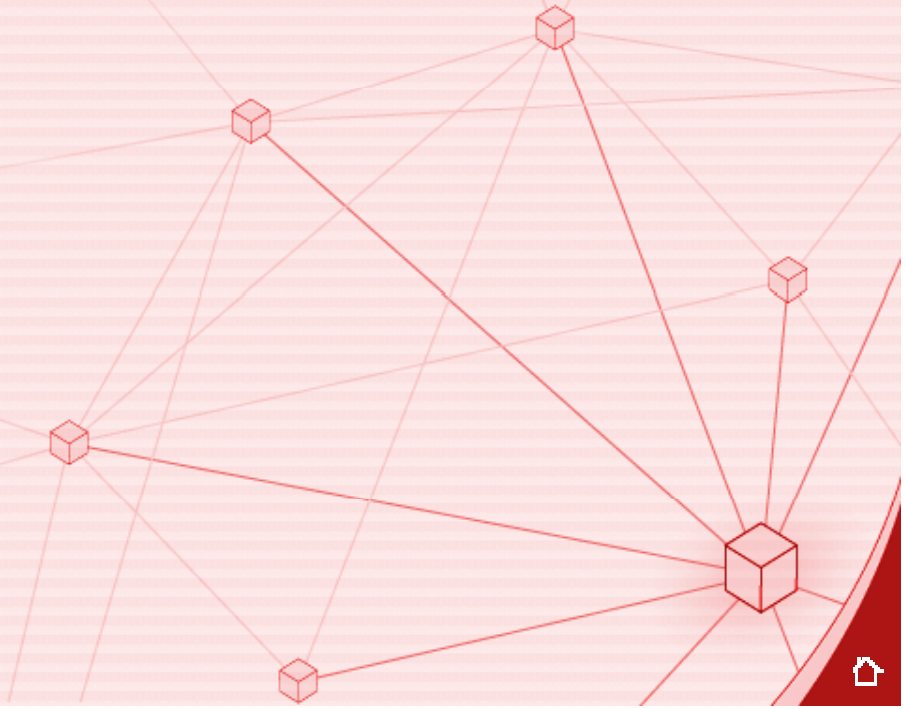
GRid-aware Optimal data Warehouse design (GROW)

Mr. Boro Jakimovski
University of Sts. Cyril and Methodius, Skopje, Macedonia

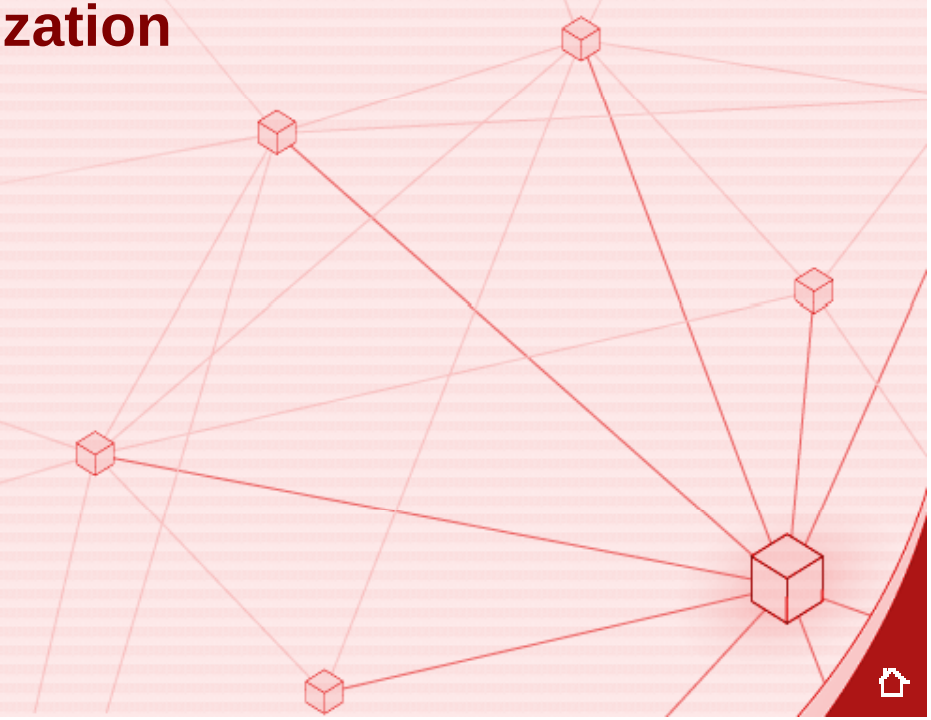
Acknowledgement

- The application and the framework was developed under the support of the SEEGRID-2 project and SEEGRID testbed



Motivation

- **Our previous research was focused on Optimizations of Data-warehouse design using Genetic Algorithms**
- **Genetic Algorithms are computationally intensive**
- **Very suitable for parallelization**
  - better performance
  - better solutions



Data-warehouse optimization 1

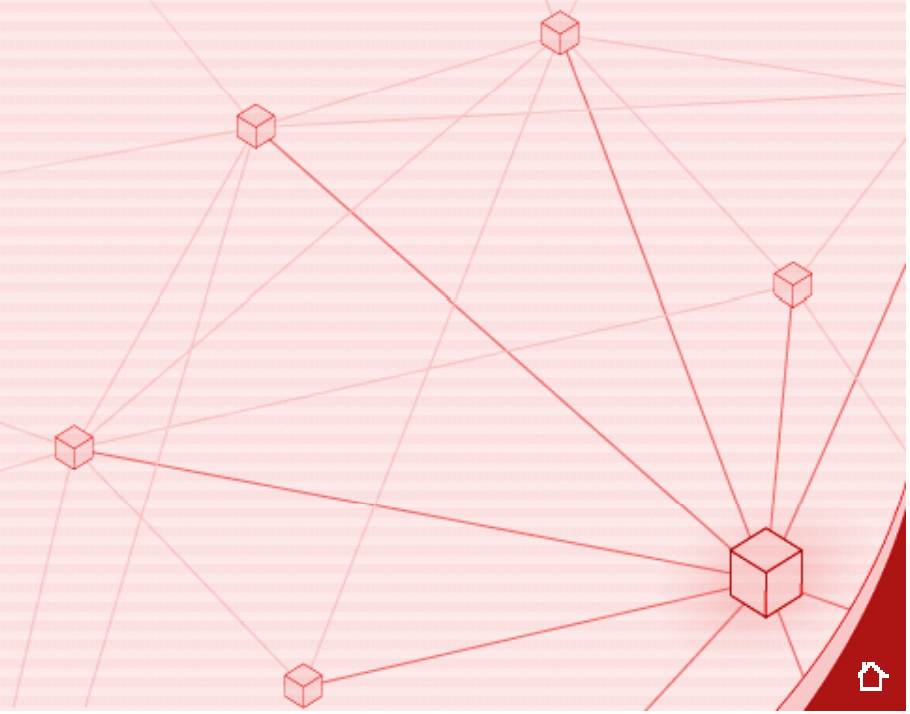
- **Data-warehouse optimization can be viewed from several aspects**
- **For simplicity we chose first to gridify simpler approach**
- **We try to solve VIS problem**
  - Optimal set of objects (Views and Indices)
  - Given database parameters and set of queries
- **Can significantly increase performaces on any large datawarehouse**

Data-warehouse optimization 2

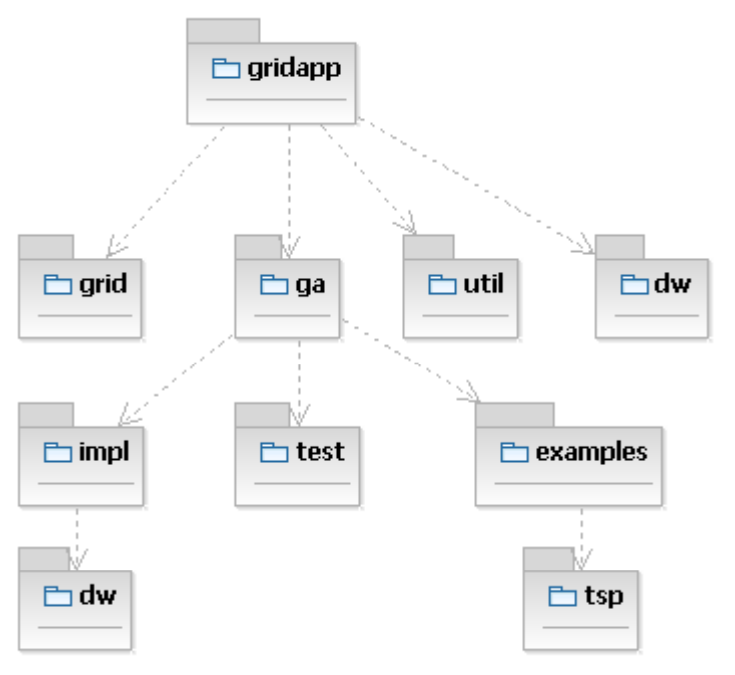
- **Our previous research was done using real database engines**
- **Optimization took long time**
- **Different approaches to increase the performance**
  - Greedy-genetic algorithms
  - Recessive genes

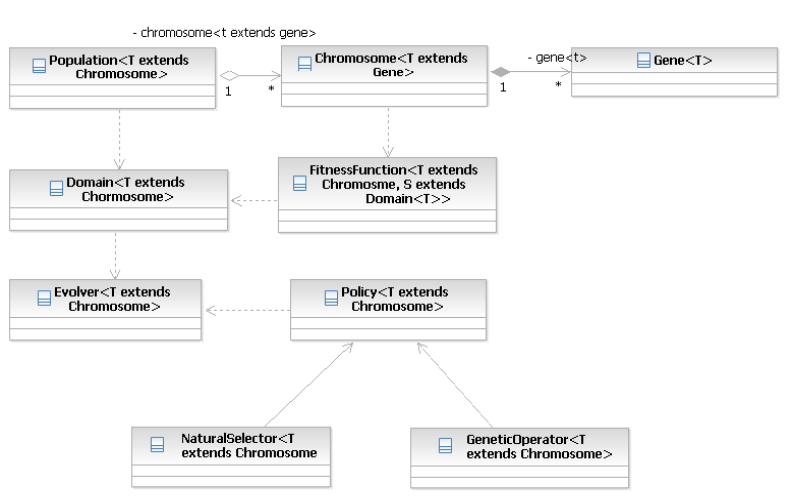
Grid Genetic Framework in Java

- **Development of Framework rather then specific solution**
- **Framework design conditions**
  - Robustness
  - Modularity
  - Extensibility
  - Flexibility
  - Adaptability
- **Java Generics**



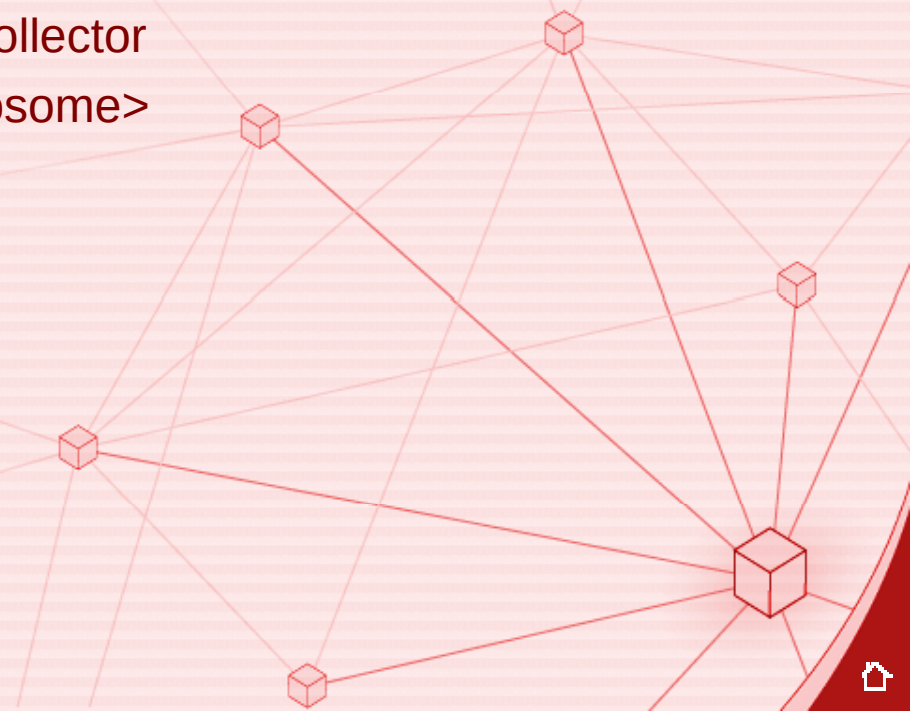
Framework components





Gridification implementation

- **Two aspects**
  - Workflow parallelization
  - Java grid usage
- **Workflow – 5 classes**
  - Creator, Breader, Migrator, Collector
  - JobGraph<T extends Chromosome>
- **Java grid usage – 1 class**
  - Simplified VOMS proxy
  - WMS
  - LB methods

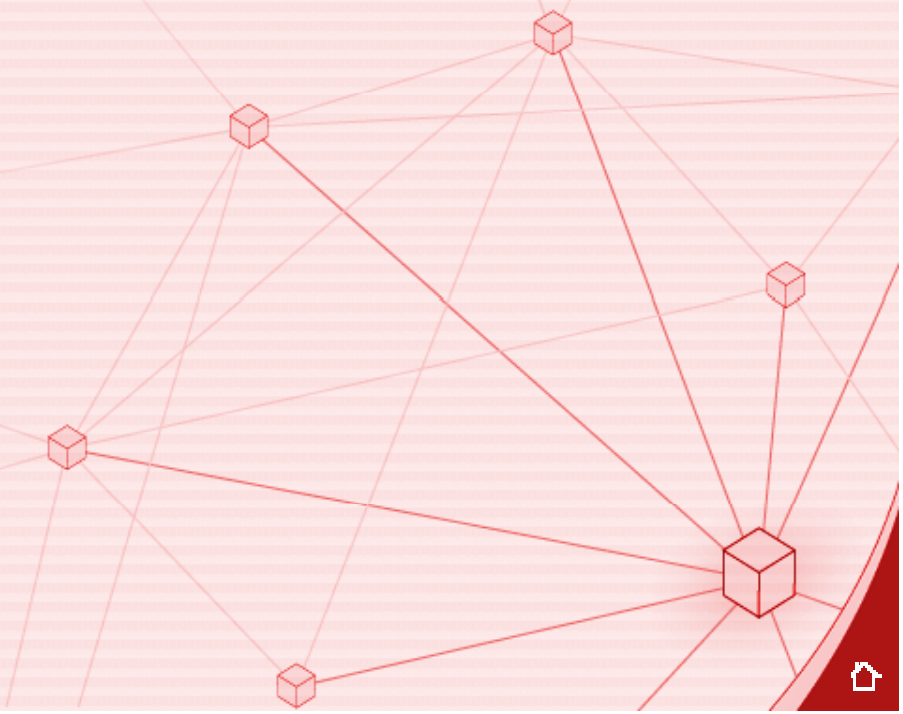


Optimization implementation

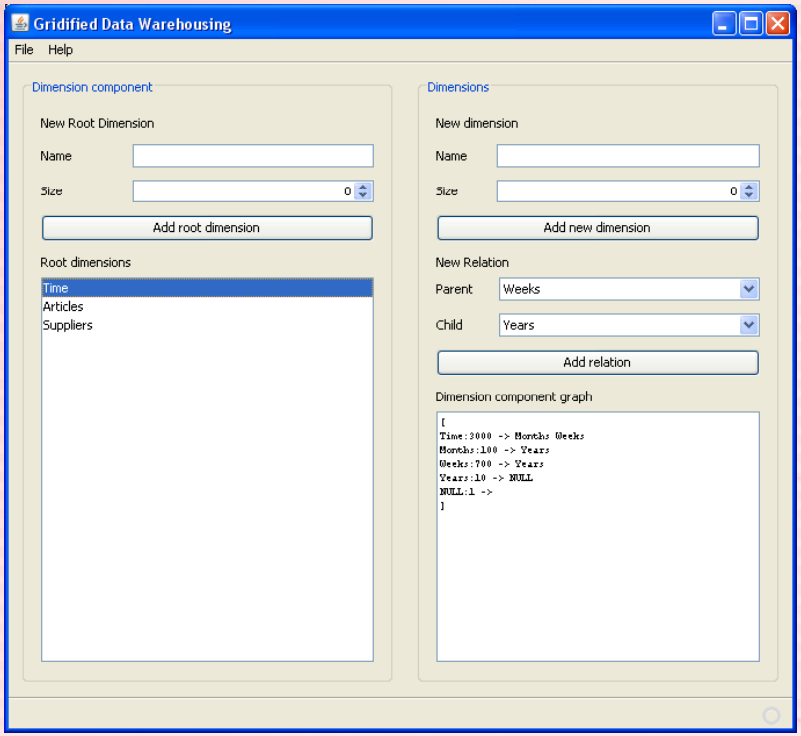
- The chromosomes are bit sequences
- Each bit representing weather particular view or index is materialized in the database
- The chromosomes are evaluated on a set of database queries, where for each query we estimate the time and memory usage for its execution
- The parameters for the GA optimization influence both per population GA execution and grid workflow execution

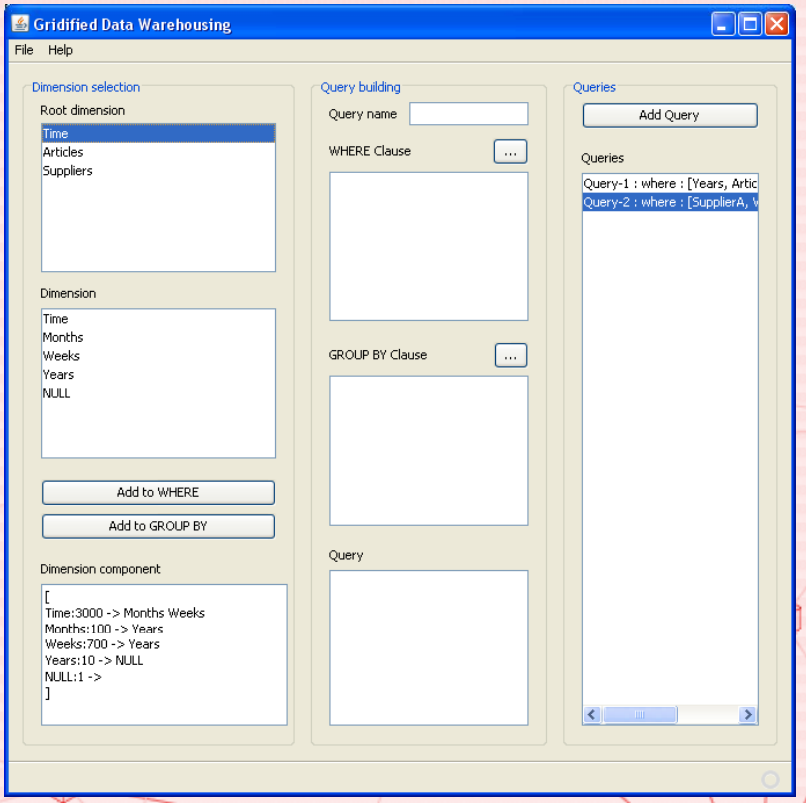
Optimization implementation

- **Extending the framework**
  - ViewGene extends Gene<Boolean>
  - ViewChromosome extends ArrayChromosome<ViewGene>
  - ViewFunction extends FitnessFunction<ViewChromosome, ViewDomain>
  - Domain<ViewChromosome>

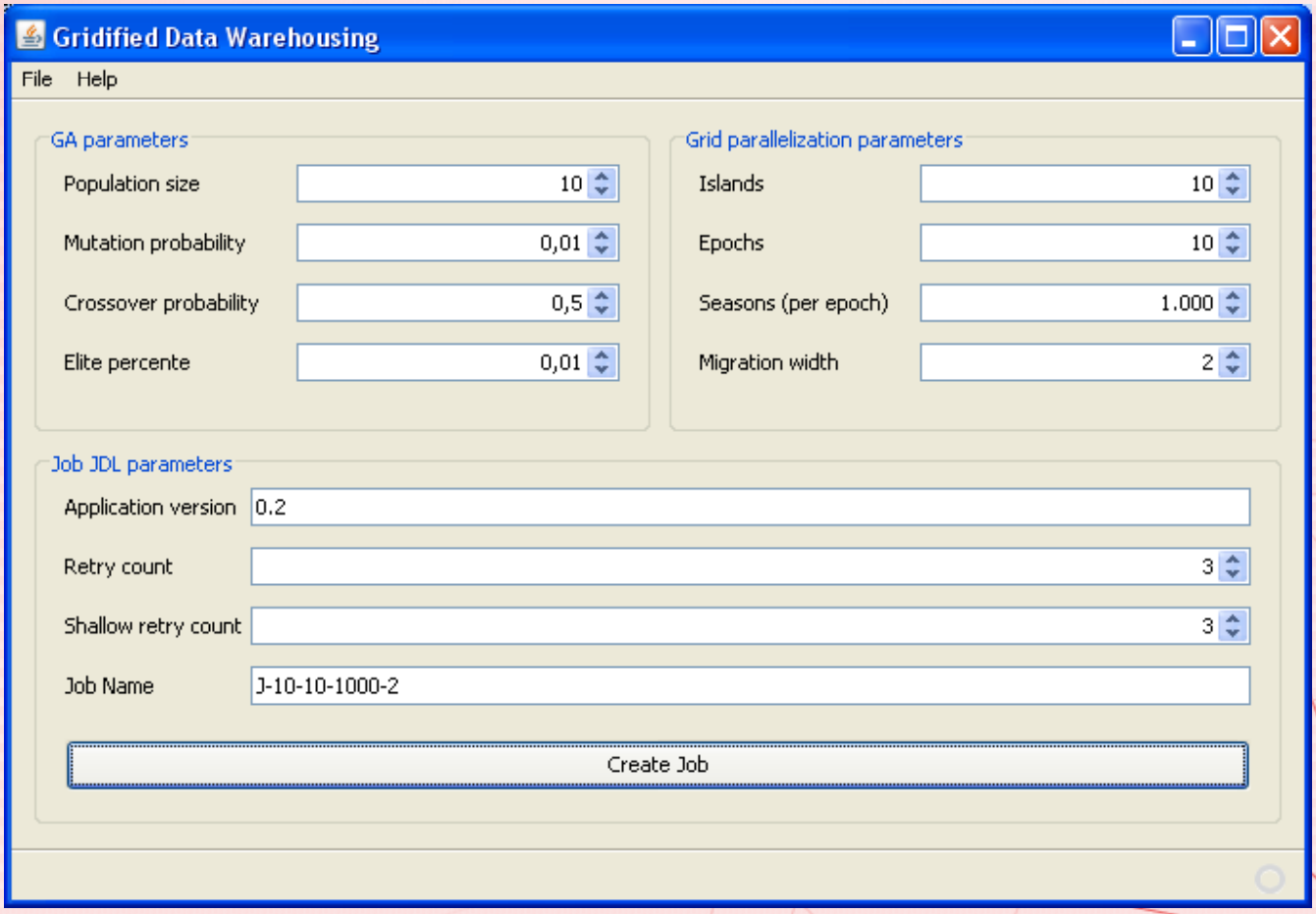


Application overview

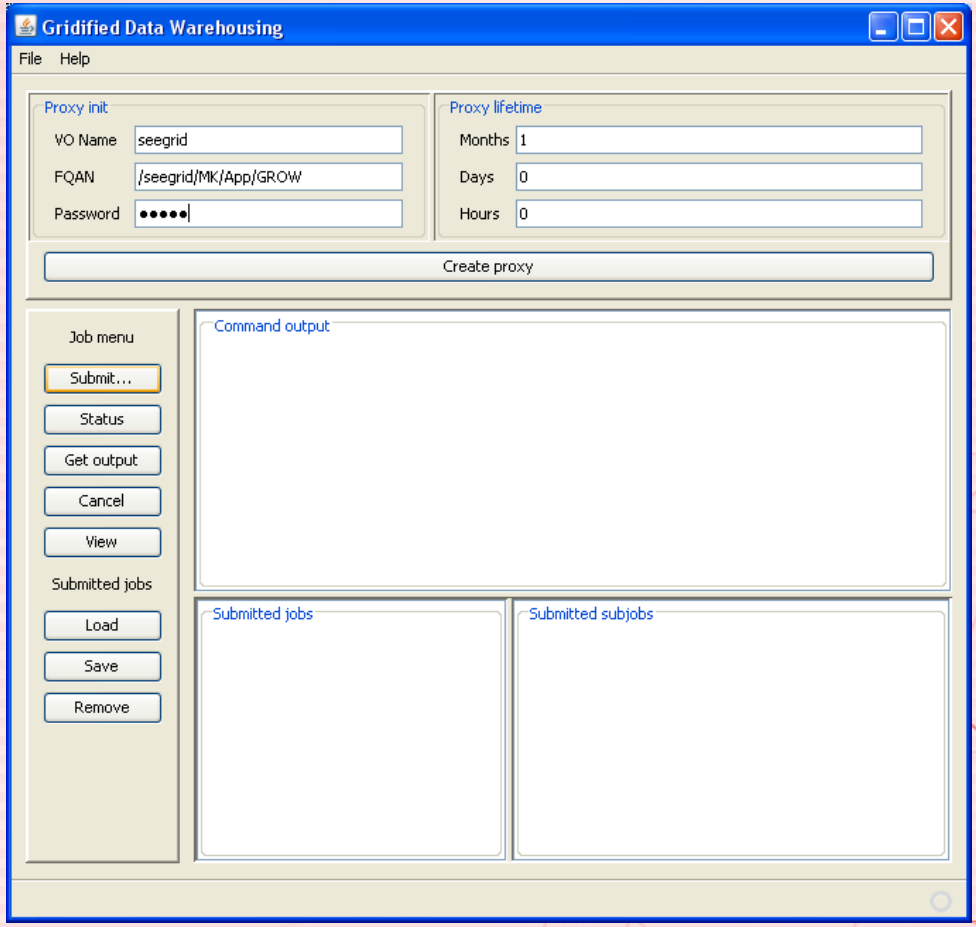




Application overview



Application overview



Conclusion and Future work

- **Easy implement and use Grid GA**
- **Extend the GridServices class**
  - Support for SRM
  - Support for BDII
  - Automatic CA Certificates retrieval
- **Detailed performance Analysis of the Application and Framework**
- **Implement more sophisticated evaluation functions and DW models**