



# GGrid-aware Optimal data Warehouse design (GROW)



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



# Acknowledgement

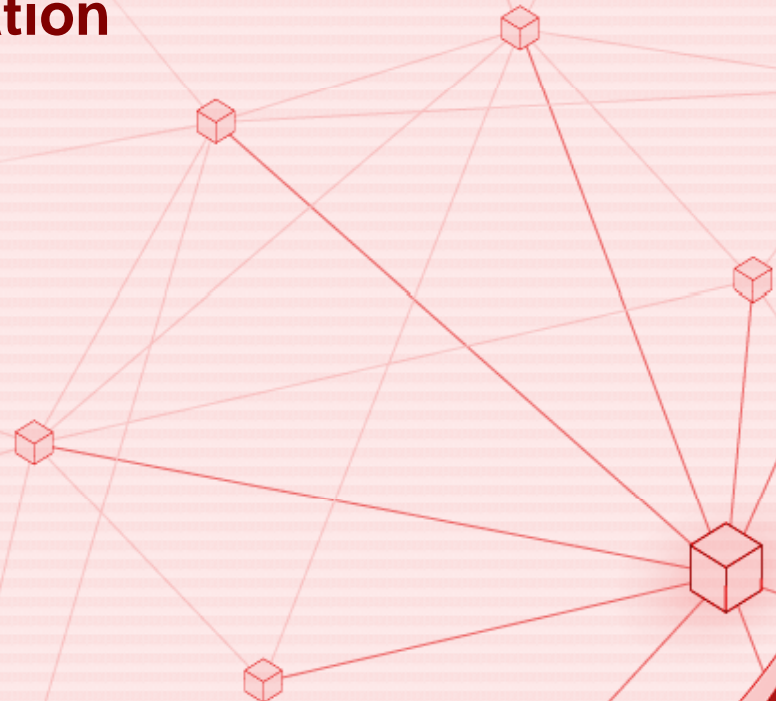
- **The application and the framework was developed under the support of the SEEGRID-2 project and SEEGRID testbed**





# 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 performances 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





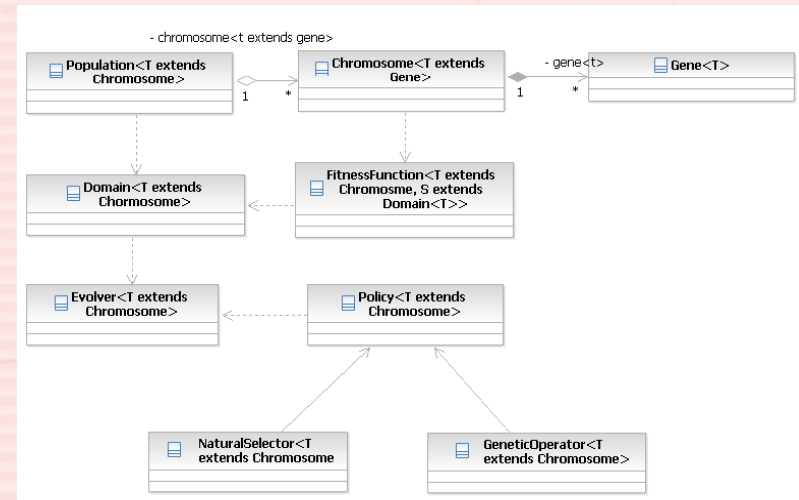
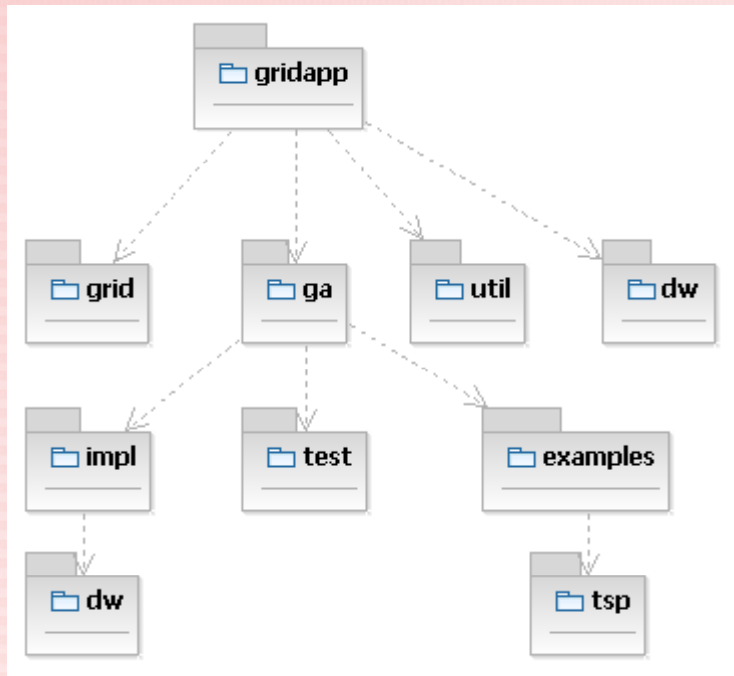
# Grid Genetic Framework in Java

- **Development of Framework rather than 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, Breeder, 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

**Gridified Data Warehousing**

File Help

**Dimension component**

New Root Dimension

Name:

Size:

Root dimensions

- Time
- Articles
- Suppliers

**Dimensions**

New dimension

Name:

Size:

New Relation

Parent:

Child:

Dimension component graph

```
[
Time:3000 -> Months Weeks
Months:100 -> Years
Weeks:700 -> Years
Years:10 -> NULL
NULL:1 ->
]
```

**Gridified Data Warehousing**

File Help

**Dimension selection**

Root dimension

- Time
- Articles
- Suppliers

Dimension

- Time
- Months
- Weeks
- Years
- NULL

Dimension component

```
[
Time:3000 -> Months Weeks
Months:100 -> Years
Weeks:700 -> Years
Years:10 -> NULL
NULL:1 ->
]
```

**Query building**

Query name:

WHERE Clause:

GROUP BY Clause:

Query:

**Queries**

Queries

- Query-1 : where : [Years, Artic
- Query-2 : where : [SupplierA, v





# Application overview

**Gridified Data Warehousing**

File Help

**GA parameters**

Population size	<input type="text" value="10"/>
Mutation probability	<input type="text" value="0,01"/>
Crossover probability	<input type="text" value="0,5"/>
Elite percente	<input type="text" value="0,01"/>

**Grid parallelization parameters**

Islands	<input type="text" value="10"/>
Epochs	<input type="text" value="10"/>
Seasons (per epoch)	<input type="text" value="1.000"/>
Migration width	<input type="text" value="2"/>

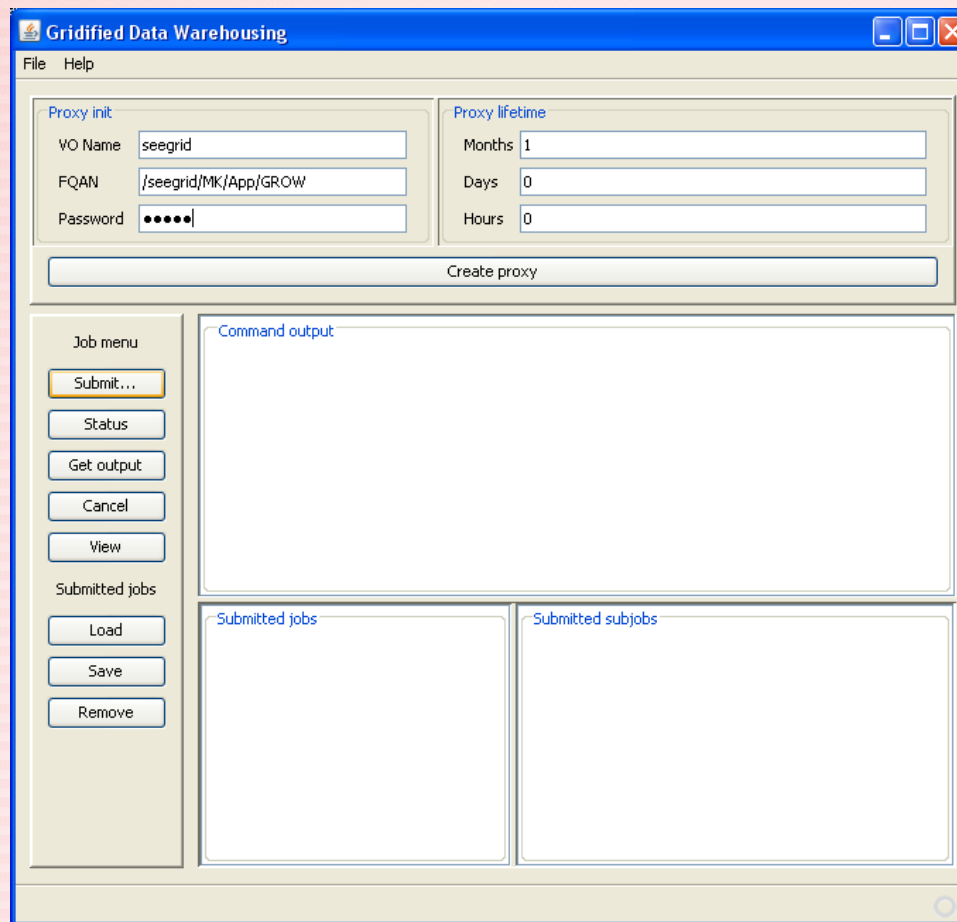
**Job JDL parameters**

Application version	<input type="text" value="0.2"/>
Retry count	<input type="text" value="3"/>
Shallow retry count	<input type="text" value="3"/>
Job Name	<input type="text" value="J-10-10-1000-2"/>





# 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**

