Implementation on Parallel Architectures

Application: Interface Relaxation as a Service (IRaaS)

S. Likothanassis, P. Alefragis and A. Korfiati

IRaaS Application

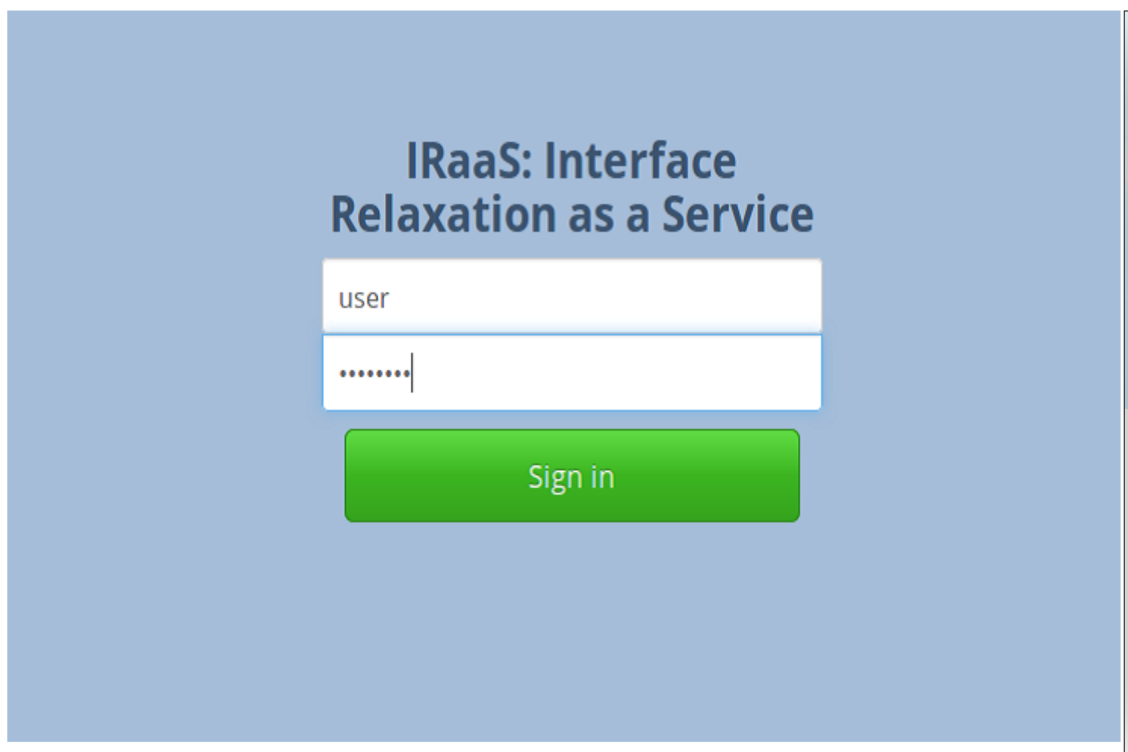
The proposed cloud application is a **solution environment** for **multiphysics/multidomain problems**

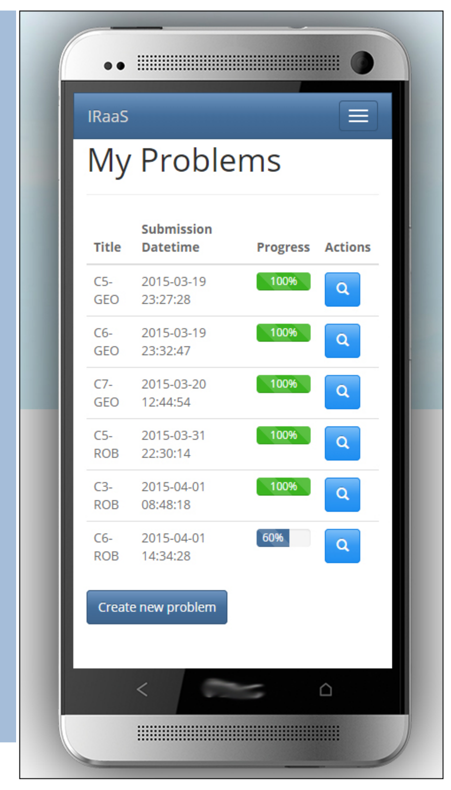
- implementing the **interface relaxation methodology**, and
- utilizing **cloud technologies** that manage pre-existing hardware, network, operating system and applications.

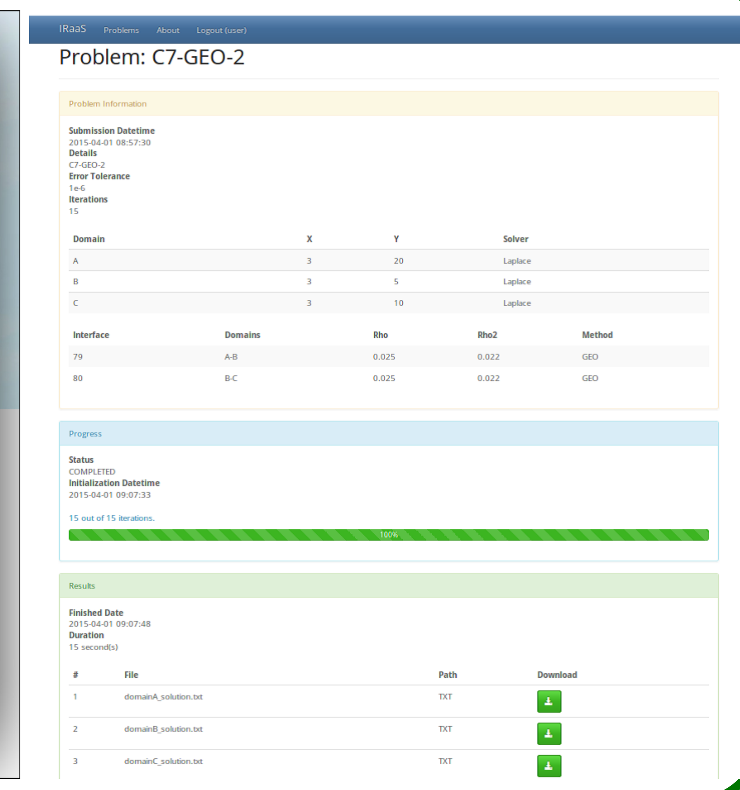
The user sets the problem's parameters, chooses the interface relaxation method (**GEO/ROB**) that fits better to the specific problem and finally gets the problem solution from any place and any device.

The application **dynamically allocates** the **minimum possible resources** automatically **in the background** without the user's interference [1], [2].

An asynchronous implementation of GEO has also been implemented [3].







System architecture's components

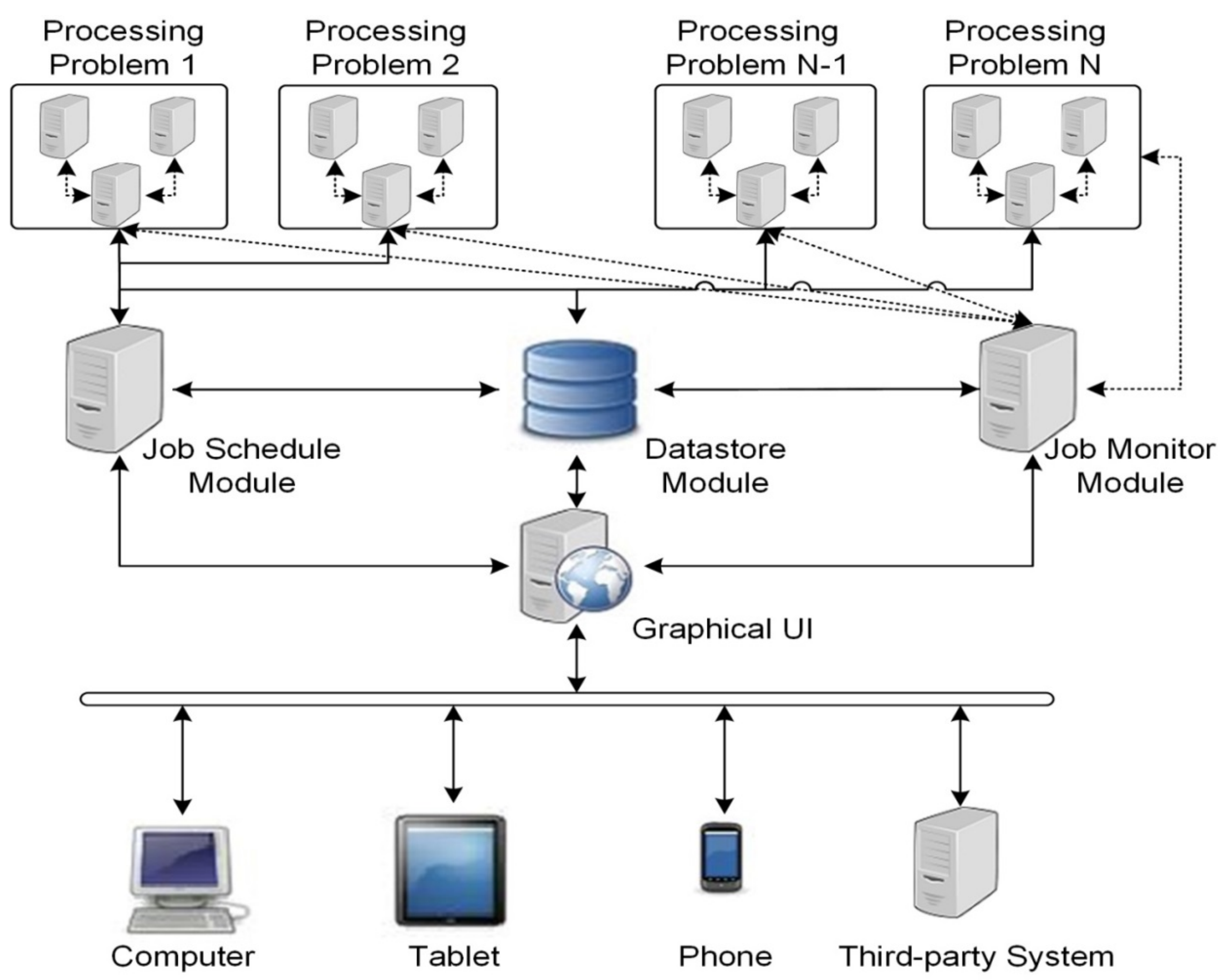


Figure 1: IRaaS architecture

The proposed application is based on the management of the allocation of the cloud resources in order to create groups of virtual machines for parallel processing. Furthermore, it provides the appropriate interfaces to end-users and systems for accessing the platform.

The major functional components of the architecture are the following:

- The **Graphical GUI** where each user has the ability to register for a new account or log in to an existing one. When logged in, users can define new problems by inserting the appropriate input and choosing the desired IR method for the problem solution. During the solution process, they can review their problems' solution status and details. Furthermore, third party systems can access the services provided by the application through a simple HTTP API that is loosely based in the exchange of JSON messages.
- The **Job Schedule Module** which is responsible to orchestrate problem execution on the cloud infrastructure. Its duties consist of a) the resources allocation needed by the Virtual Solver Nodes for the problem execution, b) the deployment of the eligible VMs along with their information, c) the initialization of the problems and d) the deletion of the VMs after the execution is finished.
- The **Job Monitor Module** is an Advanced Message Queue Server (AMQS) that handles the communication between the entities of the system. AMQS is based on the Advanced Message Queuing Protocol (AMQP) which connects systems and manages the information and messages exchange between them. A more extensive interpretation is that with AMQP programs and systems can produce and send messages, while other programs and systems can receive them and process them. In the present work RabbitMQ has been used. RabbitMQ is a messaging broker based on the AMQP and offering a common platform to send and receive messages while these messages stay safe until they reach their final destination.
- The **DataStore Module** can be accessed from all the system components and stores the input data, the results and the intermediate data of the users' problems.
- The **Virtual Solving Nodes** are created by the Job Schedule Module in order to start the problem execution. During the execution, they inform the Job Monitor Module about the state of the problem and once they finish the execution, they send their final results back to the Job Monitor Module. After job completion they are deleted.

Figure 1 depicts an overview of the proposed system's architecture with all the system's modules and their interactions, where dotted lines present communication between modules. The AMQS infrastructure is utilized for the communication between the various modules. AMQS-based communication is used in the following interaction scenarios: a) inside Virtual Solving Nodes for the interface values and b) between the Virtual Solving Nodes and the Job Monitor Module for progress monitoring as depicted in dotted lines.

Experiments and Results

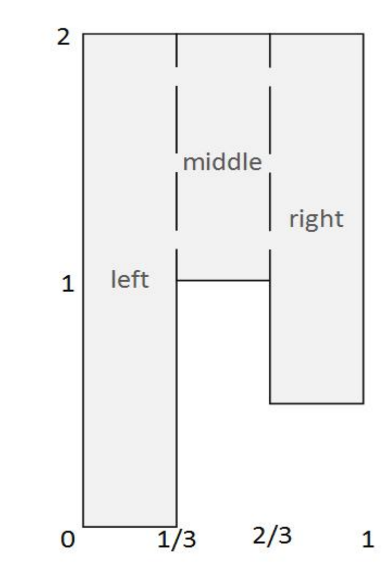


Figure 2: Examined problem

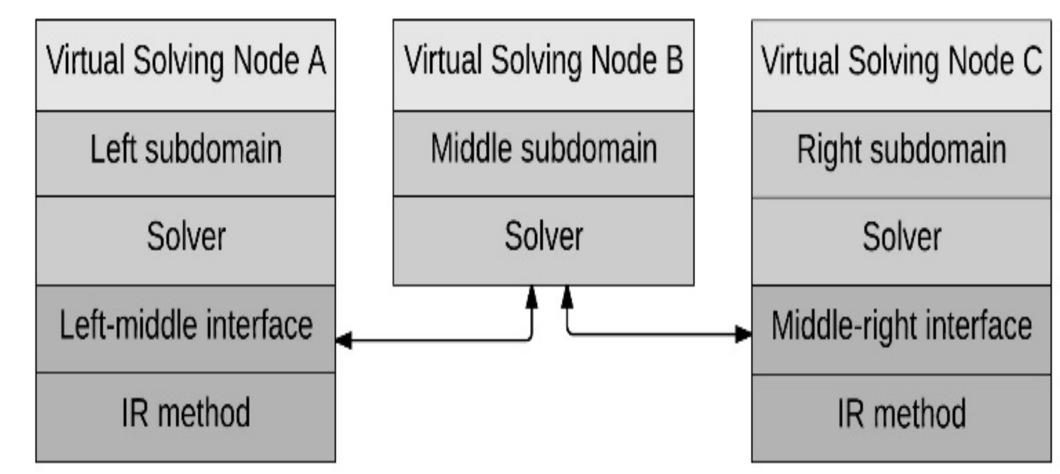


Figure 3: IR distributed solution

TABLE I
CONSIDERED CASES

Case	h	Left	Middle	Right
C1	0.1	4x21	4x6	4x11
C2	0.05	8x41	8x11	8x21
C3	0.025	14x81	14x21	14x41
C4	0.0125	28x161	28x41	28x81
C5	0.00625	55x321	55x81	55x161
C6	0.003125	108x641	108x161	108x321
C7	0.0015625	214x1281	214x321	214x641

TABLE II
RESOURCES

Resources	Left domain			Middle domain			Right domain		
	cores	GHz	GB	cores	GHz	GB	cores	GHz	GB
R1	1	2	1	1	2	1	1	2	1
R2	1	2	2	1	2	2	1	2	2
R3	1	2	4	1	2	1	1	2	2
R4	1	2	4	1	1.5	2	1	2	2

TABLE III
EXECUTION TIMES

Resources/Case	R1	R2	R3	R4
C1	15.041	15.082	15.016	15.042
C2	15.411	15.457	15.320	15.345
C3	15.724	15.894	15.736	15.735
C4	19.169	21.442	19.234	19.334
C5	28.548	37.271	28.669	28.470
C6	71.416	81.252	70.972	71.499
C7	500.146	307.241	297.122	298.870

```
if subdomain size <= 75000:
        virtual solving node RAM = 1GB
        virtual solving node processor = 1core, 2GHz
else if  75000 < subdomain size <= 150000:
        virtual solving node RAM = 2GB
        virtual solving node processor = 1core, 2GHz
else if subdomain size > 150000:
        virtual solving node RAM = 4GB
        virtual solving node processor = 1core, 2GHz
```

Problem	Execution Times	
	ROB	GEO
P1	15.348	15.041
P2	15.373	15.411
P3	16.022	15.724
P4	19.223	19.169
P5	27.313	28.548
P6	65.386	71.416
P7	265.415	297.122

Comparison Analysis

Problem	Virtual Implementation ROB	Proposed Cloud Implementation ROB	Virtual Implementation GEO	Proposed Cloud Implementation GEO
P1	16.004	15.348	15.495	15.041
P2	16.281	15.373	16.068	15.411
P3	17.392	16.022	18.140	15.724
P4	23.634	19.223	25.595	19.169
P5	45.456	27.313	51.808	28.548
P6	132.179	65.386	166.633	71.416
P7	1099.05	265.415	735.731	297.122

Conclusions and Future Work

We have presented a cloud-driven application for the solution of multidomain/multiphysics problems based on the Interface Relaxation methodology. In this way a large problem is split in smaller sub-problems, which are solved independently exchanging information about their solution iteratively until the initial large problem is solved. Through the proposed application the users can define complex multiphysics problems, select the appropriate PDE solvers for the smaller sub-problems and IR method (ROB or GEO) for the interfaces and get the computed solution of the global problem.

The resources' allocation is dynamic and is based on the computational needs of each problem. The main benefits when using the proposed application are that it allocates the minimum possible resources, solves the problem in a close to minimum execution time, and the resources allocation is performed automatically in the background without the user's interference.

In the near future, we plan to implement a larger number of PDE solvers and IR methods, as well as to provide support for more complex geometries. These modifications will enable us experiment with model problems representing the dissemination of primary brain tumors (gliomas) and the saltwater intrusion into freshwater aquifers due to overpumping.

Related publications

1. Aigli Korfiati, Niki Sfika, Konstantis Daloukas, Christos Alexakos, Panagiota Tsompanopoulou and Spiros Likothanassis, IRaaS: A Cloud Implementation of an Interface Relaxation Method for the Solution of PDEs, Proceedings of the World Congress on Engineering 2015 Vol I, 541-546
2. Niki Sfika, Aigli Korfiati, Christos Alexakos, Spiros Likothanassis, Konstantis Daloukas, Panagiota Tsompanopoulou, Dynamic Cloud Resources Allocation on Multidomain/Multiphysics Problems, Proceedings of the 3rd International Conference on Future Internet of Things and Cloud FiCloud 2015, Edited by Irfan Awan, Muhammad Younas, and Massimo Mecella, Technically Co-Sponsored by IEEE Technical Committee on the Internet (IEEE-CS TCI), 31 - 37, DOI: 10.1109/FiCloud.2015.59
3. Aigli Korfiati, Konstantis Daloukas, Panayiotis Alefragis, Panagiota Tsompanopoulou, Spiros Likothanassis, An Asynchronous Interface Relaxation Method for Multi-domain/Multi-Physics Problems, 13th International Conference on Numerical Analysis Applications and Methods (ICNAAM 2015)