Introduction to Prometheus Mediator

What is Prometheus

- A data integration system capable of supporting
  - Global-As-View
  - Local-As-View
  - Different types of data sources
    - Wrappers, databases, web services
    - Geospatial sources
- Why would I use Prometheus
  - Quickly integrate data from various sources
    - Example: integrate apartment information from two wrappers
  - Dynamically build integration plans/workflows to answer user queries

Installing Prometheus

- Download & install JDK
  - Most of you should already know how to do this
  - JDK 1.4 or higher will work fine
  - If you don't know how to do this
    - Look up at http://java.sun.com
- Download & install Prometheus
  - Download the prometheus.zip from the course documents
  - Unzip the file in one directory let's say c:\prometheus
- Download & install Graphviz
  - This is optional (http://www.graphviz.org/)
  - Will help in debugging mediator programs for project
  - Make sure you install it in C:\Program Files\ATT\Graphviz
- Download & install tomcat
  - Not needed for the mediator homework
  - May be useful for the project

Done Installing, now what?

- Look at the examples directory
  - C:\prometheus\binrelease\examples
- There are examples of accessing databases and web services
- Wrapper example will be provided in future if needed for project
- Each example directory has
  - A domain file
  - A query file
  - Optional source file
- Open the domain files in the database and database-lav directory
  - Update the path to the database ("d:\eclipse\workspace…") to reflect your installation path

Examples

- To run the example
- Open command prompt
- Go the prometheus directory
  - E.g. C:\prometheus\binrelease
- Type
  - runexample.bat examples/databases/domain.txt examples/databases/query.txt –o
- You should see some XML printed as a result

Let's analyze the domain file

- Three different parts
  - SourceSchemas
    - Defines the schema of a source
    - Names of attributes must match attribute names of the source
    - Names are followed by attribute type (mainly we will limit to string and numbers for now)
    - The 'b' or 'f' after the attribute type denotes if an attribute is required parameter
  - SourceTypes
    - Each source must have a source type
    - Mediator can generate plans for databases, wrappers and web services
    - First parameter of source type tells mediator what type of source it is
    - Second parameter tells mediator information about the access point
      - Database connection string or web service end point
  - Rules
    - GAV rules or LAV source descriptions

Defining GAV Rules in Prometheus

- Rules are put in the domain file
- In the rules section
- The head of a rule contains a domain relation
- The body of a rule contains one or more source/domain relations and zero or more constraints
  - D1(a,b,c):- s1(a,b)^s2(b,c)^(b>"5")
  - D2(a,"5",c):-s3(a,c)
- Each attribute of the domain relation must appear in at least one source relation or be a constant
- Constants are denote by double quotes(" ")
- The attributes of the source relation relate to the source schema by position
  - S1(a1:string:f,a2:string:f) → a1 = a and a2 = b
- Attributes with same name in the body denote join
- Rules with same head denote union

Writing Queries

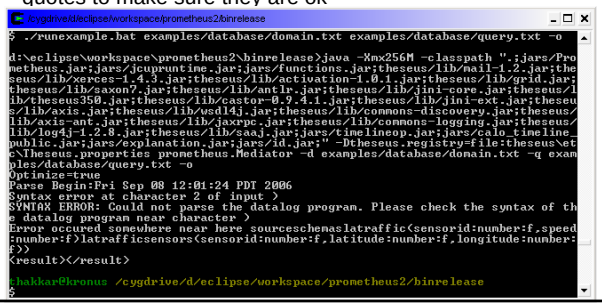
- Queries are in a separate file
- Similar to the rules
- Q1(a,b,c):-D1(a,b,c)^(c="5")
- Rules are usually generic, while queries may contain specific constraints
- For example, rules may say you can get restaurants by unioning sources s1 and s2
- A query may ask for all restaurants with cuisinetype chinese

Putting it All Together

- Creating your own model
- Create a directory under example directory
- In the directory create your domain file and query file
- Run your file by going to prometheus directory in the command prompt and typing
  - Runlocal.bat examples\myexample\domain.txt examples\myexample\query.txt -o

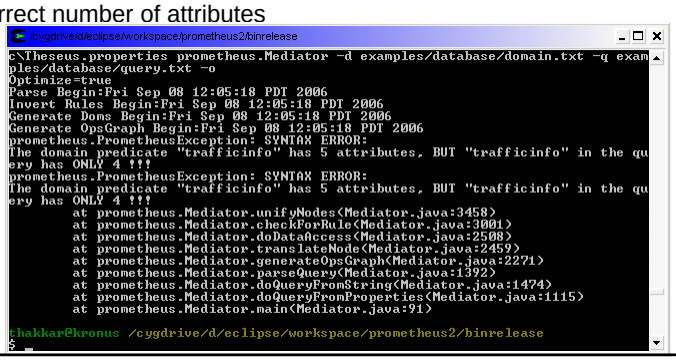
Common mistakes

- I receive complicated error that says something about syntax error or parser error
  - This means that you have either not followed syntax or are using wrong double quotes
  - Solution:
    - First thing to do is look for message error occurred somewhere....
    - Usually problem is very close to the last character shown at the end of the message
    - If you use a fancy editor open your file in notepad and replace all double quotes to make sure they are ok



Common mistakes (2)

- I receive complicated error that says something about attributes
  - This means that you have defined a source or domain relation with different number of attributes in different places
  - Solution:
    - Check all occurrences of the offending relation and make sure if has correct number of attributes



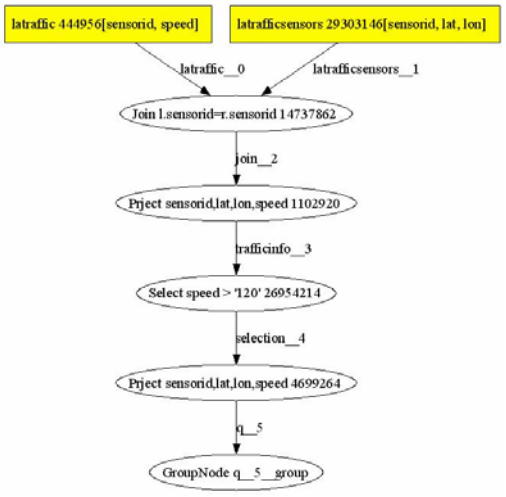
Common mistakes (3)

- I did everything ran the thing and I get empty result
  - Usually because you may have constraints that filter out everything
  - Solution:
    - Check your join conditions and selections to make sure everything looks ok
    - In general, it helps to sort of build things slowly and test after adding each new thing

FAQ

- Ok this is all good, but how do I know what the mediator is doing
  - If you look at the runexample.bat you will see that mediator can take a few arguments
  - First argument that you pass is the domain file, second is the query file
  - The –o option tells the mediator to optimize the generated plan
  - If you pass with –debug, mediator prints whole bunch of debug info
  - If you install graphviz you can run the mediator by passing –g option
    - Mediator will create jpg files showing graph of the operations

Example jpg file



Defining LAV Rules in Prometheus

- Prometheus uses the Inverse Rules algorithm
- The source descriptions are passed with the –s <sourcefile> option
- Database-lav example shows an example of LAV
- Key thing to remember in LAV for Promtheus is that joins are only inferred based based on joins in the query and joins in the domain rules

Defining LAV (Cont'd)

- Domain Rules
  - Chase rules: In Duschka's thesis chase rules are defined to reason about functional dependencies
  - For example if we know that id is a primary key in a relation, we would write a rule like the following
    - Eq(name1,name2):- d1(id1,name1)^
d1(id2,name2)^
Eq(id1,id2)
    - Note that this is the general form of equality reasoning and it is recursive
  - In prometheus we use the non-recursive and more restrictive form
    - D2(id,name2):-d1(id,name1)^d1(id,name2)
  - Recursion in Prometheus has not been implemented and tested reliably
  - Non-recursive plans are lot more efficient

Go Through LAV Example from example directory

Calling Prometheus from Your Code

- Prometheus has a java API that allows you to call it from another java program
- It can also be hosted as a Servlet through tomcat
- Information on both of those are in the documentation on totale site

Integrating Prometheus with Other Apps

- It is easy to write either java code or xquery or xslt to convert XML generated by Prometheus into KML
  - Data with lat/lons
  - Example Servlet that produces KML from the XML data is on the totale site
  - KML can be shown on Google maps or Google Earth

About the Homework

- Four parts
  - First define a model using Global-As-View
  - Define model with same sources using Local-As-View
  - Write two queries
  - Add a source to the Global-As-View model
  - Add a source to the Local-As-View model
  - Note that both your queries should run with all domain files
- Things to submit
  - <lastname><firstname>gavdomain1.txt
  - <lastname><firstname>query1.txt
  - <lastname><firstname>query2.txt
  - <lastname><firstname>lavdomain1.txt
  - <lastname><firstname>lavsource1.txt
  - <lastname><firstname>gavdomain2.txt (domain with the new source)
  - <lastname><firstname>lavdomain2.txt (domain with the new source)
  - <lastname><firstname>lavsource2.txt

About the homework (cont'd)

- Sources:
  - Tables in one database
    - Cheaprestaurants – restaurants with avgprice less than $10
    - Chineserestaurants
    - Geocoder
    - Reviews
- Your first task
  - Write a Global-As-View model with two domain relations one for restaurants and one for reviews (<lastname><firstname>gavdomain1.txt)
  - Write query files to answer the following queries
    - Find reviews for all American restaurants (<lastname><firstname>query1.txt)
    - Find restaurant info (name,address,cuisinetype,avgprice,lat,lon) for all restaurants with avgprice < $10 (<lastname><firstname>query2.txt)

About the homework (cont'd)

- Your second task
  - Write a local-As-View model with two domain relations one for restaurants and one for reviews (<lastname><firstname>lavdomain1.txt,<lastname><firstname>lavsource1.txt)
  - When we run the mediator with the local-as-view model, source description and queries you did in part 1
    - We should get same answers

About the homework (cont'd)

- Your third task
  - Add a new source restaurantsandreviews to your Global-As-View model (<lastname><firstname>gavdomain2.txt)
  - This source provides restaurant information and reviews by fodors for each restaurant
  - Run the queries you came up with in part 1 against this model and see that you get additional results from the new source

About the homework (cont'd)

- Your fourth task
  - Add a new source restaurantsandreviews to your Local-As-View model (<lastname><firstname>lavdomain2.txt, <lastname><firstname>lavsource2.txt)
  - This source provides restaurant information and reviews by fodors for each restaurant
  - Run the queries you came up with in part 1 against this model and see that you get additional results from the new source

Good Luck

- Last minute pointers
  - Start early
  - Mess around with the mediator as much as possible to get familiar
  - Use the discussion board on totale site
  - I will monitor the message boards
  - Please be patient with messages