Parallel Mesh Generation: Web-Services and COTS Software*

Nikos Chrisochoides, Andrey Chernikov, Andriy Kot, and Andriy Fedorov

Computer Science Department,
College of William and Mary
Williamsburg VA 23187, USA

Existing parallel mesh generation codes based on the parallelization of past sequential mesh generation technology. Given that it takes about ten years to develop the software infrastructure for *sequential industrial strength mesh generation libraries*, it is clear that traditional parallelization approaches deliver technology that is outdated. This problem becomes more serious if one considers that the improvements of sequential codes in terms of quality, speed, and functionality are open ended. In this paper we present a new approach to parallel mesh generation for addressing this serious problem. Our approach is based on of-the-shelf sequential mesh generation software for delivering industrial strength parallel mesh generation (web-)services over the Grid.

Parallel mesh generation procedures decompose the original mesh generation problem into N smaller subproblems that can be meshed in parallel using P ($<<$ N) nodes. The subproblems can be formulated to be either tightly or partially coupled or even decoupled to each other. The coupling of the subproblems determines the intensity of the communication and the degree of dependency (or synchronization) between the subproblems. In this paper we focus on a partially decoupled approach which balances trade-offs between domain decomposition and communication.

Our parallel mesh generation codes are implemented on top of a runtime system that provides support for one-sided communication, remote service request, global address space in the context of data mobility, transparent routing of messages, and automatic dynamic load balancing. The web-service is developed on top of Cornell's Web Services Framework. Our preliminary experimental data show that on a cluster of more than 100 workstations, for a cross section of rocket pipe, using a simple (block) decomposition we can generate about half a billion elements in less than three minutes.

*This work was supported by NSF Career Award #CCR-0049086, ITR #ACI-0085969, RI #EIA-9972853, NGS #EIA-0203974, and ITR-0312980