Predicting job running time in DIRAC

Master project

DIRAC user community meeting

Barcelona, 13.05.2011

Erik Hidle

erik.hidle@cern.ch

Stephane Poss

NTNU – Trondheim
Norwegian University of
Science and Technology

Presentation Outline

- Introduction to my Master project
- Predator
  - Architecture and design
  - Challenges
  - Other systems
  - Technology
  - Project outlook
- Summary
- Conclusion

NTNU – Trondheim
Norwegian University of
Science and Technology

Master project

Goal:

Predict the job running time given the job configuration and site binding.

Project Status:

Not implemented

Motivation for the project

- Production users
  - Want to know when their jobs are finished
- Improve scheduling
  - More accurate resource reservations
  - Resource selection

Limitations

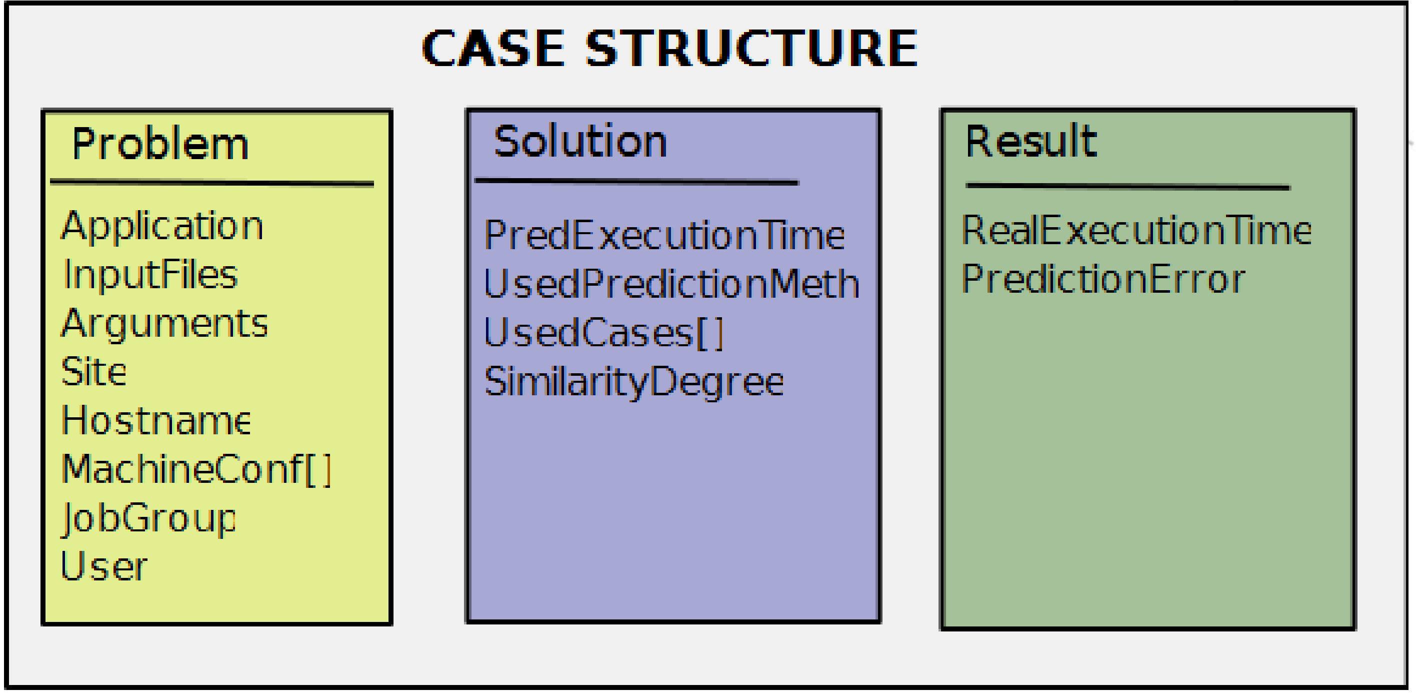
Will only consider production jobs:

- Generation
- Simulation
- Reconstruction

PREDATOR

PREDATOR

- Use Case Based Reasoning
  - Inspired by how humans solve problems
  - Learning is the process of solving problems and storing them for later reuse
  - Solve new problems by combining previous experiences
  - The knowledge of the system is the case base



CASE STRUCTURE

Problem

Application
InputFiles
Arguments
Site
Hostname
MachineConf[]
JobGroup
User

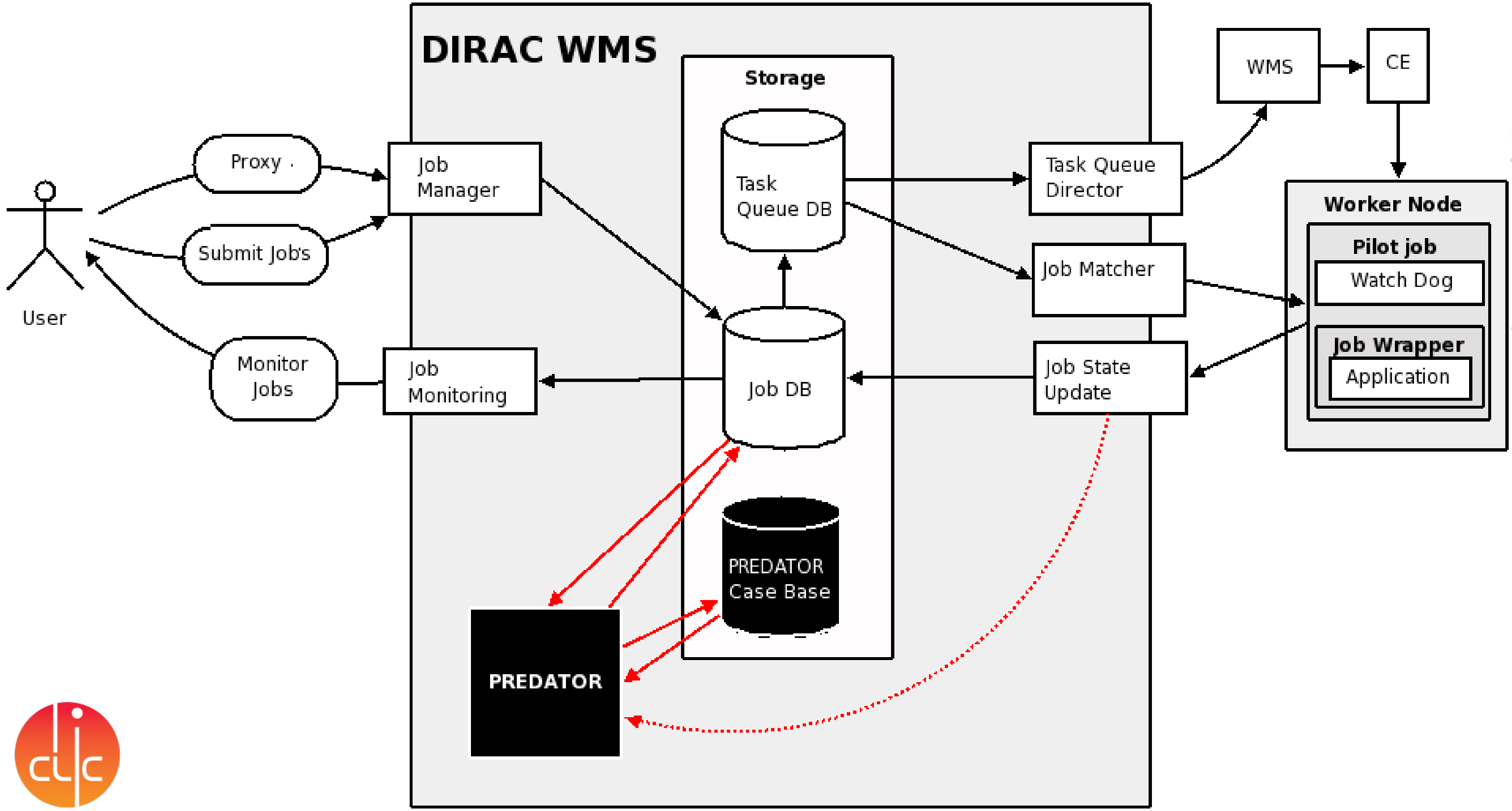
Solution

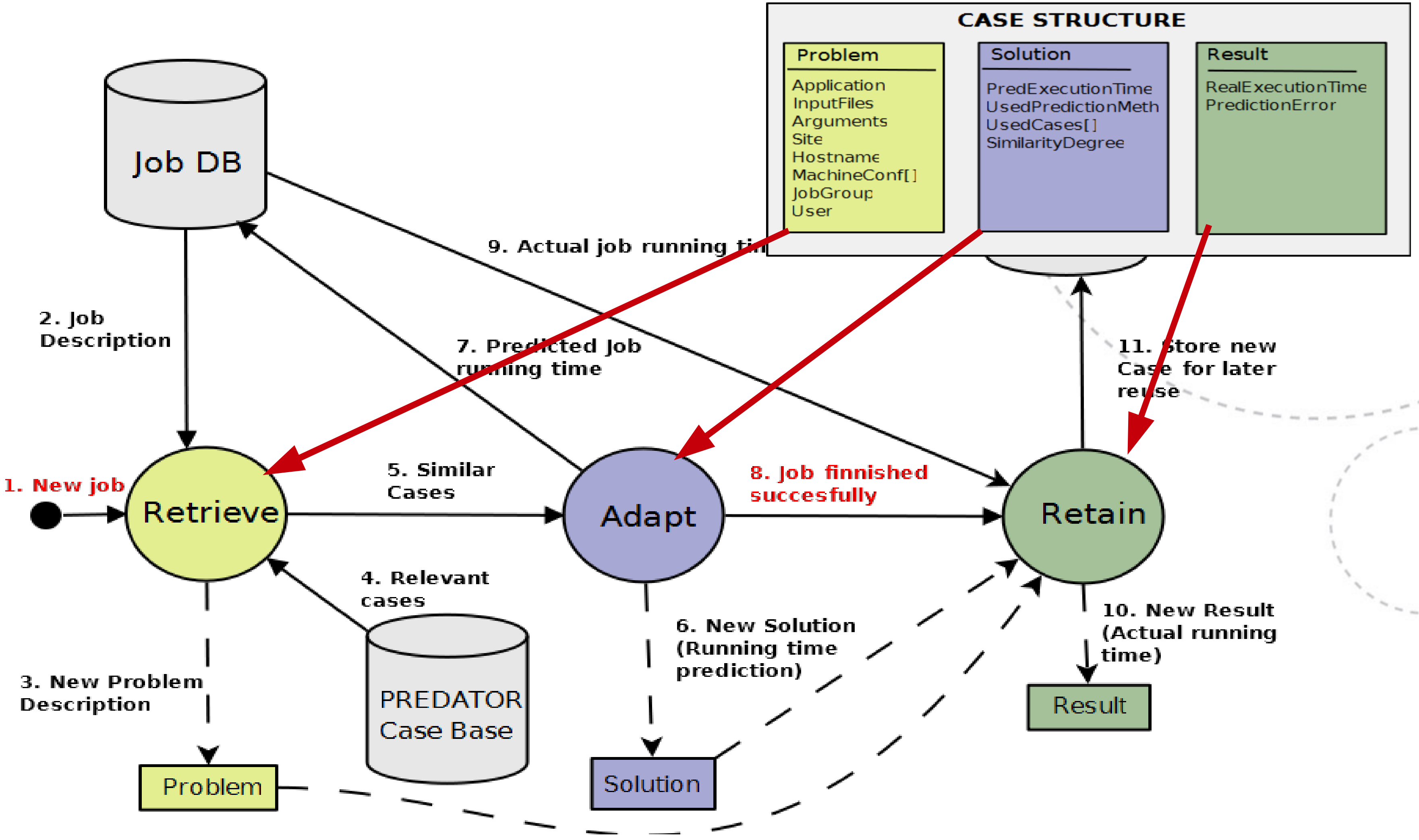
PredExecutionTime
UsedPredictionMeth
UsedCases[]
SimilarityDegree

Result

RealExecutionTime
PredictionError

PREDATOR and DIRAC WMS





CASE STRUCTURE

Problem
Application
InputFiles
Arguments
Site
Hostname
MachineConf[]
JobGroup
User

Solution
PredExecutionTime
UsedPredictionMeth
UsedCases[]
SimilarityDegree

Result
RealExecutionTime
PredictionError

Job DB

2. Job Description

9. Actual job running ti

7. Predicted Job running time

11. Store new Case for later reuse

1. New job

Retrieve

5. Similar Cases

Adapt

8. Job finnished succesfully

Retain

4. Relevant cases

6. New Solution (Running time prediction)

10. New Result (Actual running time)

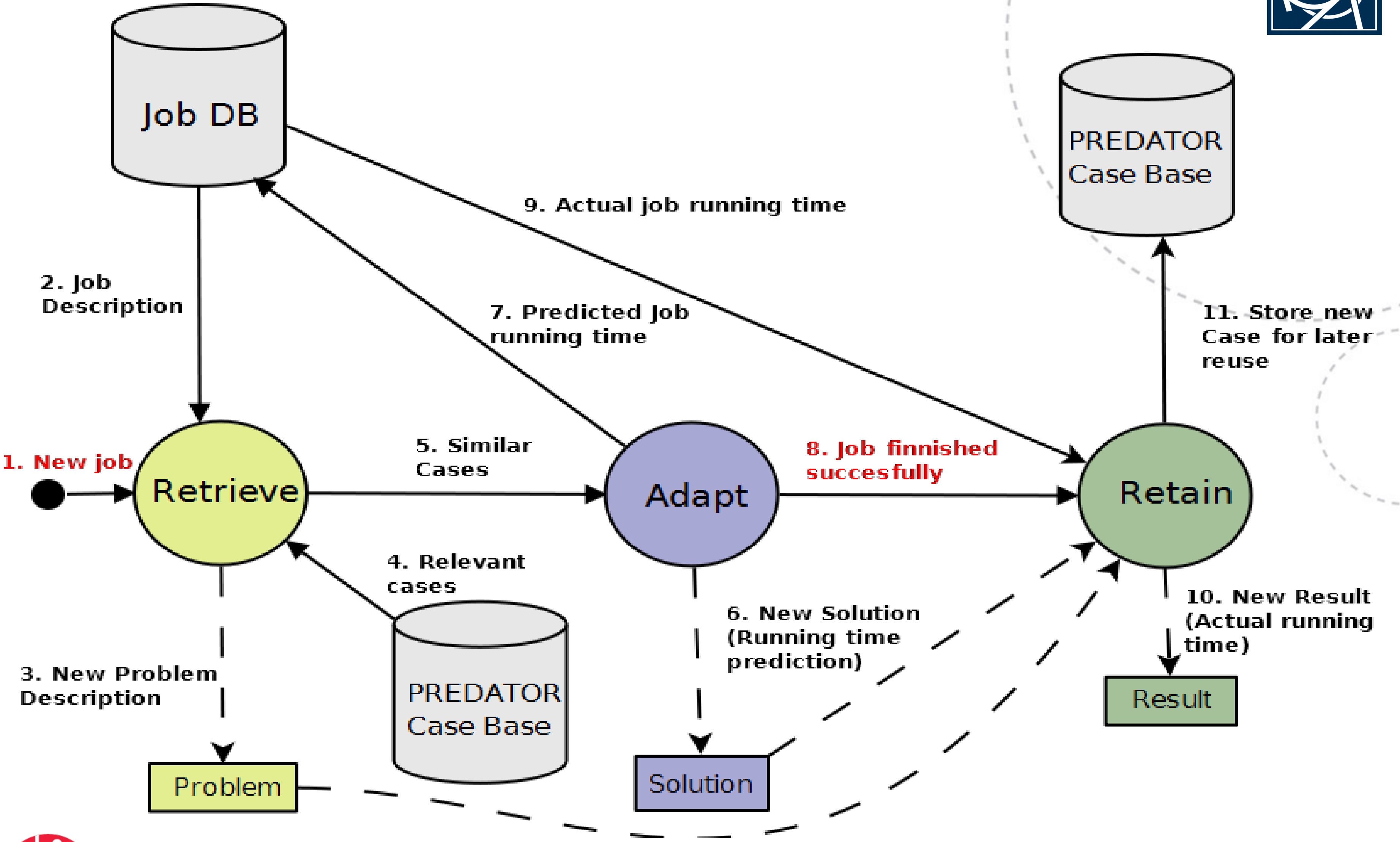
3. New Problem Description

PREDATOR Case Base

Problem

Solution

Result


Job DB
2. Job Description
9. Actual job running time
7. Predicted Job running time
PREDATOR Case Base
11. Store new Case for later reuse
1. New job
Retrieve
5. Similar Cases
Adapt
8. Job finnished succesfully
Retain
4. Relevant cases
PREDATOR Case Base
6. New Solution (Running time prediction)
10. New Result (Actual running time)
3. New Problem Description
Problem
Solution
Result

NTNU – Trondheim
Norwegian University of Science and Technology

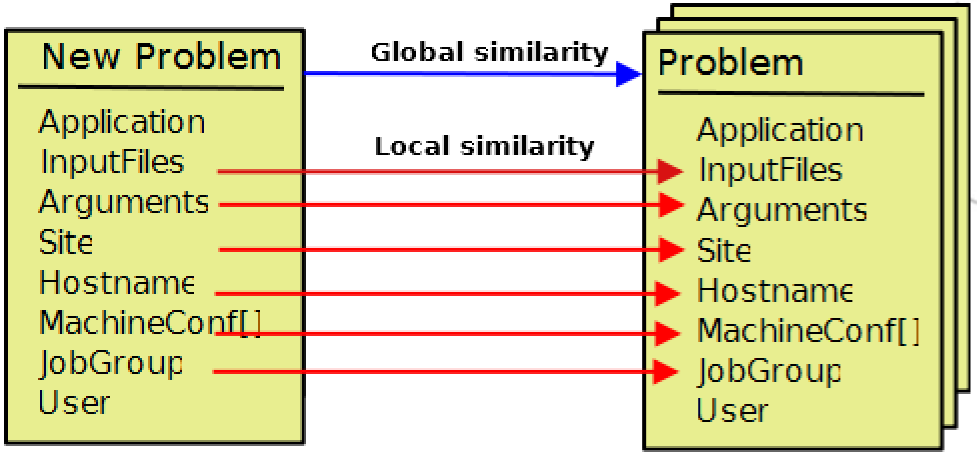
Retrieve

1. Retrieve job description from JobDB
2. Create new problem instance
3. Build relevance vector (E=[Site,Application,...])
4. Retrieve relevant cases from case base into memory
5. Apply similarity measurements
6. Select the most similar cases

Retrieve cont. Similarity measurements

Local similarity functions

- Numeric: $\text{sim}(a, b) = 1 - \frac{|a-b|}{range}$
- Taxonomy: $\text{sim}(a, b) = \frac{h(\text{common node}(a, b))}{\min(h(a), h(b))}$
- Symbolic: $\text{sim}(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{if } a \neq b \end{cases}$
- Multi-valued: $\text{sim}(a, b) = \frac{\text{card}(a) \cap \text{card}(b)}{\text{card}(a \cup b)}$



Global similarity function

$$\text{sim}(A, B) = \sum_{i=1}^{p} \omega_i \text{sim}_i(a_i, b_i)$$

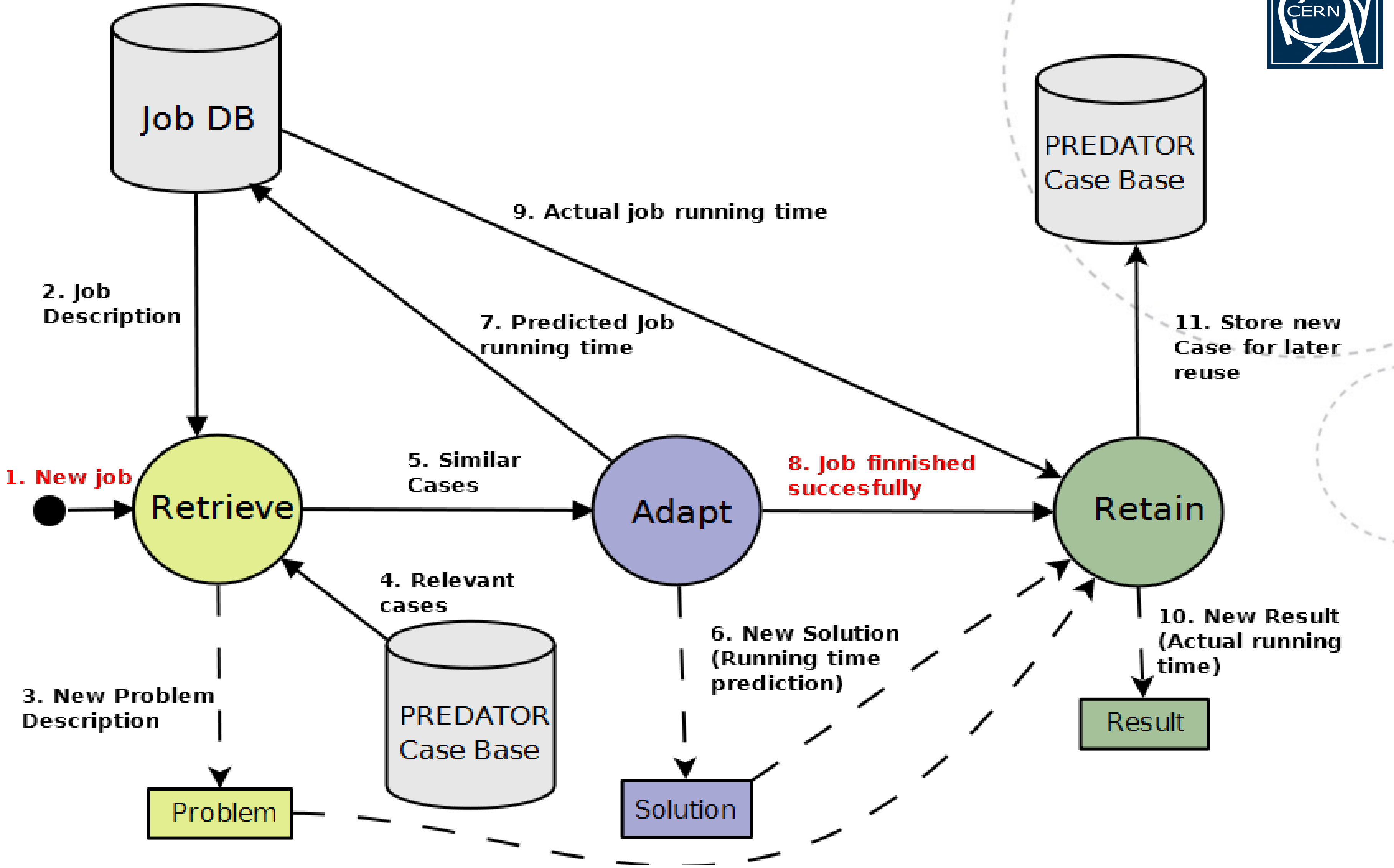
NTNU – Trondheim
Norwegian University of
Science and Technology

Adapt

- From the retrieved cases, average the real execution time of the result instances -> predicted running time
- Can also use different prediction methods based on which prediction method was most succesful in the past

Retain

- Upon job completion, running time is fetched from Job DB
- Result object is created and the prediction error is calculated
- The Problem, Solution, Result is stored in the Case Base


Job DB
PREDATOR Case Base
9. Actual job running time
2. Job Description
7. Predicted Job running time
11. Store new Case for later reuse
1. New job
Retrieve
5. Similar Cases
Adapt
8. Job finnished succesfully
Retain
4. Relevant cases
6. New Solution (Running time prediction)
10. New Result (Actual running time)
3. New Problem Description
PREDATOR Case Base
Result
Problem
Solution

NTNU – Trondheim
Norwegian University of Science and Technology

What challenges are we facing?

- The conditions on the worker node is not the same every time
- Heterogeneity within a GRID site
- Jobs are submitted in bunches → Bottlenecks

Other systems

- **PredCase**: CBR used to determine in which resource to run the job.
  - 9% Average prediction error
- **I. Foster et.al** sorts similar jobs into buckets, the predicted runtime is an average of the running time of the jobs in the bucket.

$$PredictionError = \frac{PredictedTime - MeasuredTime}{MeasuredTime} \times 100\%$$

NTNU – Trondheim
Norwegian University of
Science and Technology

Technology

- jCOLIBRI CBR framework (>= Java 1.6)
  - Need to define case structure
  - Simularity mecanism
  - Reuse\combine how the cases are combined to create the new solution
- Protege with myCBR to implement simularity funcion, can be exported and imported into jCOLIBRI

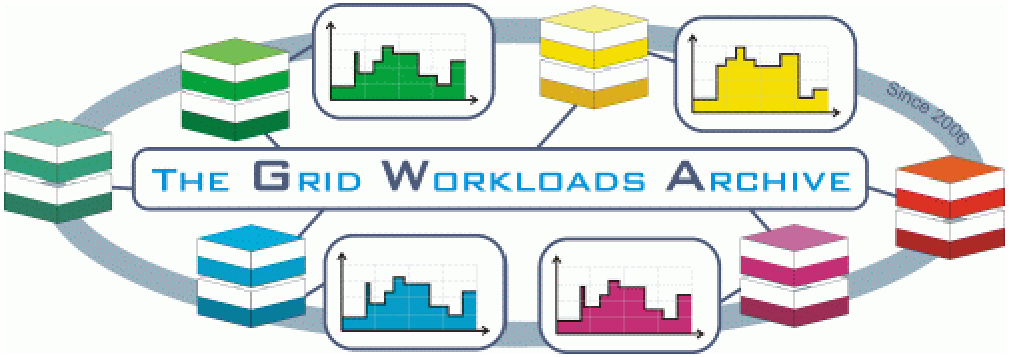
Project outlook

- Results available end of June
- Thesis by the end of July

Suggestions

- Accounting data → GRID workloads archieve
  - Standard: the Grid Workload Format
- GridSim → A Grid Simulation Toolkit

“The goal of the Grid Workloads Archive is to provide a virtual meeting place where practitioners and researchers can exchange grid workload traces. “



NTNU – Trondheim
Norwegian University of
Science and Technology

Summary

- PREDATOR: Predicting running time of production jobs in DIRAC
  - Uses CBR
- Technology: jCOLIBRI, Protege, myCBR
- GRID workload archieve, standard for grid workload traces

Conclusion

- PREDATOR must be implemented and tested
- Results should be available by end of june
- There are other systems that have shown this concept is possible

References

- W. Smith, I. Foster et.al. Predicting application run times using historical information.
  In Proceedings of the Workshop on Job Scheduling Strategies for Parallel Processing
- L. Nassif, J. Nogueira, M. Ahmed, et.al. Job completion prediction in grid using distributed case-based reasoning
- H. Li, D. Groep, J. Templon, and L. Wolters. Predicting job start times on clusters.
  In Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid