Grid Enabled Architecture For DWDM Network Design And Optimization Tool

[1]Bhupinder Singh, [2]Seema Bawa

GJCST Computing Classification
B.2.1 , C.2.4

Abstract- **High bandwidth networks are indispensable to support the present ever increasing demand for various services on internet and to cater highly bandwidth extensive application such as video streaming and multimedia conferences. Dense Wavelength Division Multiplexing (DWDM) technology based optical networks provide potentially large transmission capacity that has an obvious advantage from both technical and economic perspectives.DWDM network design and optimization tools are developed to aid the designing and deployment of networks. Design tools play an important role in facilitating routing and wavelength assignment, filter placement, DCM and amplifier placement with the aim of minimizing the overall cost of network, in minimum amount of time. Designing and optimizing large networks require lot more processing power than a single desktop machine can actually provide. This scarcity of processing power results in either sub-optimal or infeasible solutions. This paper presents the grid enabled architecture for DWDM network design and optimization tool, which aims to harness the processing power of existing idle resources in an organization to quickly provide the optimized design for huge networks.**

Keywords- DWDM network design, Grid Computing, optical network design, Grid applications.

I. Introduction

Need for the optical networking is driven primarily by three major technological advancements: 1. Dense Wavelength Division Multiplexing (DWDM) 2. Broadband optical amplification and 3. Wavelengthgranularity optical switching. Optical networks can span from hundreds to thousands of kilometres and involve thousands of different network elements [1]. Analysis and design of optical network require the solution of nonlinear equations from optical physics to combinatorial solution of NP- hard [2] optimization problems. Optical networks can also require significant investment, ranging from millions to hundreds of millions of dollars. Last but not the least,increased competition in the business also demands for an optimized design (both cost effective and energy efficient) as quickly as possible. All these reasons strongly push the demand for computer aided software tools to design, optimize, simulate, and operate DWDM networks. These tools address several aspects of optical network design, from helping sales personnel to provide optimized bids to clients, to providing a detailed selection of optical components, to planning wavelength growth in operating networks. DWDM network design tools are basically Graphical User Interface (GUI) based desktop software as depicted in the Fig.1. Flow of DWDM network design process is depicted in the flow chart given in Fig. 2.

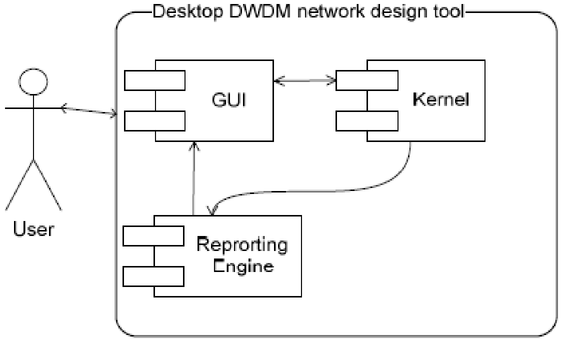

Fig. 1: High level architecture of a Desktop based DWDM design tool

It mainly consists of the three components. First, GUI based canvas that facilitates the users to draw the physical layout of the network and specify the demands. Second major component is Kernel, which is the implementation of the set of algorithms that try to optimize the overall network design, so as to fulfil the requested demands, as well as minimize the overall cost of the network. Finally third component is the reporting engine, which provides the different type of reports and network diagrams that help the sales person to place their bids and also help the deployment engineers during the actual installation of the network.

Routing and wavelength assignment (RWA), optimized DCM placement and optimized amplifier (EDFA or Raman) placement, are compute intensive problems [3]-[5]. While designing very large networks, the need for computing resource and memory grows exponentially. This cannot be fulfilled by desktop machine due to very limited availability of the resources. This leads to either infeasible or suboptimal solutions which are not cost effective. Alternate solution is to transfer the processing logic to a high end,multi-processor server machine with parallel processing capability and huge memory. But this solution is accompanied by the increased cost of server hardware.

Grid computing [6] is an emerging trend for making easy access to computing resources like an electric power grid.Inspired by the electrical power grid‘s pervasiveness, ease of use, and reliability, computational grid provides an analogous infrastructure for wide-area parallel and distributed computing. A grid enables the sharing, selection, and aggregation of a wide variety of geographically

[1]Bhupinder Singh, Module Lead, Design Tools Department, Ciena India Pvt. Ltd. Gurgaon (Haryana), India.
(bsaini@ciena.com)
[2]Seema Bawa, Professor, Computer Science and Engineering Department, Thapar University Patiala. (Punjab), India .
(seema@thapar.edu)

distributed resources including supercomputers, storage systems, data sources, and specialized devices owned by different organizations for solving large-scale resource intensive problems in science, engineering, and commerce. Since DWDM network design and optimization is also A resource intensive problem, hence it is a suitable candidate to be tackled in grid environment.This paper proposes a grid enabled architecture for optimized DWDM network design. Use of existing procedures and components makes this architecture practically realizable with little extra effort. The proposed architecture aims to harness the power of idle resources in an organization, and use it for DWDM network design and optimization.

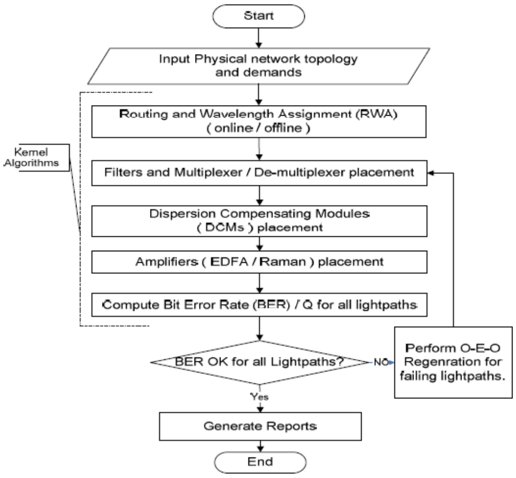
Start
Input Physical network topology and demands
Routing and Wavelength Assignment (RWA) (online / offline)
Kernel Algorithms
Filters and Multiplexer / De-multiplexer placement
Dispersion Compensating Modules (DCMs) placement
Amplifiers (EDFA / Raman) placement
Compute Bit Error Rate (BER) / Q for all lightpaths
BER OK for all Lightpaths?
NO
Perform O-E-O Regenration for failing lightpaths.
Yes
Generate Reports
End

Fig.2: Steps of DWDM network design

II. PROPOSED GRID ENABLED ARCHITECTURE

The proposed grid enabled architecture is an extension to existing desktop based DWDM design tool. Desktop based software is enhanced to a grid application, where processing is seamlessly distributed to many idle computing resources that are registered with a grid. The results computed by different processing resources on the grid are combined by the grid scheduler and are sent back. Details are discussed in next subsections. Proposed grid enabled architecture is depicted in Fig. 3.

A. Pre-Requisites For Proposed Architecture

Before discussing any further details, first we need to see how DWDM network design and optimization problem qualifies as a suitable candidate to be executed as grid application. Since grid environment, by nature, is loosely coupled, with heterogeneous resources and high latency networked environment, so grid application should be such that their work units can be parallelized into a number of independent computation units and should have high computation vs. communication time ratio [7].First, is already supported by the discussion in the previous section that DWDM design and optimization algorithms (like RWA, DCMs and Amplifier placement) are compute intensive and require a lot of processing power. To support the second point we propose to formulate the compute intensive algorithms as a linear programming (LP) / mixed integer linear programming problem (MILP). Details on how to formulate RWA and amplifier and DCM placement into LP / MILP problem is discussed in the [3]-[5]. Providing solution to large MILP problems is again A compute intensive problem. Many commercial LP / MILP solvers like CPlex, GurObi, Xpress etc. and non commercial solvers like COIN-OR‘s CBC exist that support parallel processing. So network design and optimization application qualifies to be designed as a grid application.Next point of discussion is how to distribute the parallel threads over the grid. Most solvers can only harness the power of more than one processing elements in a single machine but cannot use the processing capabilities of idle processing elements in other machines. To achieve this General Algebraic Modelling System (GAMS) [8] is used, which is specifically designed for modelling linear, nonlinear and mixed integer optimization problems. GAMS is a high-level modelling system for mathematical programming and optimization. It consists of a language compiler and a stable of integrated high-performance solvers. GAMS is tailored for complex, large scale modelling applications. A large part of the time required to develop a model is involved in ‗data preparation and transformation‘ and report preparation. Each model requires many hours of analyst and programming time to organize the data and write the programs that would transform the data into the form required by the mathematical programming optimizers. Reference [9] describes the GAMS for modelling optimization problems on a grid computing environment. This framework is easy to adapt to multiple grid engines and can seamlessly integrate evolving mechanisms from particular computing platforms. So, by integrating with any of the existing MILP optimizers, GAMS provides a lightweight, portable and powerful framework for optimization on a grid.

B. Information And Control Flow In Proposed Architecture

It is clear from the discussion in the previous sub-section that required components / procedures already exist in one form or another. A slight modification in existing procedures and a little integration effort paves the way to A very powerful and a cost-effective grid enabled network design and optimization tool. Fig. 3 shows the proposed grid enabled architecture.

GUI- provides a canvas or other input interface to provide physical network layout and demands as input to the software. It is also used to display the reports and layout diagrams to the user, that are prepared by the reporting engine

Kernel- consists of a set of algorithms as discussed earlier assuming the compute – intensive algorithms like RWA, DCM placement and amplifier placement etc. are formulated as LP or MILP.

Reporting Engine is responsible for collecting the results from the kernel and prepare different type of reports (e.g. bills of materials, network design and deployment diagrams etc.), as required by user.

Kernel-GAMS- interface provides the way to map MILP based formulation of kernel algorithms to GAMS model (A form understandable by GAMS language complier) and calls GAMS APIs (application programming interface) to solve the model. Also it returns the solution back to the kernel.

GAMS- provides a platform that accepts the input model and converts it into a form understandable by the integrated optimizer that actually does the job of solving the GAMS model. As soon as a solve statement is encountered while executing a GAMS program, control is passed to the script.This script is responsible for running the optimizer on the problem instance and passing back the solution to the GAMS.

Grid Engine- manages a pool of connected computers available as a common computing resource. It provides the effective sharing and utilization of idle computing resources and massive parallel task execution. Scheduler (e.g. condor, sun grid engine, globus, etc.) handles all management tasks. Machine (M1-Mn) is a set of machines that are registered with grid engine to share the computing resources. Any machine in the organization can be a part of a grid environment. Grid scheduler utilizes the idle CPU cycles from these machines.

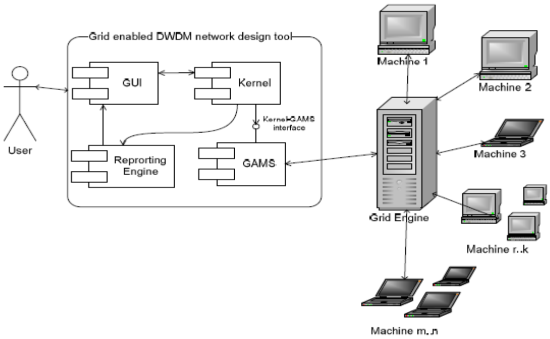

Fig. 3: Proposed grid enabled architecture

Information exchange and control flow among various components is as follows: GUI based canvas used to draw the physical layout of the network and provide it as input to the kernel along with the demands. Kernel executes a series of algorithms to design the networks. For compute intensive algorithms formulated as MILP, it prepares a GAMS understandable model and calls the GAMS APIs to solve this model via Kernel-GAMS interface. As soon as a solve statement is encountered while executing a GAMS program, control is passed to the script. This script is responsible for running the optimizer on the problem instance and passing back the solution to the GAMS. In a grid environment, GAMS use the file system to give each instance its own environment and its own directory. The script then schedules the solver execution. The submission script centralized all information required to tailor the system to A specific grid engine. Grid engine distributes the processing over the different machines, connected to the grid. The results are collected by the grid engine and returned back to the GAMS, which assembles the solution for the submitted model and returns to the kernel for further processing. As soon as the network design and optimization process is complete the designed network is used by the reporting engine to prepare different type of reports as required by the user.

III. Conclusion

A grid enabled architecture for DWDM network design and optimization tool is presented. This simplified architecture can be realized as an extension to existing desktop based network design software with slight modifications. The use of existing procedures and integration of existing components paves a way to the practical realization of very powerful and cost effective software with little extra cost and integration effort. Hence the proposed grid enabled architecture aims to harness the power of idle resources in an organization, and use it for DWDM network design and optimization. Effective utilization of the idle resources caters the exponentially growing need for computing resources for designing huge optical networks. Massive parallel processing capability provided by the computational grid cuts down the need for very high end and costly server hardware.

IV. References

1) Chekuri, C. et.al. (2006), Design tools for transparent optical networks, Bell Labs Technical Journal, vol. 11, Pp. 129-143.
2) Garey, M. R. and Johnson, D. S. (1979), Computers and Intractability: A Guide to the Theory of NP-Completeness,ed. W. H. Freeman, New York,NY.
3) Ozdaglar, A.E. and Bertsekas, D.P. (2003), Routing and wavelength assignement in optical networks", IEEE/ACM Transactions on Networking (TON) archive, vol. 11, Pp.259-272.
4) Ramamurthy, B., Iness, J. and Mukherjee, B. (1998), Optimizing amplifier placements in a multiwavelength optical LAN / MAN: the equally powered-wavelengths case, Journal of Lightwave Technology. vol. 16, Pp.1560-1569.
5) Minagar, A., Premaratne, M. and Tran, A. V. (2006), Cost optimal allocation of amplifiers and DCMs in WDM ring networks, Optics Express, vol. 14, Pp. 10278-10291.
6) Berstis, V. (2002), Fundamentals of Grid Computing Available: http://www.redbooks.ibm.com/abstracts/ redp3613.html.
7) Nadiminti, K., Luther A. and Buyya, R. (2005), User Guide for Alchemi 1.0 Available:http://www.gridbus.org/ ~alchemi/files/1.0.beta/docs/ AlchemiManualv.1.0.htm.
8) The GAMS System. Available: http://www.gams.com/docs/ intro.htm.
9) Bussieck, M. R., Ferris, M. C. and Meeraus, A. (2009), Grid-Enabled Optimization with GAMS, (2009) INFORMS Journal on Computing. Vol. 21 (3), Pp: 349-362.