ISA TRANSACTIONS®

ISA Transactions 42 (2003) 615–630

Simulating thermal power plant processes on a message passing environment

A. Gupta,[a,*] R. Ganguly,[b,*] S. Chakraborty,[c,†] C. Mazumdar,[d,‡] D. Popovic[e,§]

[a]*PPE Dept., Jadavpur University, Calcutta, India*
[b]*PPE Dept., Jadavpur University, Calcutta, India*
[c]*ME Dept., IIT Kharagpur, India*
[d]*CSE Dept., Jadavpur University, Calcutta, India*
[e]*IAT, University of Bremen, Germany*

(Received 11 June 2001; accepted 10 November 2002)

Abstract

Simulators play a very important role in the operation of thermal power plants and also in the design of control systems for these plants. To cater to this requirement elaborate methodologies have been developed to simulate thermal power plant processes in an interactive way. Due to the intensive computations involved, such simulators use one or more high through-put computers known as the simulation computers. This paper puts forward a method where parallel processing on a low latency message passing environment has been used to simulate thermal power plant processes following a modular approach. This eliminates the need of an expensive high through-put simulation computer, thus cutting down the hardware cost associated with a simulator and increasing the system reliability manifold. © 2003 ISA—The Instrumentation, Systems, and Automation Society.

Keywords: Boiler; Explicit forward difference; Message passing interface; Task decomposition; Graph matching

1. Introduction

Contemporary simulators are usually multicomputer systems, often integrated over a local area network (LAN). The architecture consists of one or more computers called the *field simulators* or the *simulation computers* which actually simulate the plant. In a way, these act as servers. The client nodes function as *simulator stations* or *instructor work stations* and both are essentially man-machine interfaces (MMI). Fig. 1 shows this typical architecture where the nodes marked as MMI are the *simulator stations*.

Because of the heavy computational requirement, the field simulators need to be very high through-put computers and they are usually high-end work stations. The computing power available with the simulator stations and the instructor work stations remains largely unharnessed. Logically therefore if the computational work assigned to the field simulator is distributed across all nodes, the high through-put field simulator could be eliminated resulting in the following benefits:

- ♦ The hardware cost is lower as a costly work station is replaced by cheaper microcomputers
- ♦ Better reliability.
- ♦ More MMI nodes for user interaction, for a given number of processors as each processor can act as a MMI node while performing simulation of the module(s) assigned to it.

**E-mail address*: jupower@vsnl.com
†*E-mail address*: suman@mech.iitkgp.ernet.in
‡*E-mail address*: chandanm@vsnl.com
§*E-mail address*: popovic@iat.uni-bremen.de

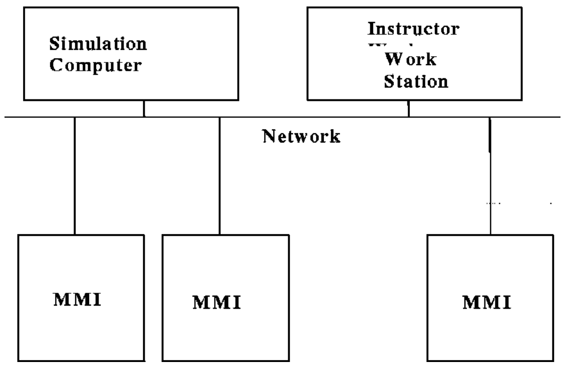

Fig. 1. A typical simulator configuration.

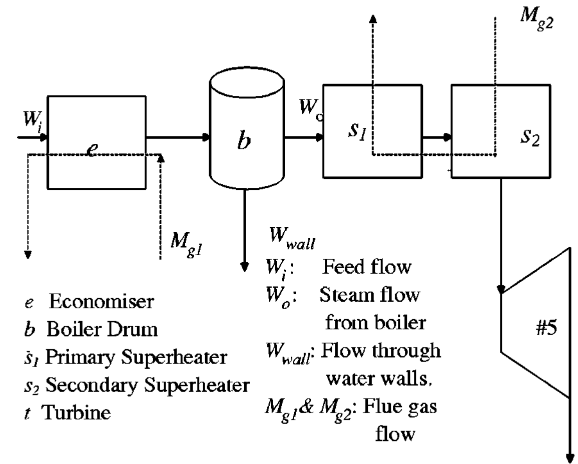

Fig. 2. Schematic representation of the simulated system.

However, there are several issues involved:

- ♦ Expressing the models in a distributed form.
- ♦ Identification of a proper parallel processing platform.
- ♦ Distributing the simulation tasks using the platform so as to meet all the requirements of a training simulator.
- ♦ Identification of a proper simulation tool to build the simulator.
- ♦ Defining a proper restart mechanism for the distributed simulator.

The primary motivation for the present research arises from the need to solve these problems so as to develop simulators using a cluster of common off the shelf (COT) personal computers (PC's).

The issue of expressing the models in a distributed form has been discussed in Ref. [1]. Reference [1] also puts forward a method by which the simulation task involving a large number of interacting multiple input multiple output (MIMO) systems can be distributed over a cluster of work stations using message passing interface (MPI) [2] using concepts of reduced dependency graph (RDG) [3].

In this paper, distributed simulation is attempted for simulation of a typical thermal power plant process involving water and flue gas as shown in Fig. 2.

For this simulation, dynamic models are developed for the economizer, drum boiler, primary superheater, secondary superheater, and the turbine using either the state-space approach or the forward difference technique with proper identification of the input and output parameters associated with the model of each component. Each model is then run as a separate process and benchmarked using a HP Vectra personal computer (P2@233 MHz) with Windows NT as the Operating System.

Next, the sequential and parallel codes for the simulation are developed to simulate a scenario where the steam outflow from the boiler drum is increased slowly as a ramp and all the parameters (set of all outputs associated with the five modules) are computed using the dynamic models developed. The RDG for the system schematically represented by Fig. 2 is then developed using the schematic representation of the interaction between the different modules. This is used to obtain the optimal task decomposition on a 2 processor and a 4 processor cluster formed by integrating 2 or 4 HP Vectra personal computers (P2@233 MHz) on a switched fast Ethernet LAN. For task decomposition, a new methodology based on *graph matching* is used. This methodology uses the RDG as the *task graph* and a *circulant graph* [4] as the *processor graph* which involves multiple rounds of data transfers between the processors, to obtain the optimal mapping of simulation modules on processors. The graph matching technique uses the well-known A^* [5] algorithm to achieve a homomorphic mapping using a *task graph* and a *processor graph* and was proposed first in Ref. [6]. This basic technique is valid for *processor graphs* where all data transfers are assumed to occur simultaneously. This has been extended to obtain optimal mapping using a RDG and a *circulant graph* where data transfers in a sequence can also be represented. This methodology therefore generates the optimal mapping of