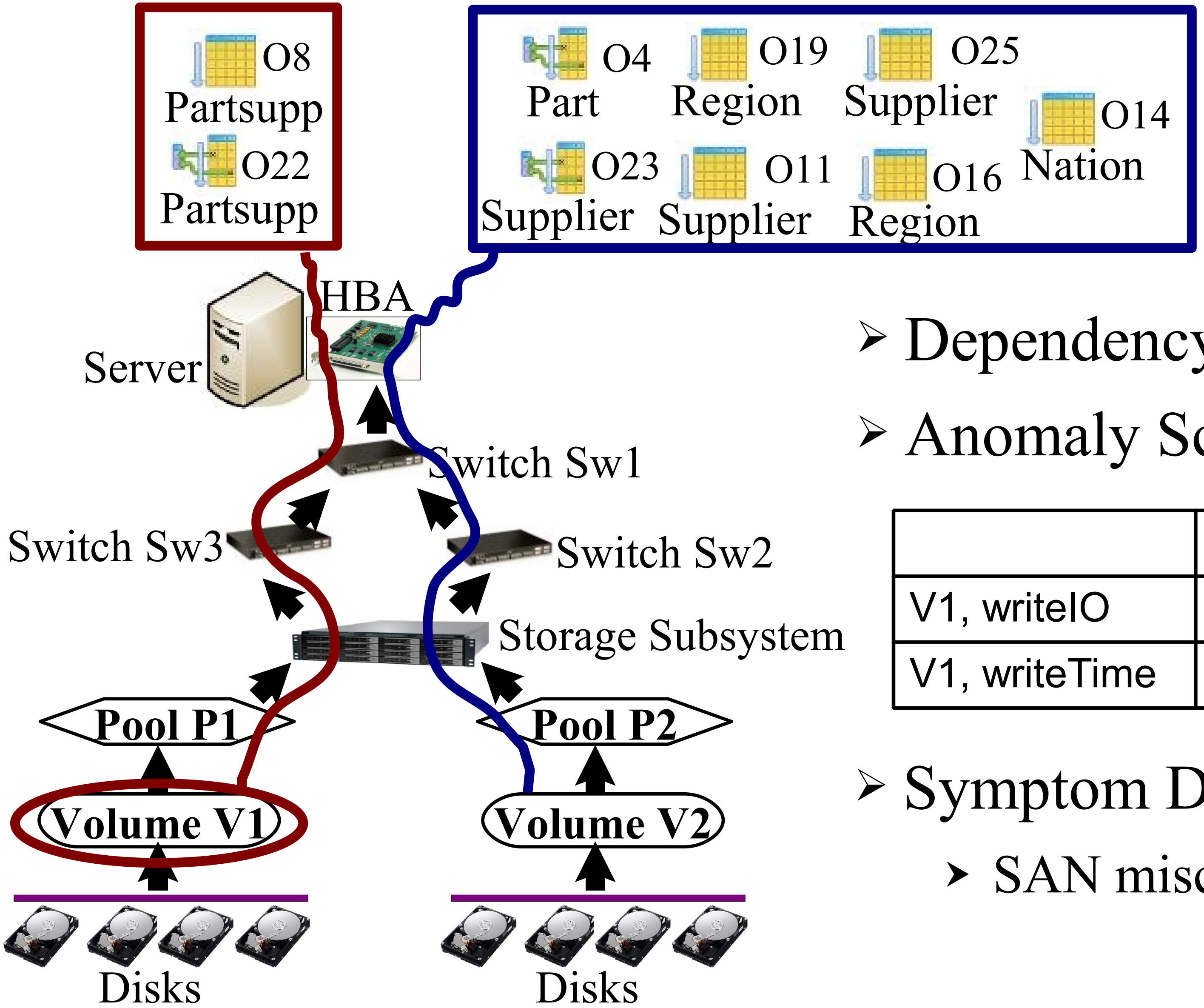
Recap of Running Example (Scenario 1)



- Problem
  - SAN misconfiguration
- Correlated Operators
  - O4, O8, O22
- Anomaly Scores

	Anomaly Score
O8	1.0
O4	0.965
O22	1.0

Recap of Running Example (Scenario 1)

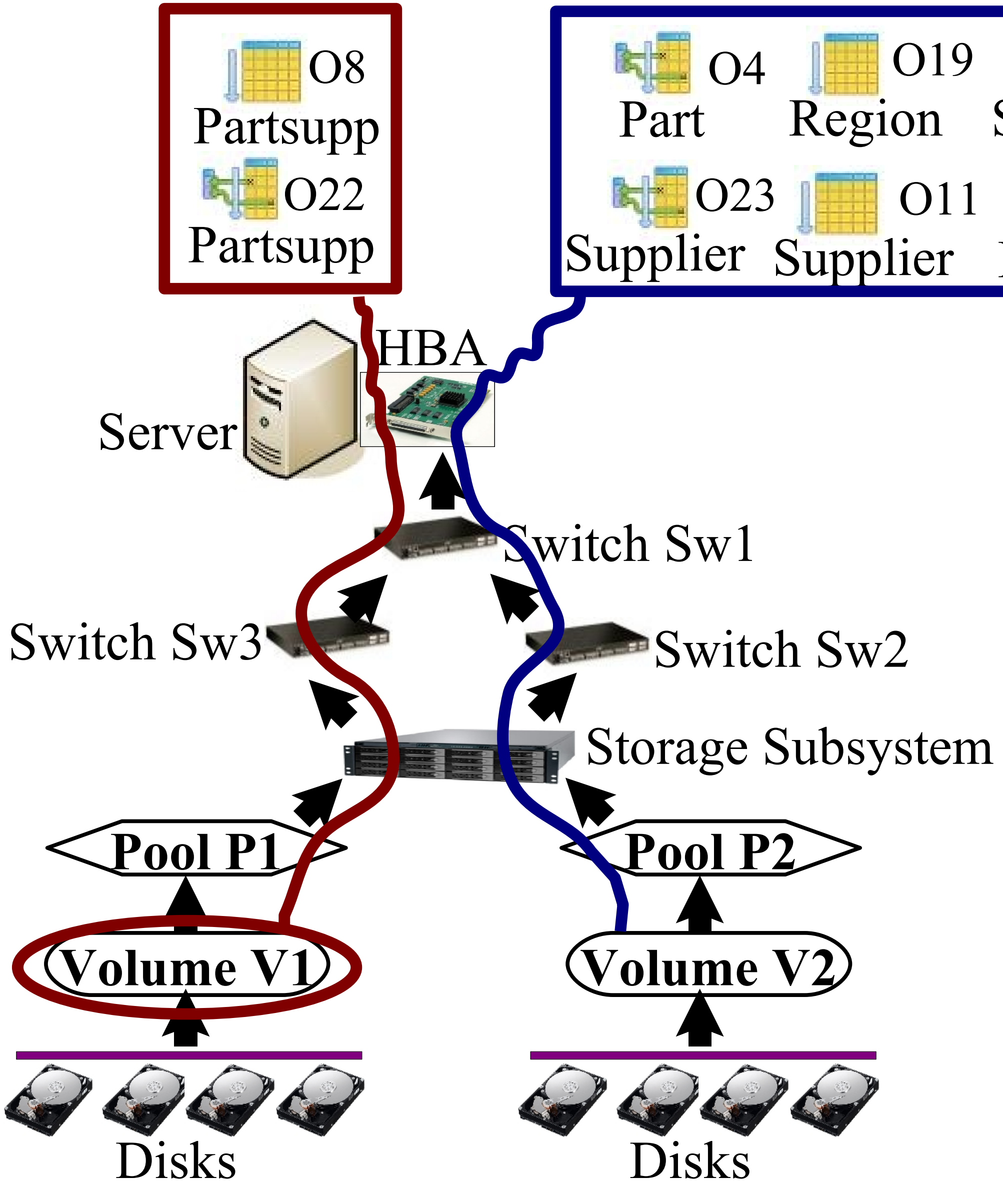


- Dependency Analysis
- Anomaly Scores

	Anomaly Score
V1, writeIO	0.894
V1, writeTime	0.823

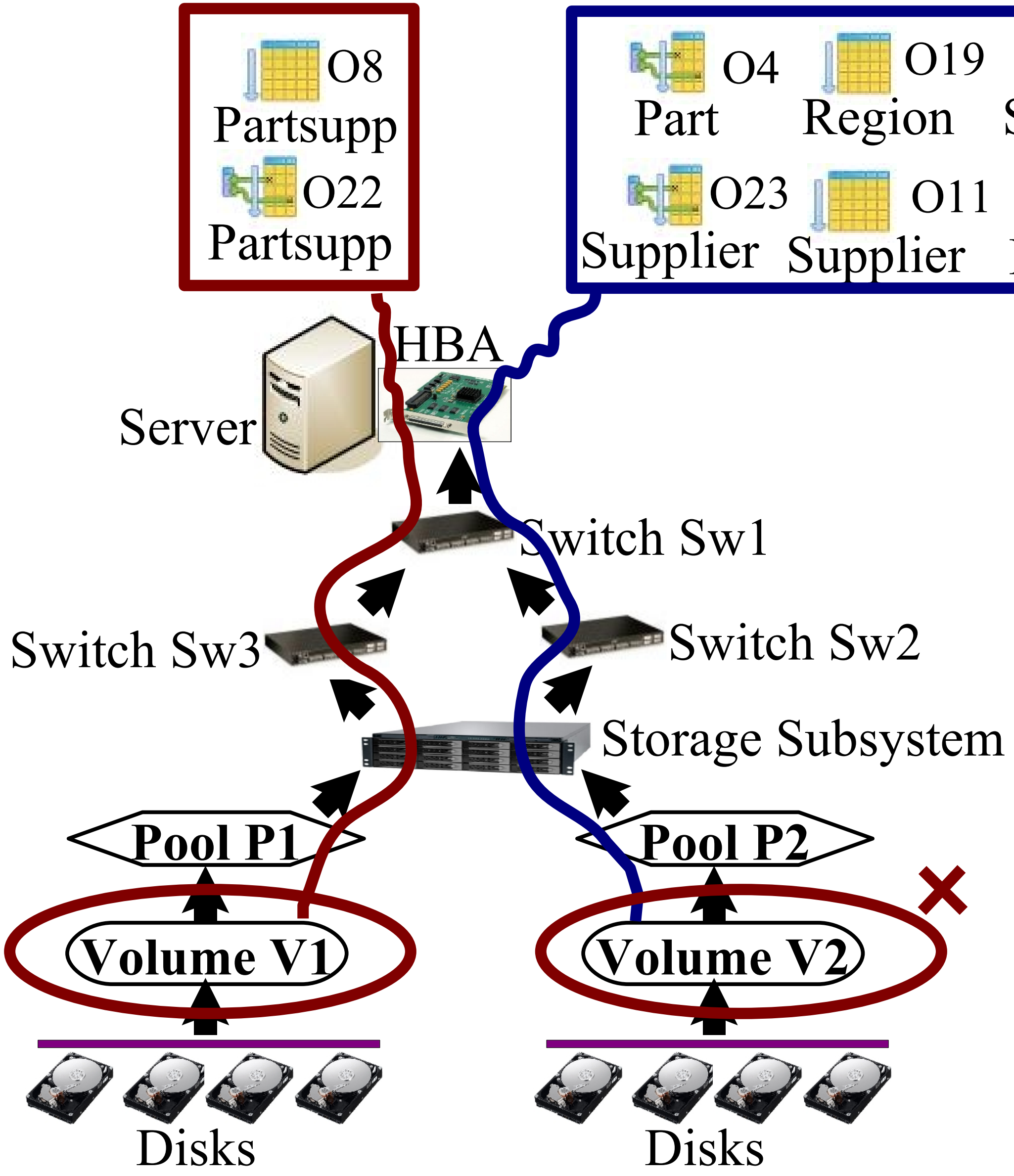
- Symptom Database
  - SAN misconfiguration

Recap of Running Example (Scenario 1)



- Impact analysis
  - High score

Scenario 2



O8 Partsupp

O22 Partsupp

O4 Part

O19 Region

O25 Supplier

O14 Nation

O23 Supplier

O11 Supplier

O16 Region

HBA

Server

Switch Sw1

Switch Sw3

Switch Sw2

Storage Subsystem

Pool P1

Pool P2

Volume V1

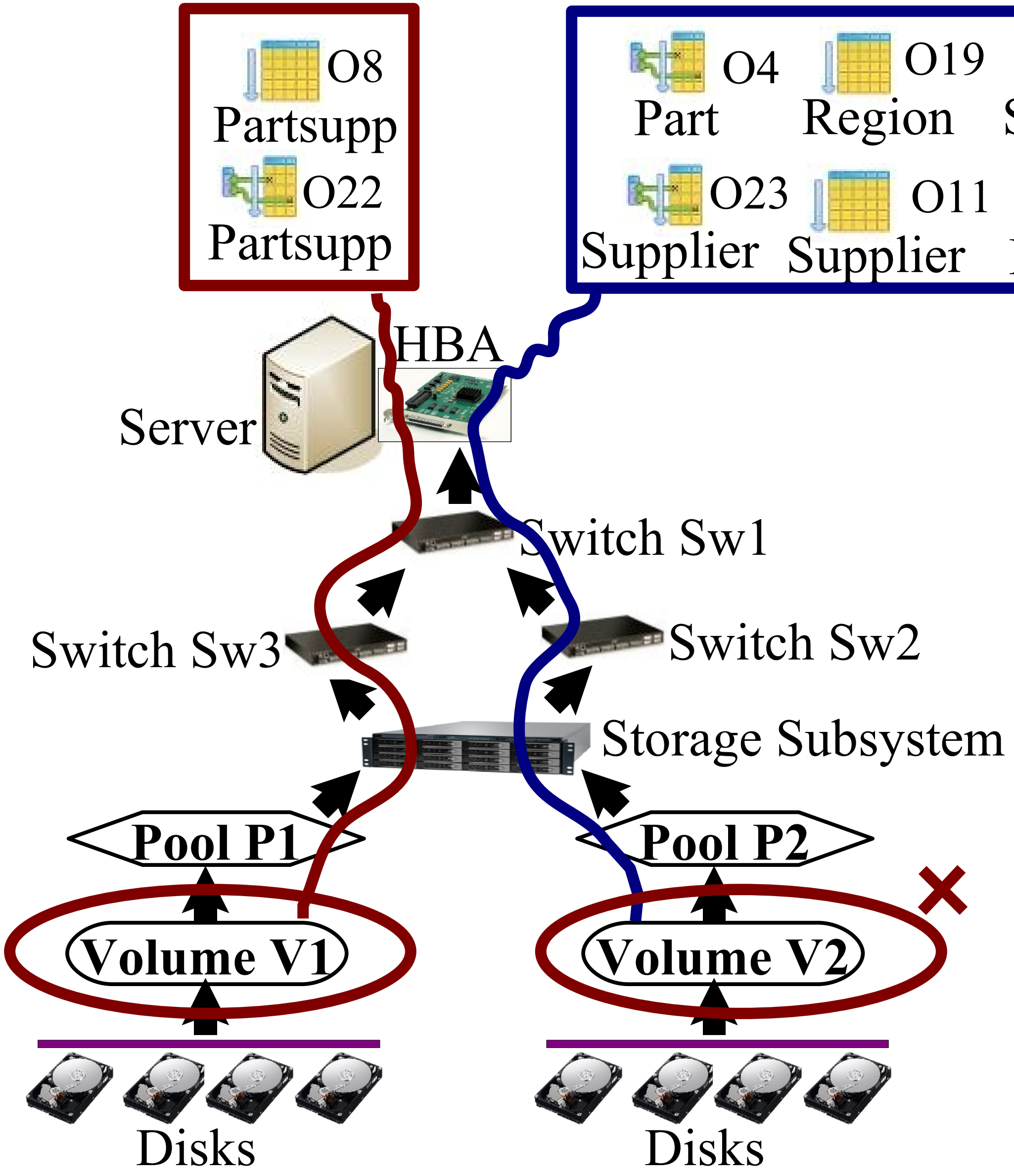
Volume V2

Disks

Disks

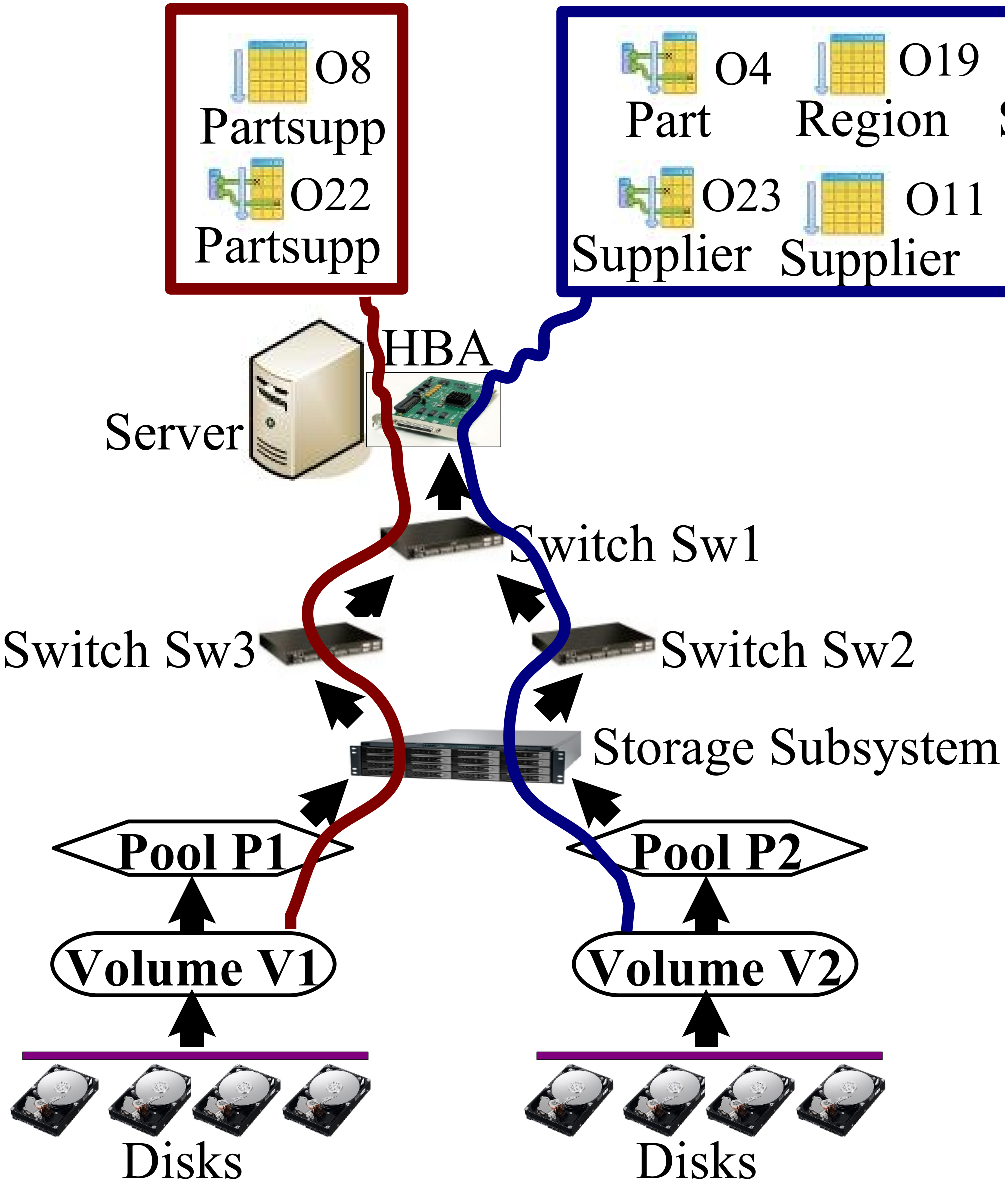
- Problem
  - Concurrent IO
  - In bursty manner
  - Query is not affected
- SAN-only tool will fail to distinguish between the two causes

Scenario 2



- Correlated Operators
- Symptom Database
  - V1 misconfiguration – High confidence score
  - V2 workload – low confidence score

Other Scenarios



- Change in data properties
- With or without concurrent SAN problems
- Spurious/missing symptoms
- More details in the paper

Related work

- DBMS level diagnosis
  - For example: Dageville et al. [VLDB'04]
- SAN level diagnosis
  - For example: Genesis [ICDCS'06]
- Machine learning techniques for diagnosis
  - For example: PeerPresure [OSDI'04]
- Incorporating expert knowledge in diagnosis
  - For example: Yemini et al. [IEEE Comm. Magazine '96]

Conclusions & Future work

- DIADS
  - APG: Provides holistic view across DBMS and SAN
  - Diagnosis workflow: Careful integration of machine learning and expert knowledge
  - Can succeed where DBMS-only and SAN-only tools fail
- Future directions
  - Alternative techniques for each module
  - Automated fix recommendation
  - Other applications of DIADS, e.g., what-if for SAN changes