

Grid Services for Multi-objective Optimisation

G. Goteng, A. Tiwari, R. Roy

Manufacturing Department, Cranfield University, Cranfield, Building 50, Bedford, MK43 0AL, UK

{g.l.goteng, a.tiwari, r.roy}@cranfield.ac.uk

Abstract

The emerging grid technology is defined as an infrastructure for secure and coordinated large-scale resource sharing. In this paper, we describe the architecture and grid services of DECGrid. DECGrid enables distributed design experts to collaborate and share resources during design optimisation. Mathematical models are built using services by experts. These models are then directly linked to NSGA-II optimisation algorithm service and allow design experts to enter design parameters of their choice. A real-life case study-welded beam problem was used to validate the prototype. The results obtained showed a wider spread in the solution space compared to the results in literature.

Keywords:

Grid services, Multi-objective optimisation, design optimisation, Mathematical model

1 INTRODUCTION

Competition and the desire to retain as well as to attract new customers are crucial reasons for optimisation of product design for companies to remain relevant in today's global business. The quality of a product is a reflection of the skills put into the design phase by bringing together different talents of design experts. Collaborative design goes beyond producing products that meet customers' requirements in terms of performance, quality and cost but should also have novel and good-looking features externally [1]. [1] observed that for this to happen, engineers and designers need to have access to tools that support design methods. The DECGrid (Decision Engineering Centre Grid) aims at providing these tools and resources that support multi-objective design optimisation (MDO) as grid services for distributed designers to collaborate and work together. DECGrid consists of mathematical model building service; design parameters input service and NSGA-II optimisation service. These services are built within the Globus Toolkit middleware and run in Linux environment. Grid technology has the capability that allows designers to recreate data, integrate distributed resources and reuse them [2]. DECGrid uses this feature to enable designers to manage workflows of design processes as well as reuse or recreate them.

2 RELATED WORK

Ever since the Global Grid Forum (GGF) met in 1993 to demonstrate the computational synergy obtained by linking 17 supercomputing centres, many grid projects have emerged thereafter to solve computationally and data intensive jobs. These projects range from middleware [3], schedulers [4] and problem solving environments (PSE) [5]. This paper is on PSE for solving MDO problems. Some similar projects that have developed PSEs for MDO applications are Geodise (Grid-Enabled Optimisation Design Search Environment), DAME (Distributed Aircraft Maintenance Environment) and FIPER (Federated Intelligent Product Environment). Geodise provides computational resources as services for optimisation engineers to access and use through a user interface [6]. Geodise consists of toolboxes for

computation, optimisation, visualisation and knowledge repository which the users are presented with through a grid portal. A wizard-like process for carrying out optimisation guides design engineers. The DAME project develops a grid-based fault diagnosis and prognosis for aircraft maintenance on the fly [7]. This is done by capturing data and information on various parts of the aircraft and providing this information to maintenance engineers as grid services when there is a deviation from normal behaviour. This helps engineers to take prompt decision. The aim of FIPER is to provide an intelligent system that leverages the emerging web technologies in which engineering tools such as CAD (Computer Aided Design), CAE (Computer Aided Engineering), PDM (Product Data Management) and optimisation algorithms act as distributed service providers as well as service requestors communicating through intelligent context models for concurrent engineering design optimisation [8]. The difference between these PSEs and DECGrid is that while these 3 provide service specifications at the programmers' level, DECGrid provides specifications document for both programmers as well as end users of the services. This ensures compliance to service level agreements between service providers and requestors.

3 FRAMEWORK AND ARCHITECTURE OF DECGRID

DECGrid runs in Linux platform built within Globus middleware. It consists of 8 nodes which perform different complementary design and optimisation functions and use Condor for scheduling. MDO being a multidisciplinary oriented field calls for the use of service oriented grid platform that will provide resources to users as services for easy data and information sharing. For example FIPER is used in the preliminary design optimisation of complex engineering product such as the gas turbine which requires the interaction between different professionals in aerodynamics, heat transfer, structural analysis, finite element analysis and computational fluid dynamics to satisfy the scalability required for these interactions [8]. DECGrid is proposed to extend this sort of scalability so that optimisation resources for MDO applications can be described, published, subscribed and used as well as enable mathematical model building. This

allows distributed users to share limited computational resources for optimisation.

3.1 Service Framework

The framework is the process of service specifications that provide design engineers the environment to specify design requirements. This is a document that shows the functionalities of the design resources and quality of service. This document represents the bond among collaborating parties and ensures the delivery and review of services. Figure 1 is the class diagram of the framework. The initial interaction is between the service provider and requestor based on service level agreement [9]. After which the service provider uses the functionality of Globus Toolkit to register design tools and resources in WebMDS (Web Monitoring and Discovery Service). All resources are aggregated in the design Aggregator Source. A search strategy interface is provided for optimisation engineers to perform search using deterministic or stochastic search algorithms. After subscription, requestors execute DesignService which has interfaces for step by step process for building a mathematical model for design optimisation in a particular field. The first interface has the main domain. This allows the design engineer to select the domain (FEA, CFD, etc) that the mathematical model is needed. The criteria, design parameters and constraints are obtained to generate the mathematical model. The design Collaboratory allows distributed design engineers to securely share data, make queries and collaborate.

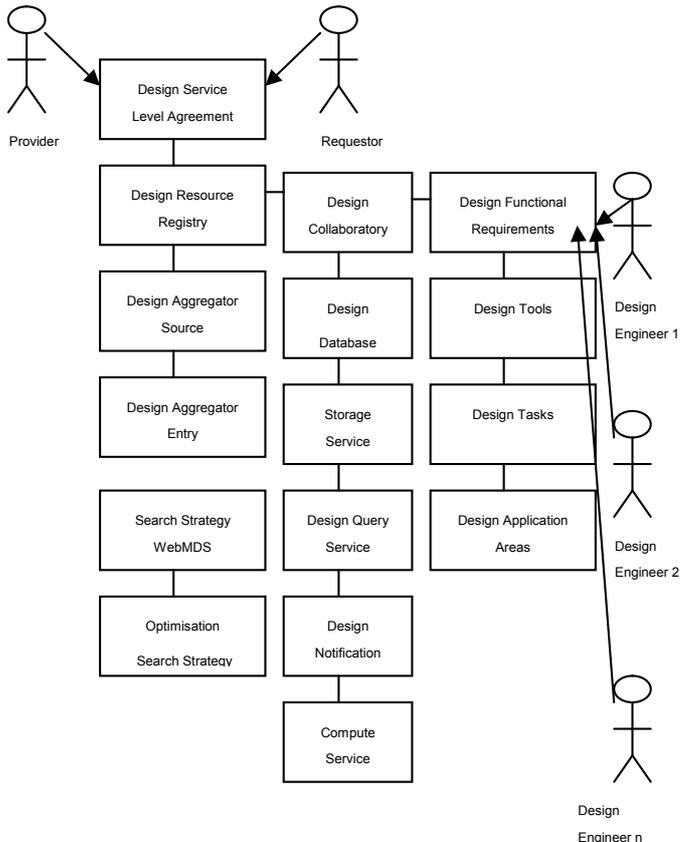


Figure 1: Service specification framework

3.2 DECGrid Architecture

The framework is implemented using 8 computers that served as grid nodes. CentOS 4.0 Linux is used as the operating system. Globus Toolkit 4.0.4 (GT4) is used as the middleware and Condor 7.0.0 as the resource

scheduler. Apache web server and WSRF (Web Services Resource Framework) are used to allow the application run both as grid services and web services. Figure 2 describes the DECGrid architecture. Service provider publishes design and optimisation resources and requestors (design engineers) access the resources and collaborate. This architecture ensures that only authorised design engineers have access to optimisation resources. This is done by entering users' grid certificate details in the map-file of grid-security as well as allowing host machines to have access to the grid resources on other nodes. Users have different levels of rights to ensure integrity of design data. The certificate authority ensures that certificates expires every 24 hours and requires that users update certificates after expiration. The DECGrid architecture makes use of the grid service data elements (SDE) to present the properties of optimisation resources as MDO services. SDE can be static or dynamic [10]. The SDE are described in the XML schema which implements the different MDO services whose life-time management are done by uniquely identifying each service instance by an element called grid service handle (GSH) and are supported and held by the grid service record (GSR). This enables the MDO services to be invoked at the same time by different collaborating users without much interference as each service instance has its state and life time managed differently through the GSH and GSR. The important grid service interfaces (portTypes) that implements the interfaces are the GridService, NotificationSource, NotificationSink, Registry and Factory. The GridService portType uses the FindServiceData to query optimisation resources and is responsible for service discovery, NotificationSource is notification of the instances of services to service requestors, Registry registers or deregisters services and Factory create service instances using CreateService operation.

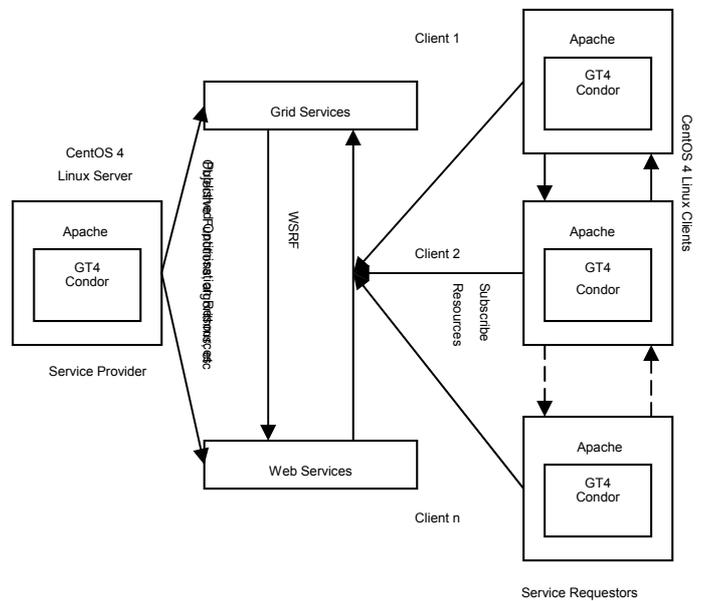


Figure 2: DECGrid Architecture

4 CASE STUDY: THE WELDED BEAM PROBLEM

To demonstrate the workings of DECGrid, a real-life case study-the welded beam problem is used. This case study is chosen because it requires the optimisation of at least 2 objectives and has variables that are sensitive to different parameters especially the lower and upper bounds. The objectives can be optimised in different nodes of the grid by running optimisation services on those nodes as well

as making some nodes build the mathematical model and others run parameter input service at the same time. This demonstrates how different grid services can handle different optimisation tasks concurrently for efficient design optimisation of products that have multiple objectives, constraints and variables. This problem describes how a beam needs to be welded on another beam and must carry a certain load. The objective of the design is to minimise cost of fabrication and minimise the end deflection. Here, the overhang portion of the beam and the applied force are specified, thus making the cross-sectional dimensions of the beam (b , t) and the welded dimensions (h , l) as the variables. The problem has four constraints. The first constraint is to make sure that the shear stress at the developed support location of the beam is smaller than the allowable shear strength of the material. The second constraint is to make sure that normal stress at the support location is smaller than the allowable yield strength of the material. The third constraint is to ensure that the thickness of the beam is not smaller than the weld thickness from a practical standpoint and the fourth constraint ensures that the allowable buckling load of the beam is more than the applied load.

3.1 Mathematical model building service

A mathematical model building service provides the interactive session for design engineers to build models collaboratively. DECGrid has 2 graphical user interfaces (GUI) for mathematical model building service. They are implicit and explicit interfaces. The explicit interface has 4 input fields that enable experts to build generic explicit mathematical models while the implicit interface provides a field where experts can upload mathematical models from a file. This case study uses the explicit interface so this paper will concentrate on explicit mathematical building service. Figure 3 below shows the explicit mathematical model service builder running. This interface is run on 2 nodes of the grid, allowing 1 expert each to build part of the welded beam model. The first expert sits at the first node and enters '6000' in the variable field and ';' (because the last variable in a relation must end with semi-colon in C programming language) in the right operators field and next step is to enter 'F=' (force applied) in the output field which when submitted and displayed will show $F=6000$; as created. The expert then submits this which get saved in the database for reuse.

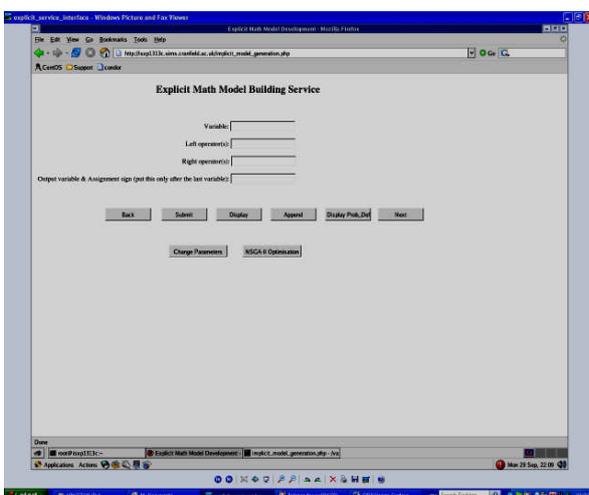


Figure 3: Explicit mathematical model building service interface

This process is repeated by this expert for the first part of the model. This part model is shown below.

```
F=6000; //force applied
b= xreal[0]; //beam width
t= xreal[1]; //beam length
h= xreal[2]; //weld width
l= xreal[3]; //weld length
l1=pow(l,2);
ht1=pow(h+t,2);
dt= 2.1952/(pow(t,3)*b); //objective
r1= F/pow(2*h*l,0.5);
```

NSGA-II uses 'xreal[]' for real variables starting with index 0 to n number of variables and 'constr[]' and 'obj[]' for constraints and objectives respectively. The second experts uses similar interface on the second node to build the remaining part of the model. The second part of the mode is shown below.

```
c1=1.10471*pow(h,2)*l;
c2=0.04811*t*b;
c3=14.0+l;
c4=pow(r1,2);
c5=0.25*(l1+ht1);
c6=pow(c5,0.5);
c7=F*(14.0+(0.5*l));
c8=(l1/12)+(0.25*ht1);
c9=0.707*h*l;
c10=2*c9*c8;
c11=c7*c6;
r2=c11/c10;
c12=pow(r2,2);
c13=l*r1*r2;
dt= 504000/(pow(t,2)*b); // objective 2
r=pow((c4+c12+(c13/c6)),0.5);
PC= 64746.022*(1-(0.0282346*t))*(t*pow(b,3));
obj[0]=c1+(c2*c3); //objective 1
obj[1]= dt; //objective 2
constr[0]= 13600-r; //constraint 1
constr[1]= 30000-dt; //constraint 2
constr[2]= b-h; //constraint 3
constr[3]= PC-F; //constraint 4
```

It is assumed that these 2 experts are specialists in the different mathematical components of the welded beam problem. This collaboration may trigger innovative and creative thinking in the design process [11]. The 2 components are merged and the complete model can be accessed by authorised grid users as an optimisation grid resource. Companies need to engage in such collaborative design approach to gain competitive advantage as customers are demanding better and customised products [12].

3.2 NSGA-II optimisation service

The mathematical model is now available as a service for other experts to use. The system has capability to link the mathematical model built to NSGA-II through 'problemdef.c' (the file that contains mathematical models

ready for optimisation) file. The third expert sits at the third node and clicks perform this linking automatically. The expert then uses the NSGA-II Optimisation facility to start the NSGA-II Optimisation Service. The service consists of 3 interfaces. The first interface allows the expert to enter 5 parameters namely number of generation, number of population, number of objectives, number of constraints and number of real variables. When these 5 parameters are entered, they get stored in a text file as well as the PostgreSQL database which can be retrieved and reused. The last value (number of variable) is used to generate a sub-interface under the first interface for the lower and upper bounds of the real variables. This process now takes the expert to the second service interface which enables the expert to enter probability of crossover of real variable, probability of mutation of real variable, distribution crossover index, distribution mutation index and number of binary numbers. Another sub-interface is generated if the last value is more than 0 for the lower and upper bounds of binary variables. The last interface allows the expert to enter choice of display (0 or 1), objective index for x-axis and objective index for y-axis. Now the optimisation can be run and results observed in graphical form or text format. The interface for NSGA-II parameters service is shown in Figure 4 below. The mathematical model building service which brought experts from different disciplines and the optimisation parameter service which allows interdependent sub-services to interact demonstrates the efficiency of distributed grid services for design optimisation.

Parameters	Description
500	number of generations
100	number of population
2	number of objectives
4	number of constraints
4	number of real variables
-5	lower bound for first variable
10	upper bound for first variable
0	lower bound for second variable
15	upper bound for second variable
-3	lower bound for third variable
3	upper bound for third variable
0	lower bound for third variable
5	upper bound for fourth variable
0.8	probability of crossover of real variable
0.05	probability of mutation of real variable
10	Distribution index for real variable crossover
50	Distribution index for real variable polynomial mutation
0	number of binary variables
1	enter 1 to display gnuplot or 0 to display only results
1	x axis objective index
2	y axis objective index

Table 1: Welded beam input parameters

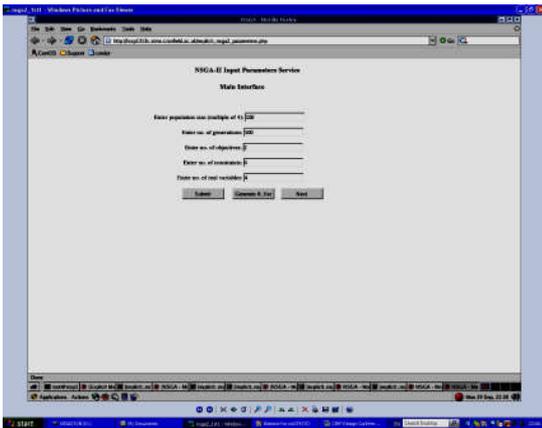


Figure 4: First parameter interface

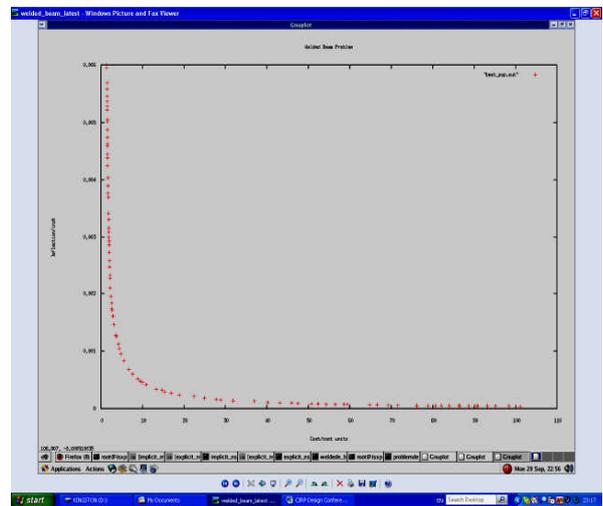


Figure 5: One solution of welded beam problem

5 RESULTS

The system is linked to the GNUPLLOT text and graphical result display interface. Different parameters are used to come up with an acceptable optimum solution for the welded beam problem. Continuous refinement of the parameters and running the optimisation proved to have varying and interesting results. One of the results of running 500 generations and 100 populations with corresponding parameters as shown in Table 1 is shown in the following graph in Figure 5. The welded beam problem has been used by many researchers [5], [13], [14]. The results when compared to some in literatures showed that in certain circumstances, our results have wider spread along the cost values and shorter along the deflection values and vice versa. Using more nodes to distribute the computations showed better spread of results than the ones using fewer nodes. This demonstrates the computational synergy of grid services in obtaining results that both show convergence and diversity of results so that experts can make better decisions on the choice of design variables and parameters.

6 SUMMARY

MDO is a field that brings together multidisciplinary experts to collaborate. These experts may be geographically distributed. The grid provides a problem solving environment (PSE) for MDO experts to collaborate and publish mathematical models, algorithms and parameters as grid resources and services. DECGrid is an implementation of such PSE collaborative tools to enable companies use grid services for multi-objective optimisation. These services provide tools as grid services to designers to enhance exploration in the design and ultimately produce wider optimum solution space for design engineers to choose from.

7 ACKNOWLEDEMENTS

We wish to thank the Petroleum Technology Development Fund (PTDF) of Nigeria and Cranfield University UK for sponsoring this research.

8 REFERENCES

- [1] Liu, H., Tang, M. and Frazer, J. H., 2004, Supporting Creative Design in A Visual Evolutionary Computing Environment. *Advances in Engineering Software.*, 35:261–271.
- [2] AISairafi, S., Emmanouil, F. S., Ghanem, M., Giannadakis, N., Guo, Y., Kalaitzopoulos Di., Osmond M., Rowe A., Syed J., Wendel P., 2003, The Design of Discovery Net: Towards Open Grid Services for Knowledge Discovery. *The International Journal of High Performance Computing Applications*, 17(3):297–315.
- [3] Foster, I., Kesselman, C., and Tuecke, S., 2001, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Intl J. Supercomputer Applications*.
- [4] Hiroyasu, T., Miki, M., Shimosaka, H. and Dongarra, J. Optimization Problem Solving System using Grid RPC. *Nihon Kikai Gakkai Sekkei Kogaku, Shisutemu Bumon Koenkai Koen Ronbunshu. L1283A*.
- [5] Parmee, I. C., Araham, J., Shackelford, M., Rana, O. F. and Shaikhali, A., 2005, Towards Autonomous Evolutionary Design Systems Via Grid-Based Technologies. *Proceedings of ASCE 2005 International Conference on Computing in Civil Engineering*.
- [6] Eres, M. H., Pound, G. E., Keane, A. J. and Cox, S. J., 2004, User Deployment of Grid Toolkits to Engineers . *Proceedings of the UK e-Science All Hands Meeting. AHM*.
- [7] Jackson, T, Austin, J., Fletcher, M. and Jessop, M.,2003, Delivering a Grid enabled Distributed Aircraft Maintenance Environment (DAME). *FGCS*.
- [8] Goel, S., Taya, S. S. and Sobolewski, M., 2005, Preliminary Design using Distributed Service-Based Computing. *Next Generation Concurrent Engineering*. 113-120, ISSN: 0-9768246-0-4.
- [9] Litke, A.,Konstanteli, K., Andronikou, V., Chatzis, S. and Varvarigou, T., 2008, Managing Service Level Agreement Contracts In OGSA-Based Grids. *FGCS: 24: 245-258*.
- [10] Antonioletti, M., Atkinson, M., Baxter, R., Borley, A., Neil, P., Hong, C., Collins, B., Hardman, N., Hume, A. C., Knox, A., Jackson, M., Krause, A., Laws, S., Magowan, J., Paton, N. W., Pearson, D., Sugden, T., Watson, P. and Westhead, M., 2005, The Design And Implementation Of Grid Database Services In OGSA-DAI. *Concurrency And Computation: Practice And Experience*. 17: 357–376.
- [11] Gero, J. S., 1994, Computational Models for Creative Design Processes. *Dartnall, T. Artificial Intelligence and creativity*. Springer, 269-280.
- [12] Fan, L. Q., Kumar, A. S., Jagdish, B. N. and Bok, S. H., 2008, Development Of A Distributed Collaborative Design Framework Within Peer-To-Peer Environment. *Computer Aided Design Journal*.
- [13] Deb, K., 2000, An Efficient Constraint Handling Method For Genetic Algorithms. *Comput. Methods Appl. Mech. Engrg.* 18 : 311-338.
- [14] Carlos, A. and Coello, C., 2000, Use Of A Self-Adaptive Penalty Approach For Engineering Optimization Problems. *Computers in Industry*. 41 :113–127.