Big Data processing and analysis challenges in mega-science experiments, Dubna, Russia

CLAVIRE Highlights

A. Boukhanovsky, D. Nasonov, S. Kovalchuk, K. Knyazkov, T. Chorov, A. Larchenko, S. Mariyn, N. Butakov
Motivation

Common problems in HPC:

• Resources heterogeneity
• Packages (software) heterogeneity
• Resource and package high cohesion
• Wide user variation needs, lots of stakeholders
• Multi-level diversity of components interactions (users, packages, resources)
• Hard access and low-effective representation of complex experiments
• etc.
CLAVIRE = Cloud Applications VIRtual Environment

CLAVIRE is instrumental and technological environment that is designed to provide complete life cycle support of computation and data intensive systems (prototyping, developing, modernization) in various domain areas. 

*Main CLAVIRE advantages are:*

1. Genericity of technology which is attained with iPSE (Intelligent Problem Solving Environment) provides the possibility to develop and maintenance virtual environments for different purposes and areas;

2. Unified approach allows easily to integrate heterogeneous data sources and computation resources (supercomputers, grids, clouds);

3. Support of interactive application management that allows development of distributed real-time systems, systems of interactive visualization and virtual reality;

4. Big Data technology native support, based on own distributed storage.
CLAVIRE application area

- **Collaborative domain-specific portal** – that can be used for general scientific needs as well as in educational purposes by social communities

- **Competence center** – cloud supported by a group of high-qualified experts of specific scientific area

- **Enterprise private cloud** – used for business needs with second generation cloud abilities

- **Public cloud (domain-neutral)**, market-like cloud with concept of integrated personal cloud

- **Urgent computing infrastructure**: computation support of a situation center
Common 3-tiered CLAVIRE architecture
Workflow is **DAG**

Dependency types are:
- **data** (parameters, files),
- **control** (order),
- **communication** (network communication, interaction)

**EasyFlow** (DSL) – unified workflow description language
Ginger user interface. Design of WF

Workspace

Script Editor

Abstract WF visualization

```python
1. require AreaBusRoutes, AreaTimes, AreaBusRoute, Full, AreaDemand;
2. require FileRoadMapGraph, nA, nB;
3. step Demand runs simple_demand
4.   Format = "json",
5.   AgentCount = 1,
6.   Graph = FileRoadMapGraph
7. );
8. ~step ModelZone runs traffic (timeDist = true,
9.   Graph = FileRoadMapGraph,
10.  TimeStep = 0.3,
11.  Agents = Demand.Result.Outputs.AgentsInJson,
12.  SimulationEndAt = 300.0,
13.  Neighbours = nA
14. );
15. ~step ModelZone runs traffic (timeDist = true,
16.   Graph = FileRoadMapGraph,
17.  TimeStep = 0.3,
18.  Agents = Demand.Result.Outputs.AgentsInJson,
19.  Neighbours = nA,
20.  SimulationEndAt = 300.0,
21.  g01 <- M0.g01
22. );
23. step Collector runs traffic_collector after ModelZone, ModelZone
25.   //Open outputAreaInput
26. step gen_pairs runs transport_genpairs
27.   partNum = 1,
28.   routes = AreaBusRoutes
29. );
30. step find_routes runs transport_findpath
31.   (input_graph = Collector.Result.Outputs.EdgesStats,
32.   output_graph = Collector.Result.Outputs.EdgesStats,
33.   path_file = "path\files\path_file")
34. );
35. step demand requires simple_demand
36.   (input_demand = simple_demand)
37. )
```
Advanced Design of WF (1/2): procedural approach
Involving knowledge-based technologies

Using of ontology formalism for description of computational processes in domain areas

Rule-based WF’s design (in quantum chemistry e.g.)
Advanced Design of WFs (2/2): system (deductive) approach. Involving “virtual objects” concept

Virtual simulation object – composition of the models for evaluation parameters of some real-life object
☑ Available for description of complex system existing in some environment
☑ Interpretable (both from domain and simulation infrastructure point of view)
☑ Supports interactive simulation process

- Object parameters are available for every model in the object
- Object processing mode - forecast, static analysis, parameter optimization

Baltic sea : Sea
Object processing mode
- forecast
- static analysis
- parameter optimization

Object parameters
- are available for every model in the object

Data status
- correct (OK)
- Need to be (re-)defined (?)
- unavailable (X)

Models
- are transformed into parts of composite workflow for running simulation
- Can be tuned: Software and scenario selection, parameter setting
- Can be switched off by the user

- Data sets
- can be obtained from models internal or external for object
- additional setting mode: storage, external services, manual setting
- are checked for correctness and consistency

- Grid
- Bathymetry
- Aloft wind
- Near-water wind
- Sea waves
- Wave spectrum
Problem oriented interface (POI)

- Unified Package description allows to generate user friendly UI
- One-click workflow execution

Problem-oriented interfaces can be automatically generated with help of provided formal package description.
Package (software) embedding:

- **Retrieve necessary information** from the package.
- **Add package description on EasyPackage DSL in PackageBase** service, including:
  1. Common package information, input and output parameters.
  2. Configuration files templates if needed.
- **Add package installation form to autodeploy service** if possible.

```java
name "CNM_VIS"
display "Complex network modelling"
vendor "ITMO"
url "http://escience.ifmo.ru/
license "GPLv3"
description "CNM help to understand and analyse how i.e. rumors are being spread."
inputs {
  public param {
    name "in_format"
    display "Формат входных данных"
    type enum ["short","full","alt"]
    default "short"
    required
  }
  public param {
    name "coefName"
    display "Коэф. Пуассона","Степ. показатель"
    type string
    default "Коэф. Пуассона"
    required
  }
  public param {
    name "coefValue"
    display "Значение коэффициента"
    type float
    default "0"
    validator {
      val,ctx| val <= 20
    }
  }
}```
Computational resource embedding.

- **Add description** with PRManager or direct with ResourceBaseService that provides all functionality to manage resources
- **Install client** if needed (in example, REX-Windows client)
- **Physical package installation on resource**, if package is not uploaded to PackageInstallerService for automatically deployment

```json
<resourceType> = {
  "common resource parameters, including controller type"
  "NodeDefaults": {
    "common node default parameters, including core count, hardware parameters"
    "Packages": [{
      "common package parameters, including functions “copyOnStart” and “cleanDirectory”
    }]
    "Nodes": [ {
      "NodeName": "<NodeName>", node name
      "NodeAddress": "< NodeAddress >", - node address
      "Services": {"ExecutionUrl": "<ClientWCF>"} – execution path for WS
    }]
```
CLAVIRE Big Data Infrastructure

- Distributed data storage: core + agents
- Data replication and version control
- MapReduce model for distributed data (files) processing
- Java implementation of data processor
- Parallel efficiency up to 97%

![Diagram of CLAVIRE ITMO UNIVERSITY CLAVIRE UI](Image)
Optimization in CLAVIRE

**WF Scheduling optimization:**
- **Hybrid algorithms** combine heuristic and meta-heuristic approaches to get best advantages from both sides;
- **Window-based algorithms** take into account time windows to get maximized utilization with deadline constraints.

**Data placement optimization:**
- **Static algorithm** optimizes data placement according to data usage during computation requests;
- **Dynamic algorithm** optimizes data placement in time when environment and computation request are changing.
CA: flash-mob evacuation modeling

- **Generate** complex network;
- **Exclude** random nodes;
- **Modeling** of information spreading in the group of networks;
- **Aggregation** and statistics computation;
- **Evacuation** modeling and visualization;
CA: Flood warning system

- **Data elements** collect data from sources
- **Data uncertainty mask** is generated and applied to wind velocity; ensemble **modeling** of wind-generated waves and water level in Baltic Sea area;
- **First Results** aggregation and visualization
- **Decision support on close plan process is executed.**
- **Final Result** selection

---

Deadline driven planning with task preemption.

---

CLAVIRE Highlights
Education principles based on CLAVIRE

CLAVIRE Highlights

1. VSOs, SWFs, VLL’s
2. Text Resources
3. Assessment’s Packages
4. Commands and dataflows
5. Computational resources and installed application packages
6. Educational Communities

- Teachers and students
- Scientific Communities
  - Scientific Teams
  - Scientific Papers and Documents
  - Programming and Experimental Documentations

- Educational Communities
  - Teachers and students

- CLAVIRE
  - Interfaces
    - VSO, SWF, application packages descriptions
  - Component “Site”
CLAVIRE with Panda integration concept

CLAVIRE can be useful for Panda, for the following reasons:
• CLAVIRE manages all CA as workflows with provided capabilities for urgent execution.
• CLAVIRE implements IWF technology.
• CLAVIRE encapsulates principles of composite applications, that allow Panda to extend its functionality on different types of services.
• CLAVIRE can operate with wide range of different resources.

Panda can be useful for CLAVIRE, for the following reasons:
• Panda can increase efficiency of used Grid resources
• Panda can optimize storage organization
• Panda provide pilot concept
Conclusion and future plans

CLAVIRE as a multipurpose platform has great potential for integration and cooperation with other computation environments, providing rich and clear interface functionality for fast composite applications development and efficiently its usage.

Besides different platform modules optimization, BigData technologies within hybrid idea of binding “code-to-data” and “data-to-code” concepts is one of most important current research directions that is investigating by our team.
CLAVIRE VIDEO DEMOSTRATION
Thank you for your attention!

Big Data processing and analysis challenges in mega-science experiments, Dubna, Russia
Security model

- **Basic ACL** security model on data and service access;
- **EasyPackage** limits access to needed parameters only with prohibited direct access to resources and packages.
## Workflow Management Platforms

| Supported resources | Taverna University of Manchester, UK | Kepler Community | LONI Pipeline University of California, USA | WS-VLAM University of Amsterdam, Netherlands | Pegasus University of Southern California, USA | CLAVIRE ITMO University, Russia |
|---------------------|-------------------------------------|-----------------|-------------------------------------------|---------------------------------------------|---------------------------------------------|---------------------------------
| Windows support     | +                                   | +               | −                                         | +                                           | −                                           | +                               |
| Abstract WF         | −                                   | −               | −                                         | +                                           | +                                           | +                               |
| Interactive WF      | −                                   | −               | −                                         | +                                           | −                                           | +                               |
| Exception handling  | +                                   | +               | +                                         | −                                           | +                                           | −                               |
| Tools for history analysis | + | − | + | − | + | + |