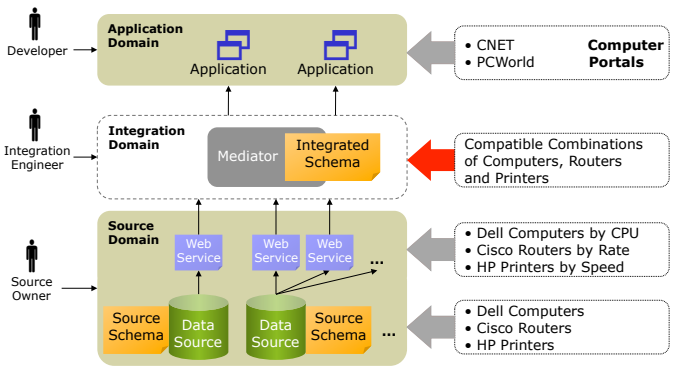
Exporting and Interactively Querying Web Service-Accessed Sources: The CLIDE System

Michalis Petropoulos

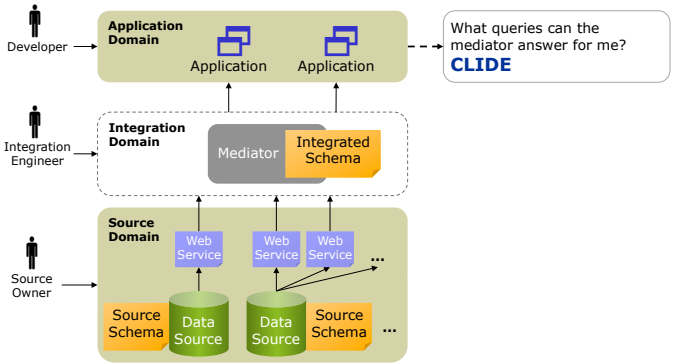
University at Buffalo
The State University of New York

Database Seminar, February 2010

Large-Scale Data Integration Systems



Large-Scale Data Integration Systems



Running Example

Parameterized Views

Dell

Schema

Computers(cid, cpu, ram, price)
NetCards(cid, rate, standard, interface)

Views

V1 ***ComByCpu(cpu) → (Computer)****

SELECT DISTINCT Com1.*
FROM Computers Com1
WHERE Com1.cpu=***cpu***

V2 ***ComNetByCpuRate(cpu, rate) → (Computer, NetCard)****

SELECT DISTINCT Com1.*, Net1.*
FROM Computers Com1, Network Net1
WHERE Com1.cid=Net1.cid
AND Com1.cpu=***cpu***
AND Net1.rate=***rate***

Computers for a given *cpu*

Computers & NetCards for a given *cpu* & *rate*

Cisco

Schema

Routers(rate, standard, price, type)

Views

Wired() → (Router)*

SELECT DISTINCT Rou1.*
FROM Routers Rou1
WHERE Rou1.type=**'Wired'**

Wired Routers

Wireless Routers

) → (Router)*

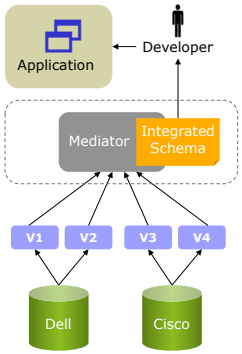
SELECT DISTINCT Rou1.*
FROM Routers Rou1
WHERE Rou1.type=**'Wireless'**

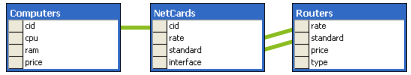
Conjunctive Queries CQ

- Equality & Comparison Conditions
- Parameters

Running Example

Integrated Schema





- Integrated schema puts together the Dell and Cisco schemas

Attribute Associations

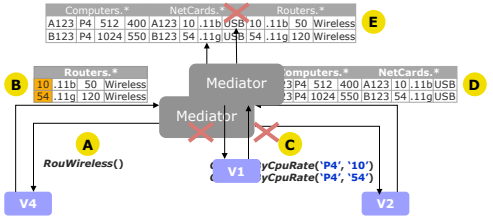
- (Computers.cid, NetCards.cid)
- (NetCards.rate, Routers.rate)
- (NetCards.standard, Routers.standard)

Sophisticated Mediators Make Feasible Queries Hard to Predict

Feasible Queries FQ

- Equivalent CQ query rewritings using the views
- Might involve more than one views
- Order might matter

Query: Feasible
Get all Computers, together with their **NetCards** and their compatible **'Wireless' Routers**



Problem

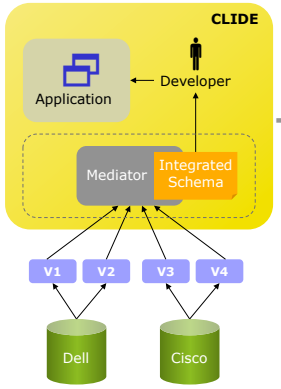
1. Large number of sources
2. Large number of views (web-services)
3. Mediator capabilities

Developer formulates an application query
➔ Is an application query feasible?
➔ If not, how do I know which ones are feasible?

Previous options:
- The developer had to browse the view definitions and somehow formulate a feasible query
- Or formulate queries until a feasible one is found (trial-and-error)

No system-provided guidance

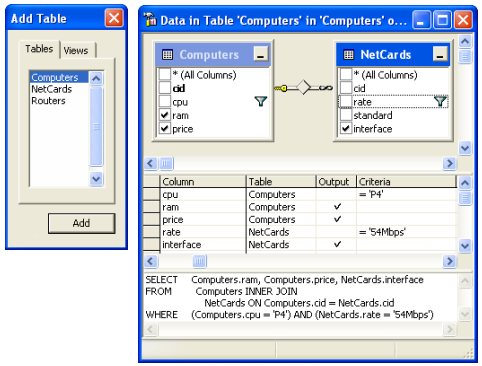
The CLIDE Solution



➔ A query formulation interface, which interactively guides the developer toward feasible queries by employing a coloring scheme

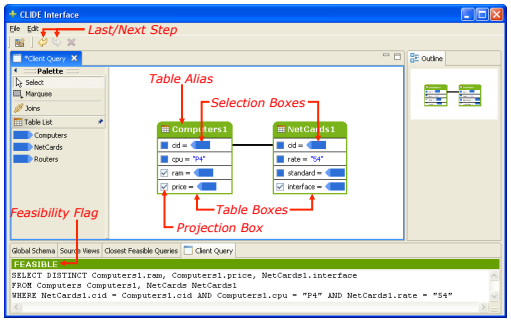
QBE-Like Interfaces

Microsoft SQL-Server



9

CLIDE Interface

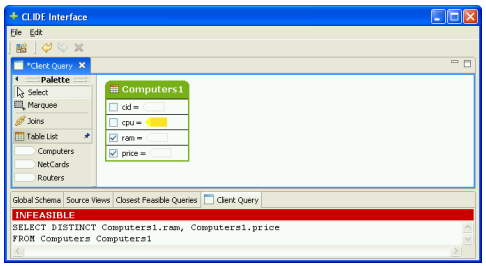


- Table, selection, projection and join actions
- Feasibility Flag
- Color-based suggestions

10

Example Interaction

Snapshot 1



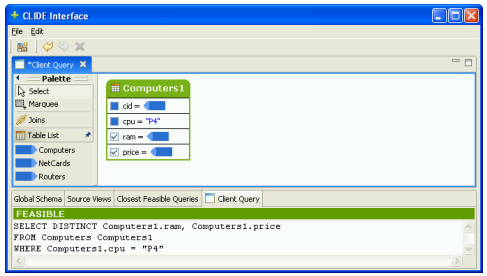
Yellow ➔ required action
- All feasible queries require this action

White ➔ optional action
- Feasible queries can be formulated w/ or w/o these actions

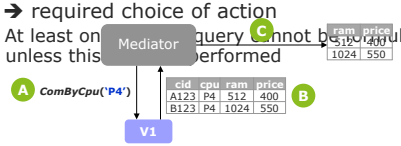
11

Example Interaction

Snapshot 2



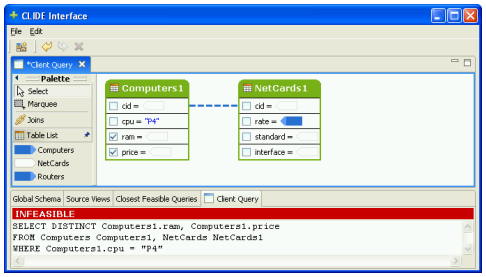
Blue ➔ required choice of action
- At least on query cannot be formulated unless this performed



12

Example Interaction

Snapshot 3

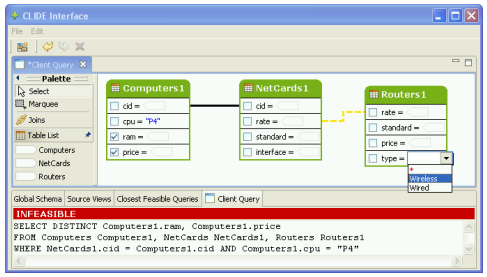


Join Lines:

- Only yellow and blue are displayed
- Must appear in Attribute Associations

Example Interaction

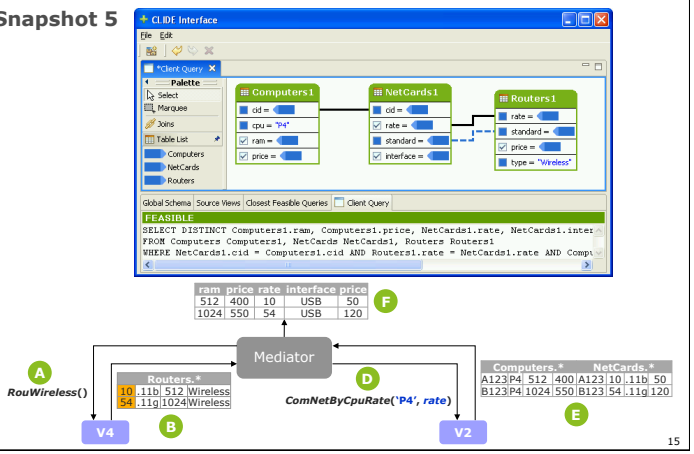
Snapshot 4



- * ➔ any other constant
- **Red** ➔ prohibited action
  - Does not appear in any feasible query
  - Lead to "Dead End" state

Example Interaction

Snapshot 5



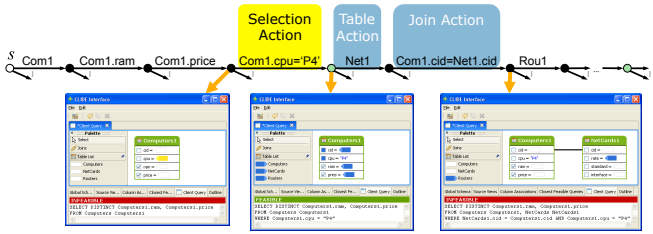
Demo

CLIDE Properties

- **Completeness of Suggestions**
  - Every feasible query can be formulated by performing yellow and blue actions at every step
- **Summarization of Suggestions**
  - At every step, only a minimal number of actions is suggested, i.e., the ones that are needed to preserve completeness
- **Rapid Convergence By Following Suggestions**
  - The shortest sequence of actions from a query to any feasible query consists of suggested actions

17

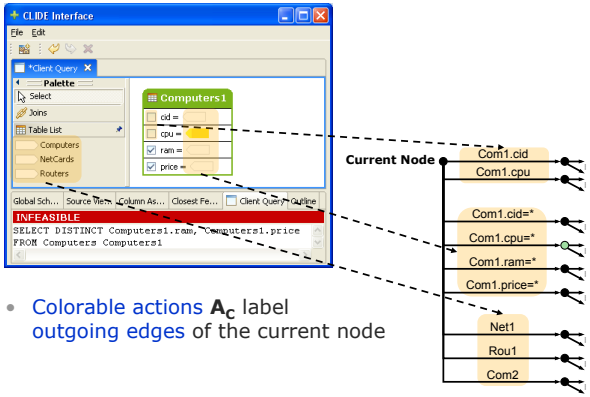
Interaction Graph



- Nodes are queries: One for each q∈CQ
- Edges are actions: Table, selection, projection and join actions
- Green nodes are feasible queries
- Infinitely big structure
  - All CQ queries
  - All possible combinations of actions formulating them

18

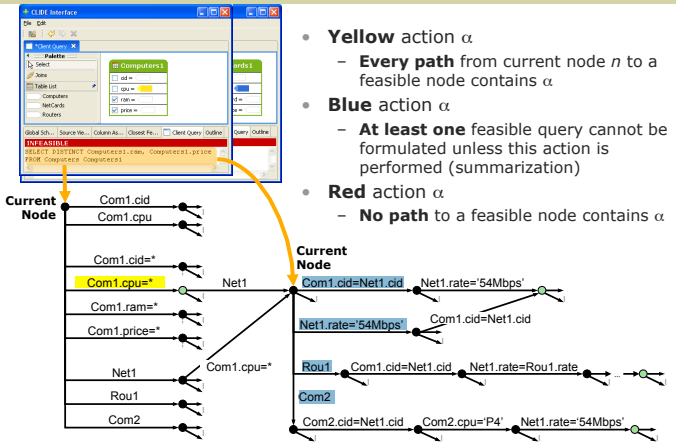
Interaction Graph: Colorable Actions



- Colorable actions $\mathbf{A_C}$ label outgoing edges of the current node

19

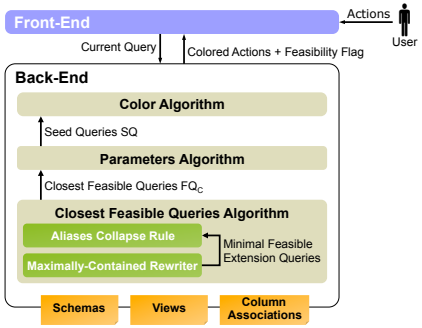
Interaction Graph: Colors



- **Yellow** action α
  - **Every path** from current node *n* to a feasible node contains α
- **Blue** action α
  - **At least one** feasible query cannot be formulated unless this action is performed (summarization)
- **Red** action α
  - **No path** to a feasible node contains α

20

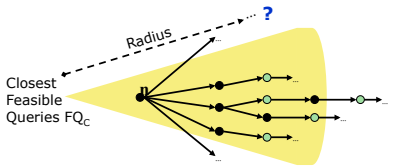
CLIDE Architecture



- Back-End invoked every time the user performs an action
  - i.e., the user arrives at a new node in the interactions graph

Color Determined By a Finite Set of Feasible Queries

Challenge: Infinitely Many Feasible Queries



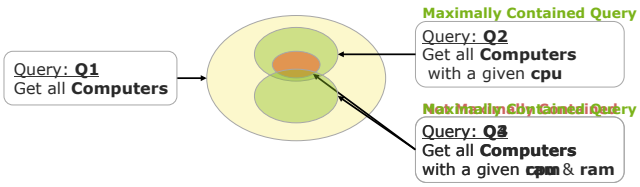
Solution: Closest Feasible Queries FQ_C

- FQ_C is sufficient to color actions in $\mathbf{A_C}$
- **Theorem: Set of Closest Feasible Queries is Finite**

Challenge: How far can the Closest Feasible Queries FQ_C be?

Solution: Based on Maximally Contained Queries FQ_{MC}

Maximally Contained Queries FQ_{MC}

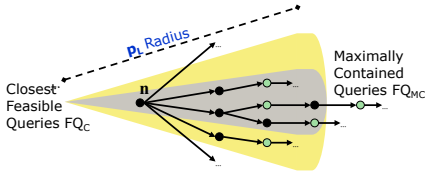


- Assuming fixed SELECT clause (projection list)
- Covered extensively in literature
  - MiniCon, Bucket, InverseRules Algorithms
- FQ_{MC} is finite

Closest Feasible Queries FQ_C Algorithm

Challenge: How far can the Closest Feasible Queries FQ_C be?

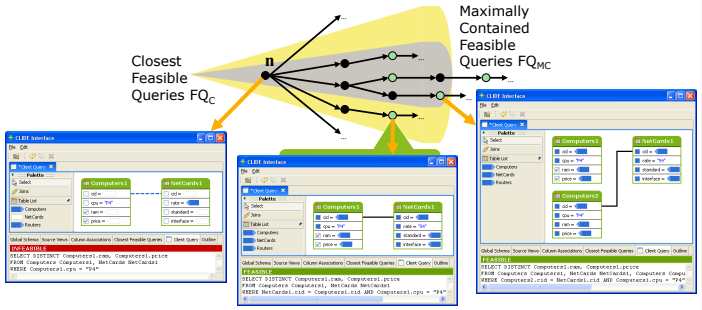
Solution: Maximally Contained Queries FQ_{MC}



- Compute maximally contained queries FQ_{MC}
- **Theorem: All FQ_C queries are reachable via a path of length $p \leq p_L$**
- The radius p_L is the longest path to a maximally contained query

Closest Feasible Queries FQ_C Algorithm

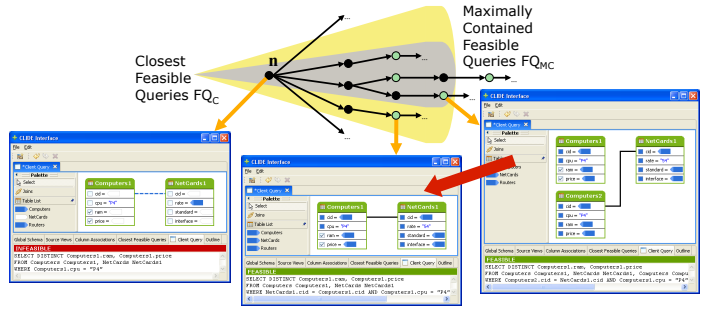
Challenge: Find the Closest Feasible Queries



- **Theorem:** All queries in FQ_{MC} are in FQ_C
- But not all queries in FQ_C are in FQ_{MC}

Closest Feasible Queries FQ_C Algorithm

Solution: Collapse Aliases



- Collapse Aliases to compute $FQ_C \setminus FQ_{MC}$
- Check satisfiability

Color Algorithm

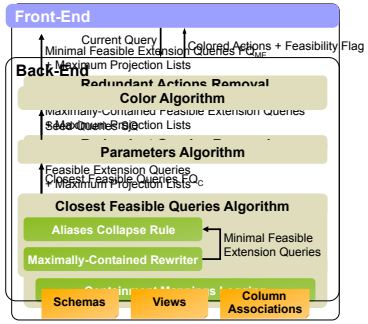
Yellow and Blue

- An action α is colored based on which closest feasible queries it appear in
- Yellow, if α appears in all queries in FQ_C
- Blue, if α appears in at least one (but not all) query in FQ_C

White and Red

- Attach *Maximum Projection Lists* to Closest Feasible Queries
  - Projections that can be added to a feasible query, without compromising feasibility
- Projection α is white if in the maximum projection list
- Color selections based on projections

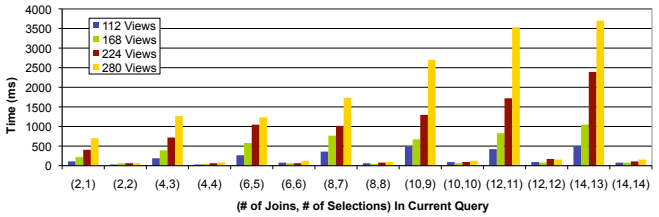
CLIDE Implementation & Optimizations



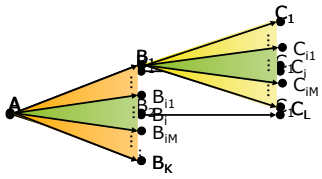
- Views expansion introduce redundancy
  - Affects CLIDE's rapid convergence and summarization
- Efficient containment test crucial to redundancy removal

CLIDE Performance

Chains of Stars – No Parameters



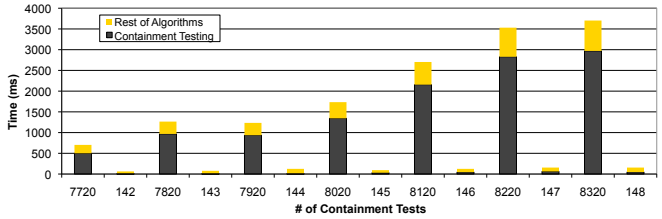
- Queries



A-span = 7
B-span = 3
Selections = 4,6,8,10

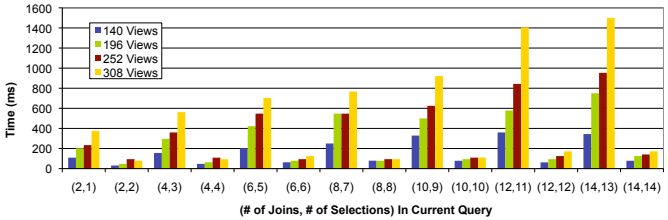
CLIDE Performance

Chains of Stars – No Parameters

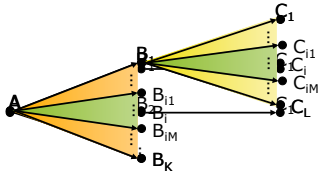


CLIDE Performance

Chains of Stars – With Parameters



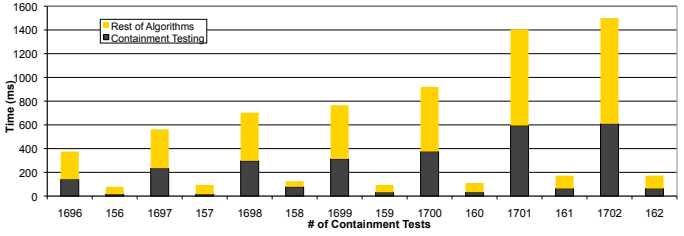
- Queries



A-span = 7
B-span = 3
Selections = 4,6,8,10

CLIDE Performance

Chains of Stars – With Parameters



CLIDE Summary

First interactive query formulation interface based on source and mediator capabilities

Applicability

- Service-Oriented Architectures
- Privacy-Preserving Services

Contributions

- Interaction Guarantees: Rapid Convergence, Completeness, Summarization of Suggestions
- Interaction Graph
- Back-End Algorithms
  - Closest Feasible Queries, Colors, Parameters
- Modular, Customizable Architecture

http://www.clide.info

33