lfcs | Laboratory for Foundations of Computer Science

Detecting Inconsistencies in Distributed Data

Wenfei Fan
Floris Geerts
Shuai Ma
Heiko Müller

THE UNIVERSITY of EDINBURGH informatics

Agenda

- Inconsistencies in Distributed Data
- Problem Statement & Complexity
- Detection Algorithms
- Experimental Results
- Conclusion & Outlook

Motivation

Data Quality & Data Cleaning

Data of **poor quality** has negative impact on business cost, decision making, and research outcome.

Data quality tools support two tasks for data cleaning:

- Detect data of poor quality (***detection***).
- Modify data of poor quality (***repair***).

Integrity constraints restrict valid database states.

Constraint violations highlight errors (***inconsistencies***).

CFDs

Conditional Functional Dependencies (CFDs)

CFDs **extend FDs** with conditions to **restrict their scope** and allow for **more expressive data quality rules**.

Embedded FD

Definition

*A **CFD** φ is defined as* $(X \rightarrow Y, T_p)$

Pattern Tableau

Bohannon, Fan, Geerts, Jia, Kementsietsidis, ICDE 2007.

Example

Integrity Constraint

[CC, ZIP]→[Street]

in UK (CC = 44) and Netherlands (CC = 31)

D

Name	Title	CC	AC	Phone	Street	City	ZIP
Sam	DMTS	44	131	2501984	Princess Str.	EDI	EH2 4HF
Philip	DMTS	44	131	4459011	Crichton Str.	EDI	EH4 8LE
Adam	VP	44	131	1326184	Mayfield Rd.	EDI	EH4 8LE
Joe	MTS	01	908	9075271	Queensway Drive	NYC	07974
Bob	DMTS	01	908	7732134	57th St.	MH	07974
Jef	DMTS	31	20	7464774	Muntplein	AMS	1012 WR
Steven	MTS	31	20	4521633	Spuistraat	AMS	1012 WR
Bram	MTS	31	10	8974638	Kruisplein	ROT	3012 CC

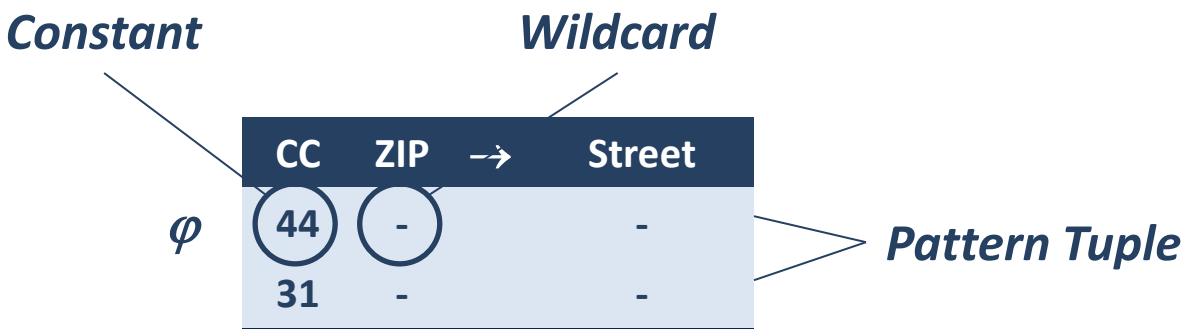
Example (cont.)

Integrity Constraint

[CC, ZIP]→[Street]

in UK (CC = 44) and Netherlands (CC = 31)

Example:



Semantics

*Only for tuples that **match the LHS** of a pattern tuple the embedded FD has to hold.*

Inconsistencies

Matching tuples

φ

CC	ZIP	→	Street
44	-		-
31	-		-

Note

*Tuples that match the LHS of a pattern tuple are potential inconsistencies (**matching** or **relevant** tuples).*

D

Name	Title	CC	AC				
Sam	DMTS	44	131				
Philip	DMTS	44	131	4459011	Crichton Str.	EDI	EH4 8LE
Adam	VP	44	131	1326184	Mayfield Rd.	EDI	EH4 8LE
Joe	MTS	01	908	9075271	Queensway Drive	NYC	07974
Bob	DMTS	01	908	7732134	57th St.	MH	07974
Jef	DMTS	31	20	7464774	Muntplein	AMS	1012 WR
Steven	MTS	31	20	4521633	Spuistraat	AMS	1012 WR
Bram	MTS	31	10	8974638	Kruisplein	ROT	3012 CC

Inconsistencies (cont.)

Constraint Violations

φ

CC	ZIP	→	Street
44	-		-
31	-		-

D

Name	Title	CC	AC	Phone	Street	City	ZIP
Sam	DMTS	44	131	2501984	Princess Str.	EDI	EH2 4HF
Philip	DMTS	44	131	4459011	Crichton Str.	EDI	EH4 8LE
Adam	VP	44	131	1326184	Mayfield Rd.	EDI	EH4 8LE
Joe	MTS	01	908	9075271	Queensway Drive	NYC	07974
Bob	DMTS	01	908	7732134	57th St.	MH	07974
Jef	DMTS	31	20	7464774	Muntplein	AMS	1012 WR
Steven	MTS	31	20	4521633	Spuistraat	AMS	1012 WR
Bram	MTS	31	10	8974638	Kruisplein	ROT	3012 CC

Classification of CFDs

We distinguish two classes of CFDs

Constant CFDs

φ_1

CC	AC	→	City
44	131		EDI
01	908		MH

Variable CFDs

φ_2

CC	ZIP	→	Street
44	-		-
31	-		-

φ_3

ZIP	→	City
-		-

Classification of CFDs

We distinguish two classes of CFDs

Constant CFDs

Variable CFDs

Decomposition of CFDs

φ_1

CC	ZIP	→	City
01	-		-
01	H0H 0H0		NORTH POLE

φ_2

CC	ZIP	→	City
01	-		-

φ_3

CC	ZIP	→	City
01	H0H 0H0		NORTH POLE

Classification of CFDs

We distinguish two classes of CFDs

Constant CFDs

φ_1

CC	AC	→	City
44	131		EDI
01	908		MH

Variable CFDs

φ_2

CC	ZIP	→	Street
44	-		-
31	-		-

φ_3

ZIP	→	City
-		-

Violations

Constant CFDs *are violated by a* ***single tuple.***

Variable CFDs *are violated by at least* ***two tuples.***

Validation

Detecting Constraint Violations

In a **centralized setting** violations of a set of CFDs can be detected using **two SQL queries**.

How to detect violations in a **distributed** setting?

Validation (cont.)

Detecting Constraint Violations (cont.)

D_1

Name	Title	CC	AC	Phone	Street	City	ZIP
Sam	DMTS	44	131	2501984	Princess Str.	EDI	EH2 4HF
Philip	DMTS	44	131	4459011	Crichton Str.	EDI	EH4 8LE
Bob	DMTS	01	908	7732134	57th St.	MH	07974
Jef	DMTS	31	20	7464774	Muntplein	AMS	1012 WR

D_2

Name	Title	CC	AC	Phone	Street	City	ZIP
Joe	MTS	01	908	9075271	Queensway Drive	NYC	07974
Steven	MTS	31	20	4521633	Spuistraat	AMS	1012 WR
Bram	MTS	31	10	8974638	Kruisplein	ROT	3012 CC

D_3

Name	Title	CC	AC	Phone	Street	City	ZIP
Adam	VP	44	131	1326184	Mayfield Rd.	EDI	EH4 8LE

Validation (cont.)

Detecting Constraint Violations (cont.)

Violations

*Inconsistent tuples may be **distributed across different sites**.*

D_1

Name	Title	CC	AC	Phone	Street	City	ZIP
Sam	DMTS	44	131				
Philip	DMTS	44	131				
Bob	DMTS	01	908				
Jef	DMTS	31	20	7464774	Muntplein	AMS	1012 WR

D_2

Name	Title	CC	AC	Phone	Street	City	ZIP
Joe	MTS	01	908	9075271	Queensway Drive	NYC	07974
Steven	MTS	31	20	4521633	Spuistraat	AMS	1012 WR
Bram	MTS	31	10	8974638	Kruisplein	ROT	3012 CC

D_3

Name	Title	CC	AC	Phone	Street	City	ZIP
Adam	VP	44	131	1326184	Mayfield Rd.	EDI	EH4 8LE

Validation (cont.)

Detecting Constraint Violations (cont.)

In a centralized setting violations of a set of CFDs can be detected using **two** SQL queries.

How to detect violations in a **distributed** setting?

Data shipment required to detect violations.

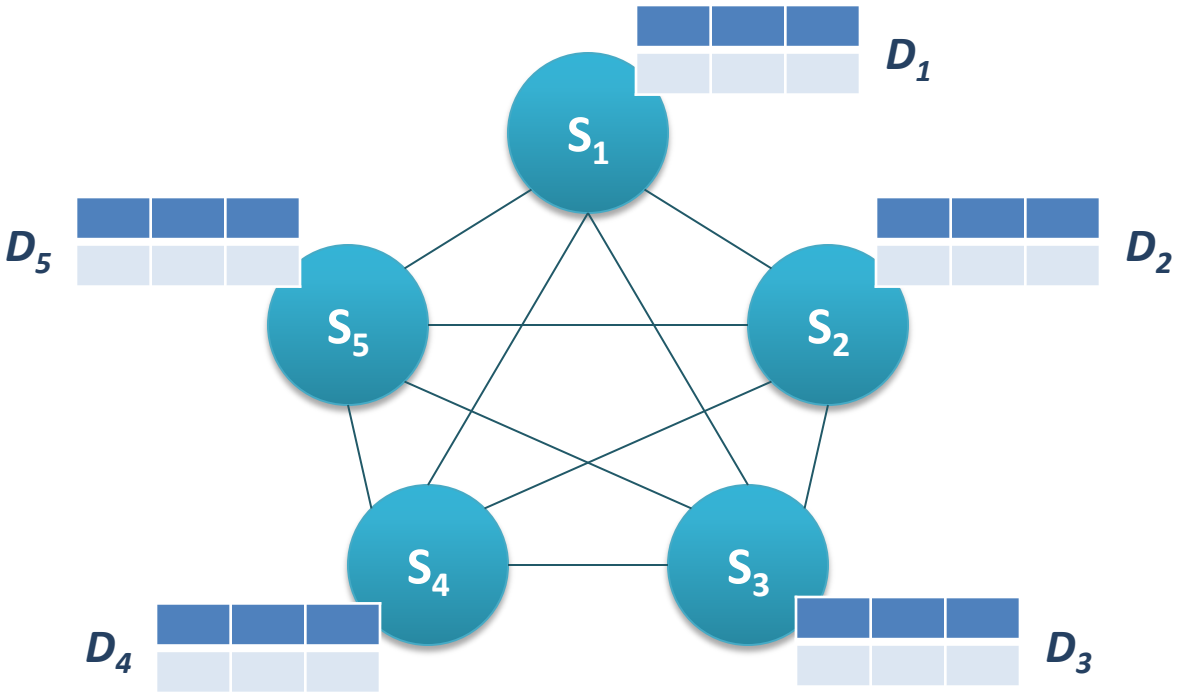
Contributions

1) *Violation detection as **optimization problem**.*
2) *Establish **complexity bounds**.*
3) *Efficient and scalable **algorithms** for violation detection.*

Problem Statement

Detecting Inconsistencies in Distributed Data

Given a set of CFDs $\Sigma = \{\varphi_1, \ldots, \varphi_n\}$, and a fragmented relation $D = \{D_1, \ldots, D_k\}$.



Problem Statement

Detecting Inconsistencies in Distributed Data

Given a set of CFDs $\Sigma = \{\varphi_1, \ldots, \varphi_n\}$, and a fragmented relation $D = \{D_1, \ldots, D_k\}$.

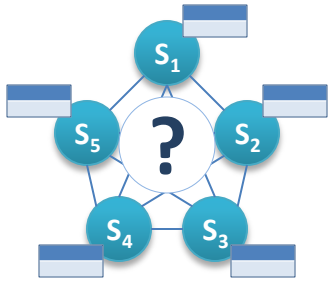
Find all violations of CFDs $\varphi \in \Sigma$ in D with …

- *minimal* ***data shipment*** (number of tuples), or
- *minimal overall* ***response time*** (shipment + local detection).

Complexity

The problem is ***NP-complete*** *in either setting!*

Distributed Data



Fragmentation of data

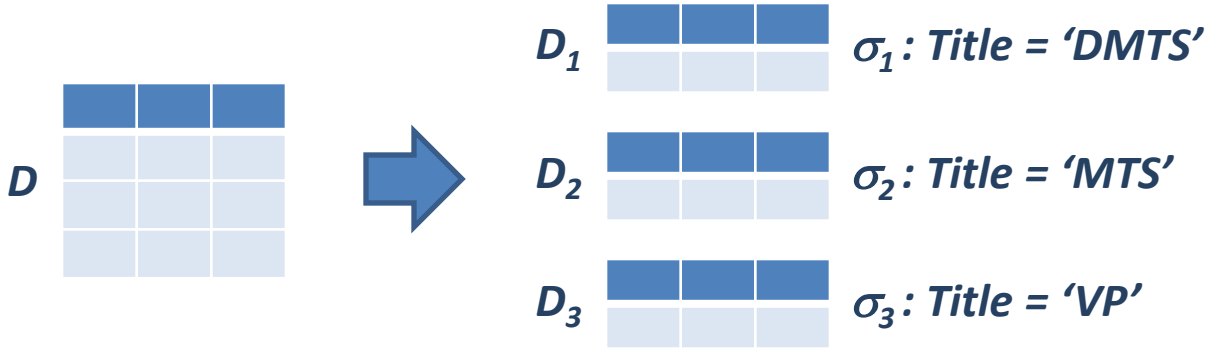
Horizontal fragmentation partitions ***D*** based on Boolean predicates $\sigma_1, \ldots, \sigma_k$ such that:

a) fragments are pair-wise disjoint, and

b) original relation ***D*** results from $\sigma_1 \cup \ldots \cup \sigma_k$.

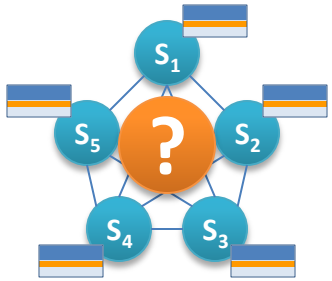
Each fragment resides at different network site.

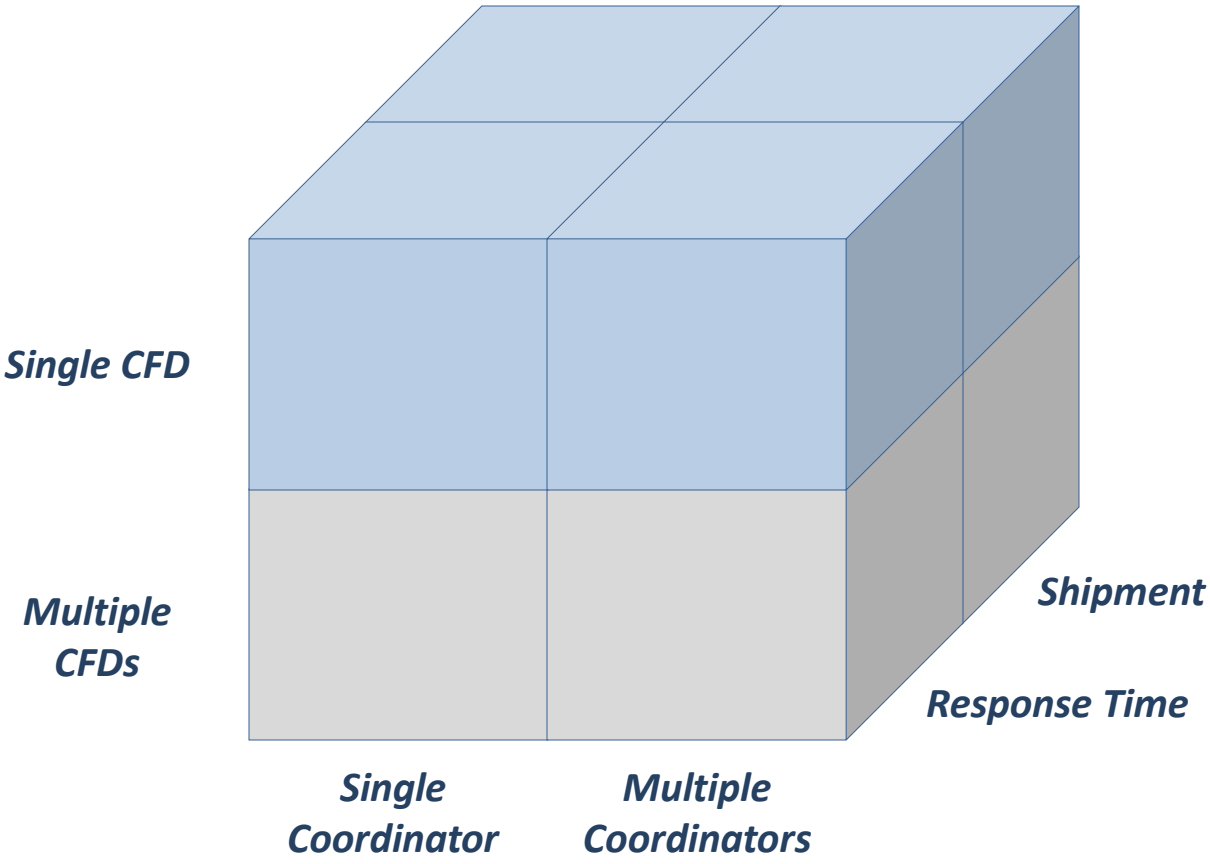
Example:



Algorithms

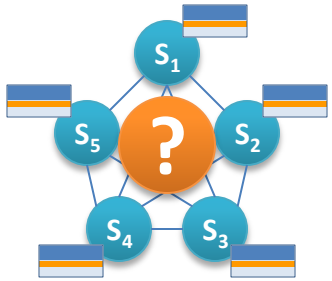
Validating Horizontally Partitioned Data





Single CFD

Local Validation of CFDs



In two cases data shipping can be avoided.

1) Constant CFDs

	CC	AC	→	City
φ	44	131		EDI
	01	908		MH

2) Partitioning Condition

	CC	Title	→	Salary
φ	-	DMTS		-

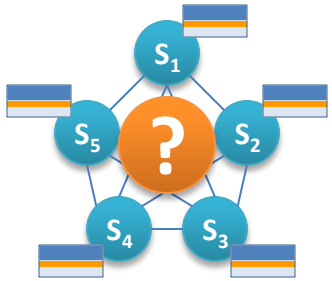
σ_1 : Title = 'DMTS'

σ_2 : Title = 'MTS'

σ_3 : Title = 'VP'

Single CFD (cont.)

Validation of CFDs involving Shipments



All algorithms are based on **four main steps:**

Algorithm Outline

1. ***Broadcast local statistics***
2. ***Decide on coordinator(s)***
3. ***Ship data to coordinator(s)***
4. ***Violation detection at coordinator(s)***

Variation (1)

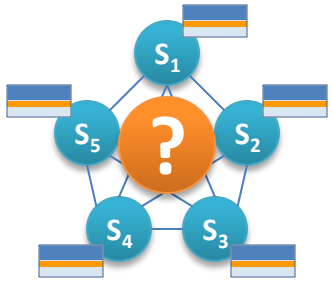
*Differ between **single coordinator** and **multiple coordinator** approach.*

Variation (2)

*Depends on **cost function**.*

Single CFD (cont.)

Central approach



Ship all relevant tuples to a **single site** (***coordinator***).

Detect violations **at coordinator** site.

CentralDetect

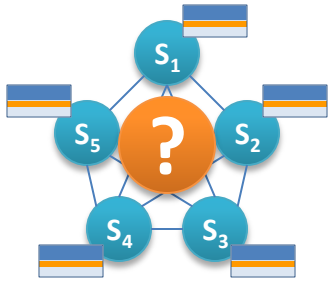
1. *Broadcast local statistics*
2. *Decide on coordinator*
3. *Ship data to coordinator*
4. *Violation detection at coordinator*

Statistics

Total number *of matching tuples.*

Single CFD (cont.)

Central approach (cont.)



φ

CC	ZIP	→	Street
44	-		-
31	-		-

D_1

Name	Title	CC	AC	Phone	Street	City	ZIP
Sam	DMTS	44	131	2501984	Princess Str.	EDI	EH2 4HF
Philip	DMTS	44	131	4459011	Crichton Str.	EDI	EH4 8LE
Bob	DMTS	01	908	7732134	57th St.	MH	07974
Jef	DMTS	31	20	7464774	Muntplein	AMS	1012 WR

3

D_2

Name	Title	CC	AC	Phone	Street	City	ZIP
Joe	MTS	01	908	9075271	Queensway Drive	NYC	07974
Steven	MTS	31	20	4521633	Spuistraat	AMS	1012 WR
Bram	MTS	31	10	8974638	Kruisplein	ROT	3012 CC

2

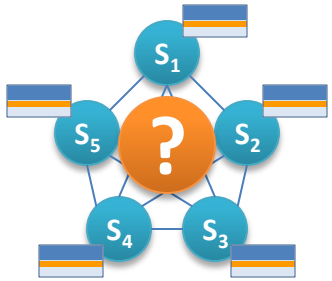
D_3

Name	Title	CC	AC	Phone	Street	City	ZIP
Adam	VP	44	131	1326184	Mayfield Rd.	EDI	EH4 8LE

1

Single CFD (cont.)

Central approach (cont.)

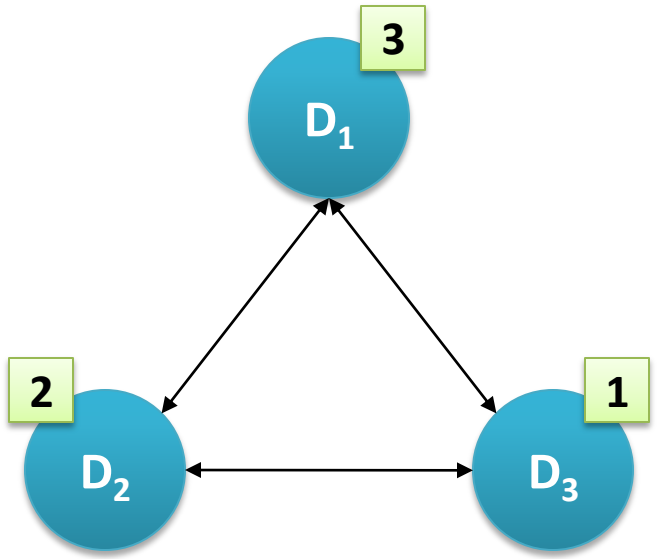


Ship all relevant tuples to a **single site** (***coordinator***).

Detect violations **at coordinator** site.

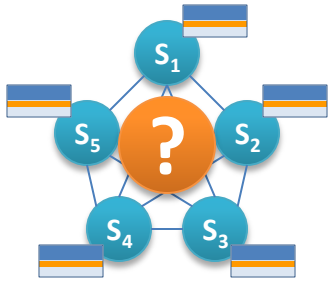
CentralDetect

1. *Broadcast local statistics*
2. *Decide on coordinator*
3. *Ship data to coordinator*
4. *Violation detection at coordinator*



Single CFD (cont.)

Central approach (cont.)

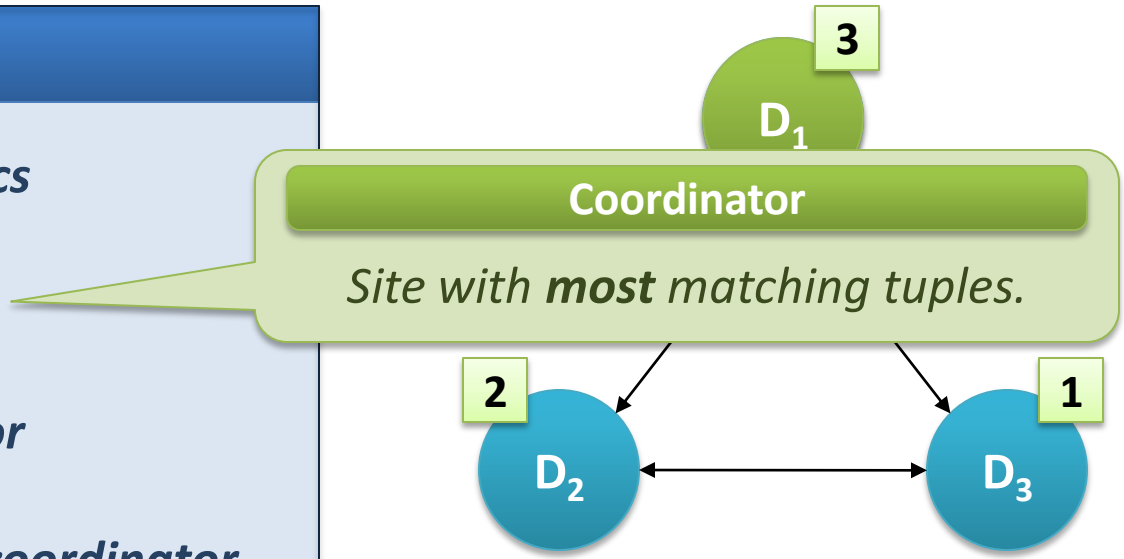


Ship all relevant tuples to a **single site** (***coordinator***).

Detect violations **at coordinator** site.

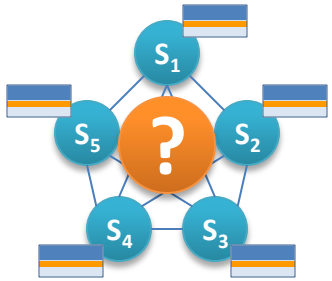
CentralDetect

1. *Broadcast local statistics*
2. *Decide on coordinator*
3. *Ship data to coordinator*
4. *Violation detection at coordinator*



Single CFD (cont.)

Pattern coordinator approach

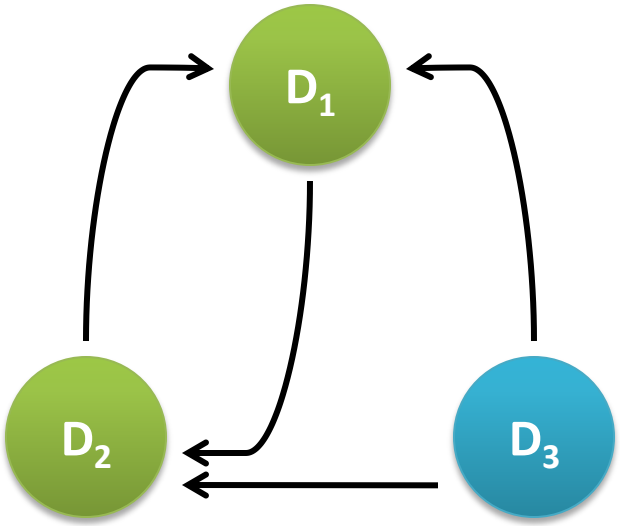


Leverage **structure of pattern tableau**.

- Assign **coordinator** for **individual pattern tuples**.
- Distribute detection process to **multiple coordinators**.

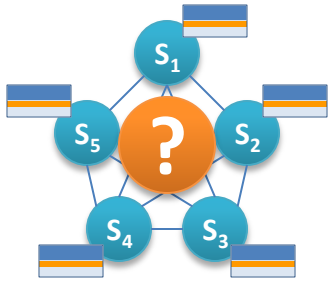
PatternDetect

1. *Broadcast local statistics*
2. *Decide on coordinator(s)*
3. *Ship data to coordinator(s)*
4. *Violation detection at coordinator(s)*



Single CFD (cont.)

Pattern coordinator approach (cont.)



How to ensure that tuples **matching multiple pattern tuples** are counted and sent **only once**?

How to ensure that we **don't miss any violations**?

	A	B	→	C
tp_1	a	-		-
tp_2	-	b		-

A	B	C
a	b	c

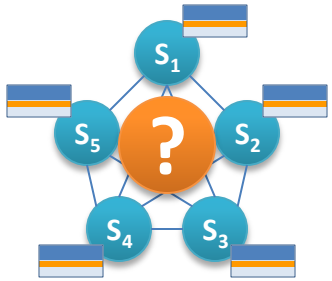
A	B	C
a	b	d

Coordinator for tp_1

Coordinator for tp_2

Single CFD (cont.)

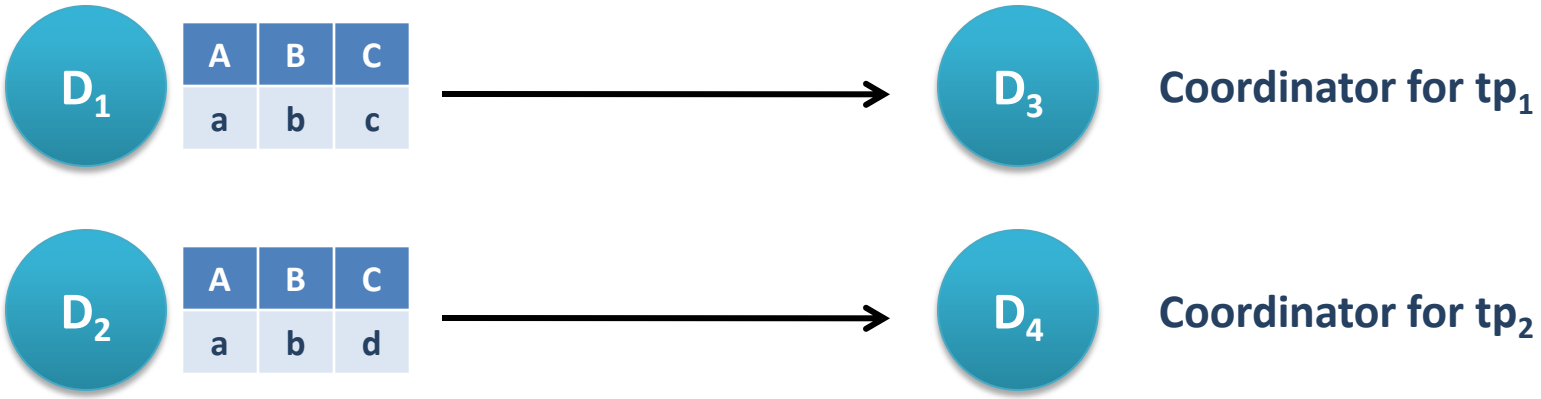
Pattern coordinator approach (cont.)



How to ensure that tuples **matching multiple pattern tuples** are counted and sent **only once**?

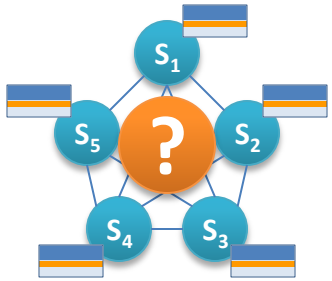
How to ensure that we **don't miss any violations**?

	A	B	→	C
tp_1	a	-		-
tp_2	-	b		-



Single CFD (cont.)

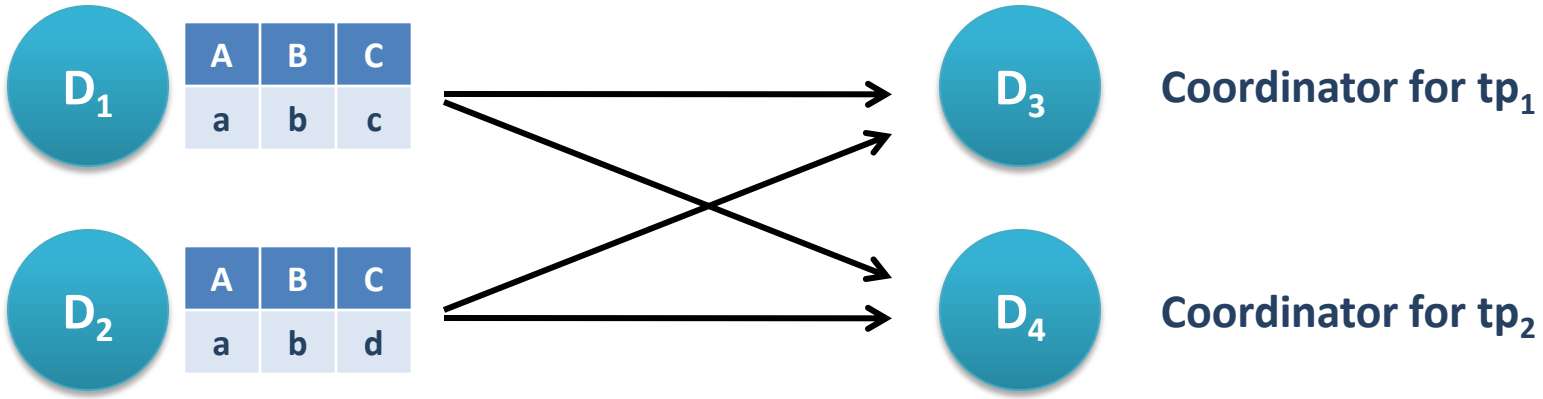
Pattern coordinator approach (cont.)



How to ensure that tuples **matching multiple pattern tuples** are counted and sent **only once**?

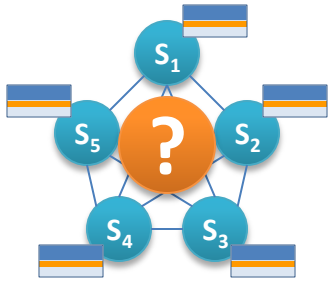
How to ensure that we **don't miss any violations**?

	A	B	→	C
tp_1	a	-		-
tp_2	-	b		-



Single CFD (cont.)

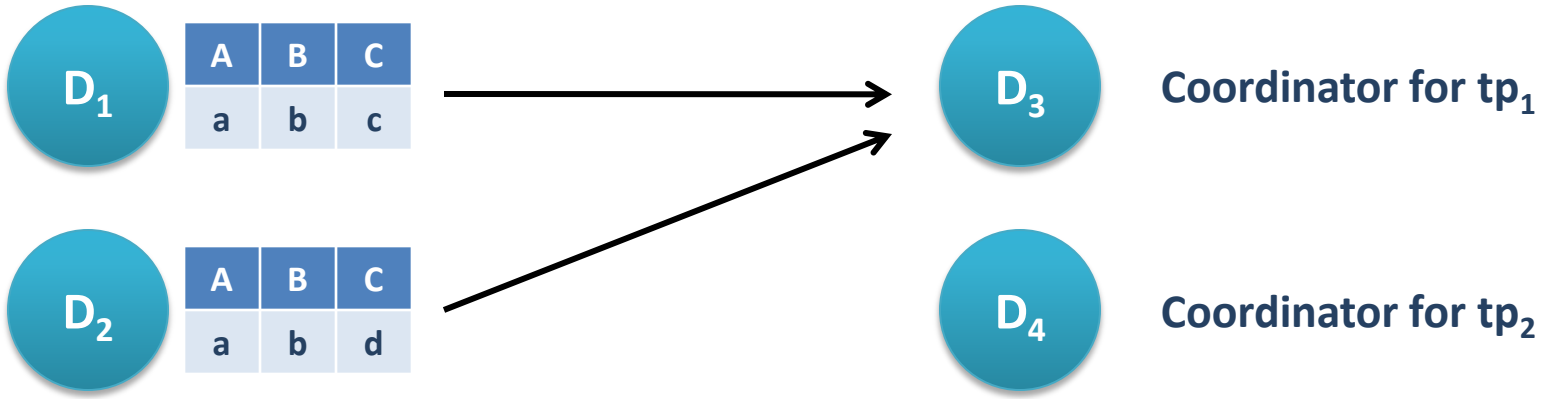
Pattern coordinator approach (cont.)



How to ensure that tuples **matching multiple pattern tuples** are counted and sent **only once**?

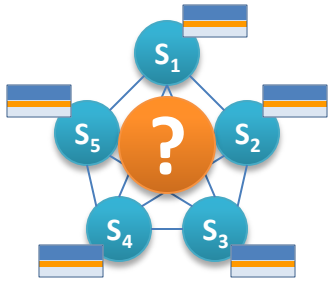
How to ensure that we **don't miss any violations**?

	A	B	→	C
tp_1	a	-		-
tp_2	-	b		-



Single CFD (cont.)

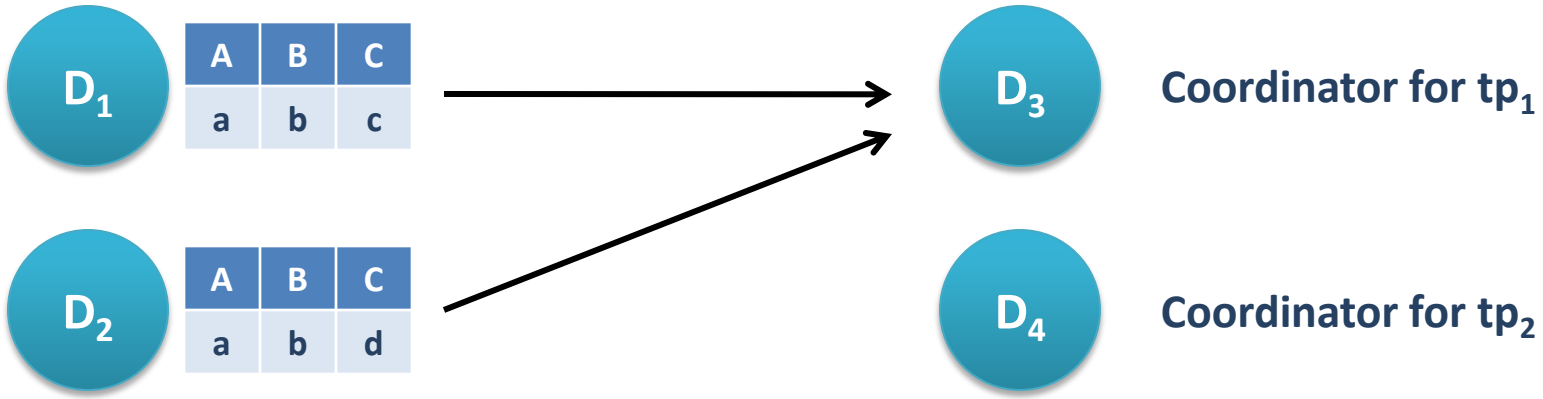
Pattern coordinator approach (cont.)



Each tuples accounts for (and is send to the coordinator of) the **first matching pattern tuple**.

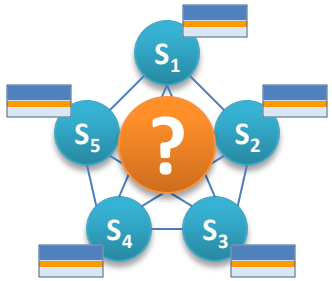
Requires a fixed **order on pattern tuples**!

	A	B	→	C
tp_1	a	-		-
tp_2	-	b		-



Single CFD (cont.)

Pattern coordinator approach (cont.)



		CC	ZIP	→	Street
φ	tp_1	44	-		-
	tp_2	31	-		-

D_1

Name	Title	CC	AC	Phone	Street	City	ZIP
Sam	DMTS	44	131	2501984	Princess Str.	EDI	EH2 4HF
Philip	DMTS	44	131	4459011	Crichton Str.	EDI	EH4 8LE
Bob	DMTS	01	908	7732134	57th St.	MH	07974
Jef	DMTS	31	20	7464774	Muntplein	AMS	1012 WR

tp_1	2
tp_2	1

D_2

Name	Title	CC	AC	Phone	Street	City	ZIP
Joe	MTS	01	908	9075271	Queensway Drive	NYC	07974
Steven	MTS	31	20	4521633	Spuistraat	AMS	1012 WR
Bram	MTS	31	10	8974638	Kruisplein	ROT	3012 CC

tp_1	0
tp_2	2

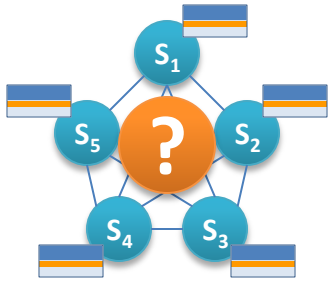
D_3

Name	Title	CC	AC	Phone	Street	City	ZIP
Adam	VP	44	131	1326184	Mayfield Rd.	EDI	EH4 8LE

tp_1	1
tp_2	0

Single CFD (cont.)

Pattern coordinator approach (cont.)

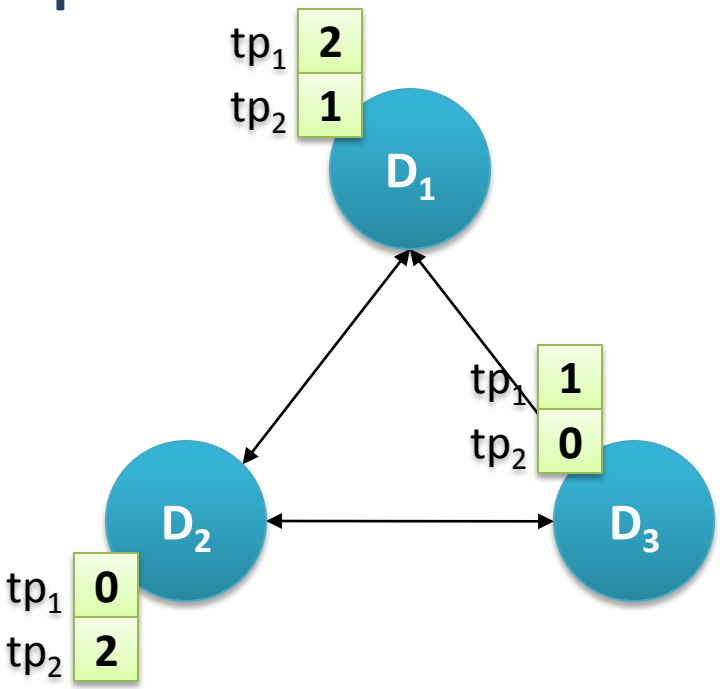


Leverage **structure of pattern tableau**.

- Assign **coordinator** for **individual pattern tuples**.
- Distribute detection process to **multiple coordinators**.

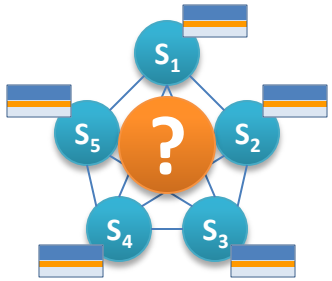
PatternDetect

1. *Broadcast local statistics*
2. *Decide on coordinator(s)*
3. *Ship data to coordinator(s)*
4. *Violation detection at coordinator(s)*



Single CFD (cont.)

Pattern coordinator approach (cont.)

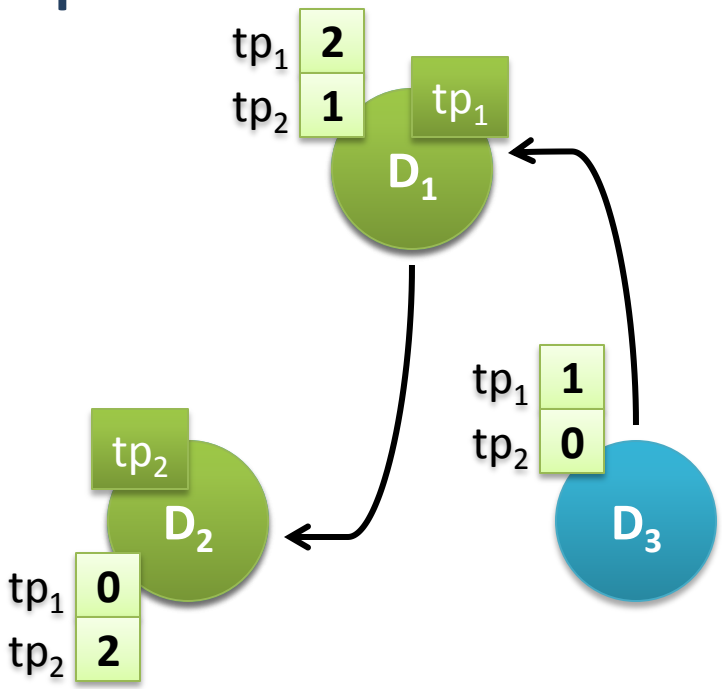


Leverage **structure of pattern tableau**.

- Assign **coordinator** for **individual pattern tuples**.
- Distribute detection process to **multiple coordinators**.

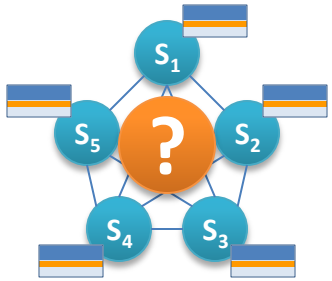
PatternDetect

1. *Broadcast local statistics*
2. *Decide on coordinator(s)*
3. *Ship data to coordinator(s)*
4. *Violation detection at coordinator(s)*



Pattern Mining

Impact of the presence of wildcards



For **sparse tableau** (e.g., **FDs**) partitioning has minor (or no) impact.

Extend pattern tableau by **instantiating wildcards** with **frequent pattern tuples**.

MinePatternDetect

1a *Mine locally for frequent patterns above given threshold*

1b *Exchange patterns and statistics*

1c *Construct extended pattern tableau*

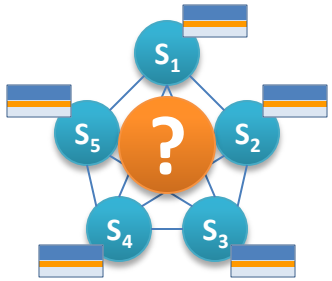
2 …

Note

*Can significantly reduce **shipment**.*

Multiple CFDs

Validating a set of CFDs



Given a set of CFDs $\Sigma = \{\varphi_1, \ldots, \varphi_n\}$.

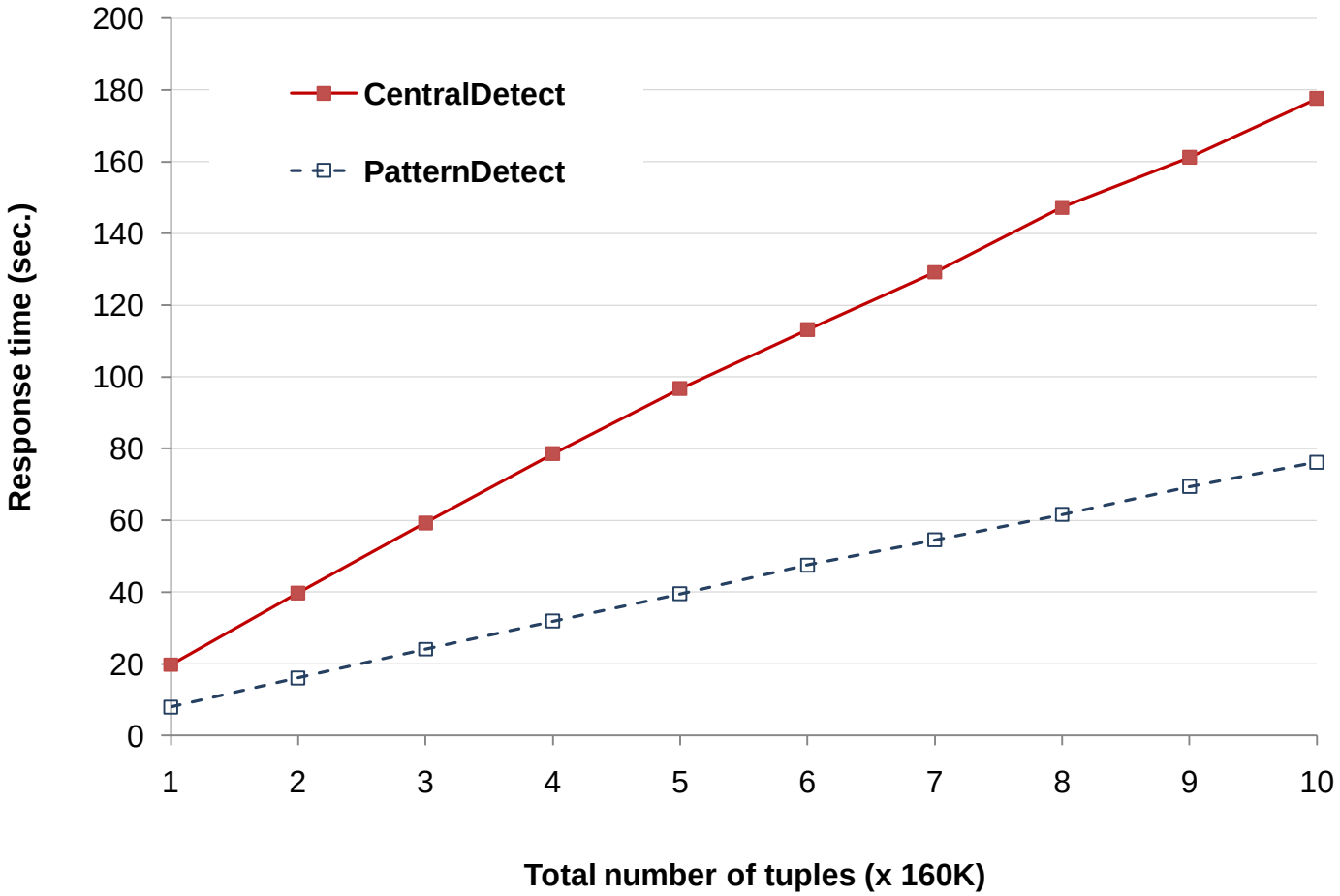
SeqDetect:
Sequentially executing one of the previous algorithms.

ClustDetect:
Reduce unnecessary data shipment by leveraging structure of **embedded FDs**.

- **Merge** CFDs that overlap on LHS.
- **Sequentially** validate merged CFDs.

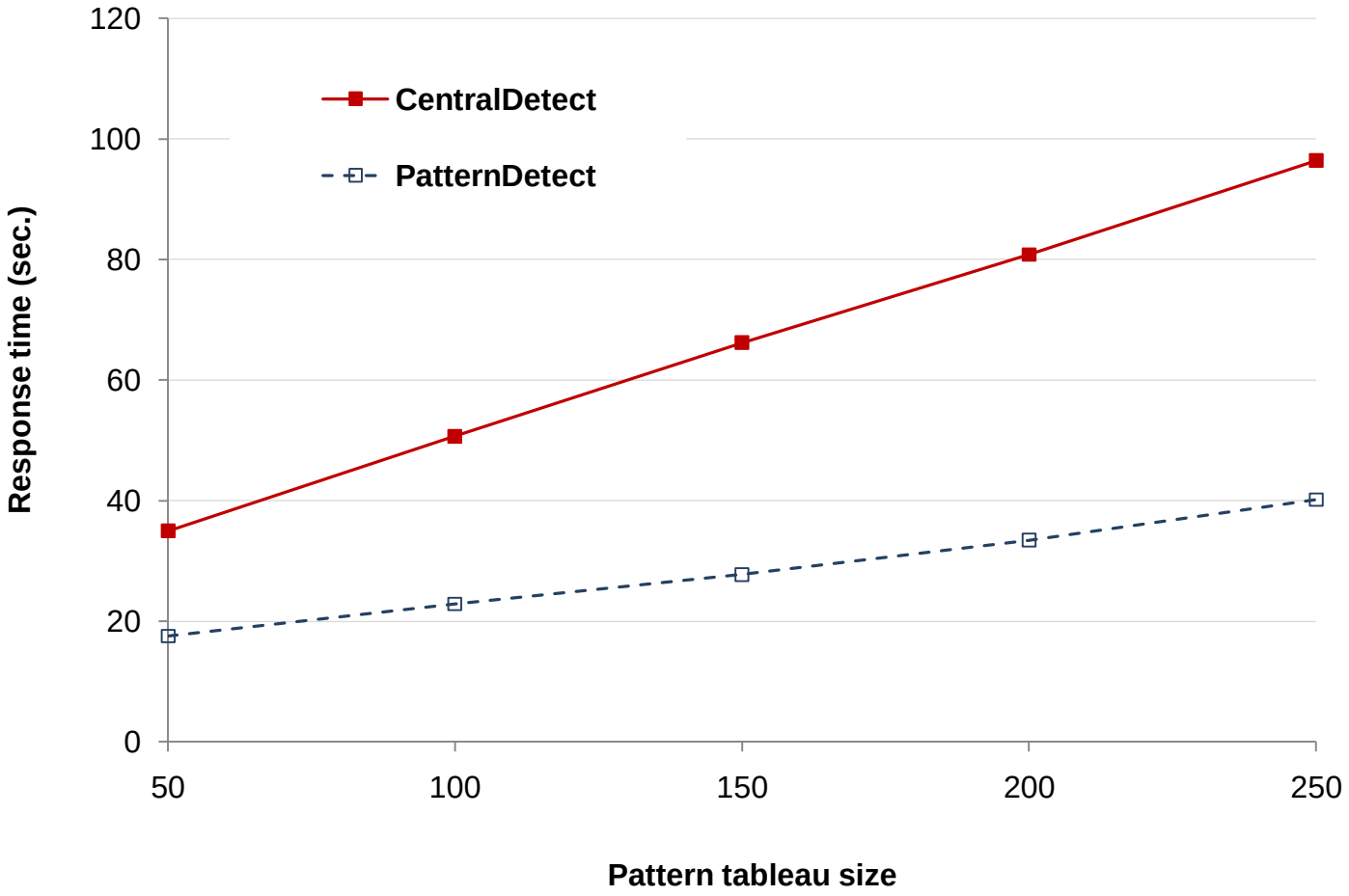
Experiments

Scalability with |D|



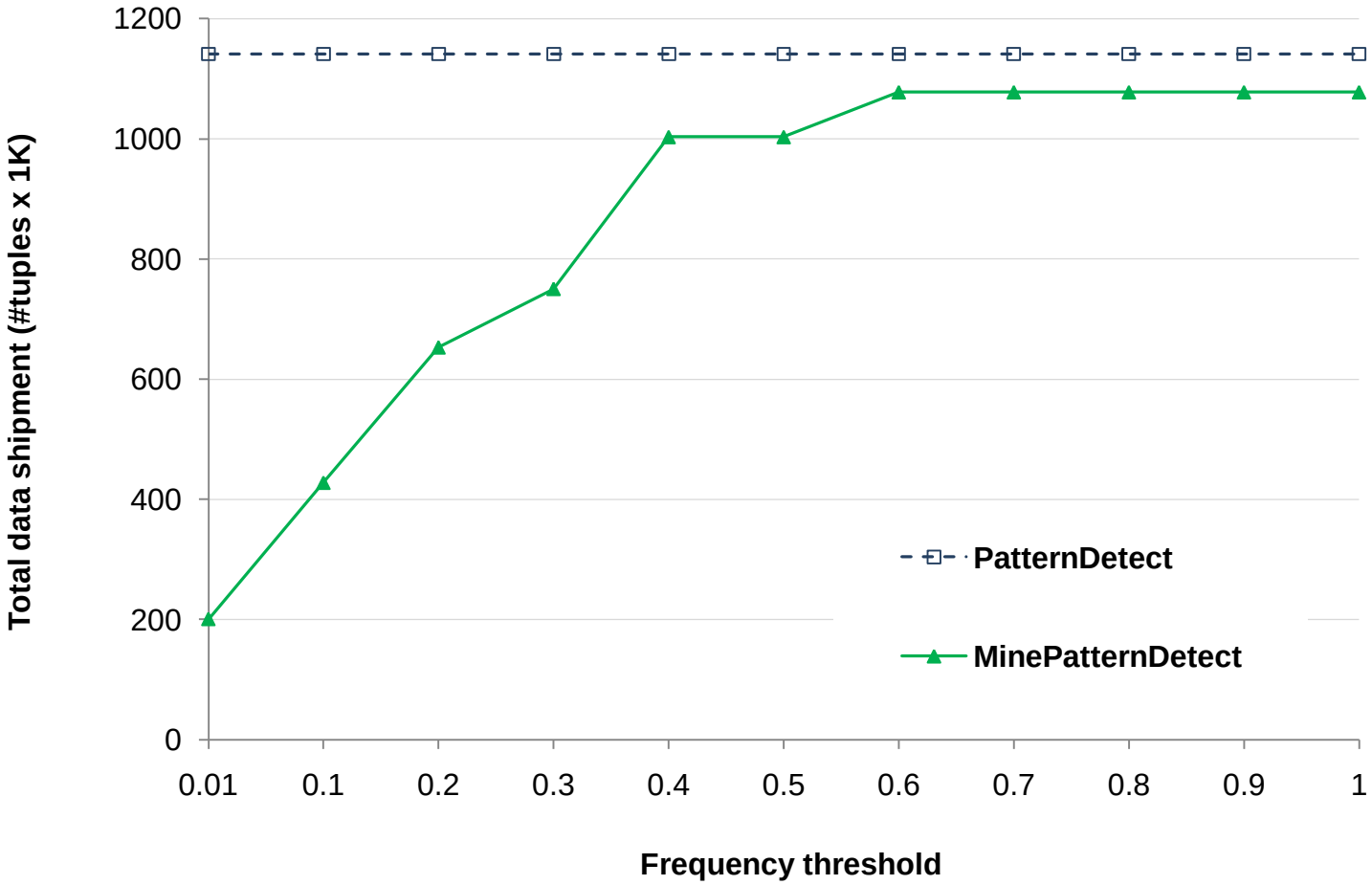
Experiments (cont.)

Scalability with $|T_p|$



Experiments (cont.)

Impact of mining on shipment



Conclusion

Conclusion

Validating CFDs over distributed data requires data shipment.

Reduce shipment and response time by leveraging **pattern tableau** and structure of **embedded FD**.

Outlook

Develop validation algorithms for **vertically (and horizontally)** partitioned data.

Agenda

- ✓ Inconsistencies in Distributed Data
- ✓ Problem Statement & Complexity
- ✓ Detection Algorithms
- ✓ Experimental Results
- ✓ Conclusion & Outlook

Cost Function

Minimizing Response Time

Shipment $M_{(i,j)}$, Data transfer rate c_t, Packet size p

Cost(D, Σ, M)

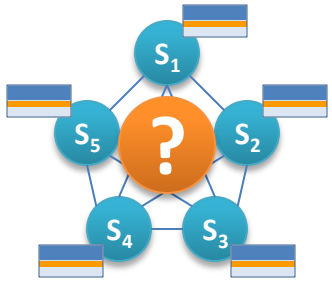
$$1/c_t \bullet \max\{\text{sum}_{i\in[1,n]}(|M_{(i,j)}|)/p\} + \max_{i\in[1,n]}\{\text{check}(D'_i, \Sigma)\}$$

(1) Maximum time taken by each site to **send data**.

(2) Maximum time for each site to **detect violations**.

Single CFD (cont.)

Pattern coordinator approach (cont.)



Each tuples accounts for (and is send to the coordinator of) the **first matching pattern tuple**.

Requires a fixed **order on pattern tuples**!

Preferred order has **constants before wildcards** ('-').

Example:

ORDER BY A, B

A	B	→	C
a	b		-
-	b		-
-	-		-

A	B	→	C
-	b		-
a	b		-
-	-		-

A	B	→	C
-	-		-
a	b		-
-	b		-

Experiments

Datasets

Synthetic data (company's sales records)

- 800K and 1600K in size, ~10% inconsistencies.

Real-life data (Ensembl cross-reference data)

- 800K in size, < 2% inconsistencies.

CFDs

4 CFDs for each dataset having 3-5 attributes.