PARALLEL OPTIMIZATION OF OPERATION FOR ETHYLENE CRACKING FURNACE

Jian Wang, Qiang Xu, Bingzhen Chen*, Xiaorong He
Department of Chemical Engineering, Tsinghua University
Beijing, China 100084

Abstract

The solution of realistic simulation and optimization problems is computationally very intense. Along with the development of parallel computing environment, it is necessary to construct an effective parallel strategy in order to exploit supercomputing power. A parallel solution strategy was developed for the optimization of the whole periodic operation of the cracking furnace. The most time-consuming computation step for the gradients is deserialized, an approximate linear speed-up is acquired and the optimization calculation is well accelerated. The optimization work was carried out successfully on a cluster of workstations and properties of the coarse-grained parallel computation technique were analyzed. The results suggest that the execution time of the steam cracking process optimization can be reduced significantly by distributing the heavy simulation duties onto several computing nodes, which communicate via communication networks.

Keywords

Optimization, Simulation, Parallel computing, Distributed memory, Steam cracking, Ethylene plant

Introduction

For the simulation and optimization of industrial-scale chemical processes, the computational time is enormous because of the complexity of the systems. Along with the rapid progress in computer science and related technologies, utilizing parallel computing techniques to improve the computational efficiency has become significative (Zitney, 1995). Because of the trend of distributed computing, researches in this field were focused on coarse-grained parallel algorithms. Quite a few researches have been done successfully on the solution of large sparse systems of linear equations (Mallya, 1997a, 1997b), the most computationally intensive step of the simulation of chemical processes. For distributed memory parallel architectures, however, there are few investigations about coarse-grained parallel solution strategies for the integrated optimization of industrial processes.

Pyrolysis of naphtha and other hydrocarbons is of primary importance in the manufacture of olefins. In general, the complex reactions set existing in the cracking coils determines the yields of various products of an ethylene plant. The formation of coke reduces the overall heat transfer coefficient and increases the pressure drop over the reactor. When the increasing of the pressure drop or the external tube skin temperature exceeds certain limits, the furnace has to be removed from production to clear away the coke layer (Schools, 1997). Because the decoking operation reduces the on-stream time of the firebox and production capacity, it is very valuable to optimize the whole periodic operations and to determine the optimal duration. Unfortunately, the optimization of this problem is quite time-consuming (several hours), thus it is necessary to develop an effective and practical solution strategy for the optimization.

In this paper, a parallel optimization scheme for the whole periodic operation of a practical cracking furnace is proposed. The test implementation of the software is successfully performed using the PVM (Parallel Virtual Machine) message-passing system on a cluster of workstations. The execution times and other parallel features of the solution are listed and discussed.

Parallel computing Environment

For our distributed parallel environment, a parallel computing system is composed of a cluster of workstations

*Corresponding author. Dept. of Chem. Eng., Tsinghua Univ., Beijing, China 100084
Tel.: +86-10-62784572; fax: +86-10-62770304; *E-mail address*: dcecbz@tsinghua.edu.cn

interconnected through network. Each workstation is equipped with a PIII-800 CPU and 256 MB of RAM memory, and operates under the Windows 2000 operating system.

PVM is a flexible software package that permits a heterogeneous collection of Unix and/or NT computers hooked together by a network to be used as a single large parallel computer. In this research work, PVM is used to accomplish message-passing operation and control the parallel computing system. PVM source code can be downloaded from Netlib (http://www.netlib.org/pvm3).

Parallel optimization scheme

Description of the process

In this paper we present the optimization of the whole periodic operation of naphtha pyrolysis in a GK-V industrial cracking furnace.

The reactor model equations given by Froment (1990), Heyndericks (1998) and Xu (2001) are used. The main assumptions of this model are:

1. According to the run time, the whole period is divided into many quasi-steady states, because the coking reactions are much slower than cracking reactions.
2. The steam-oil ratio is fixed.
3. The molecular kinetic model and coke formation mode applied in the simulation were developed by Kumar & Kunzru (1985a, 1985b).

The yield of ethylene, which is the key economic performance of an ethylene plant, is mainly determined by the cracking coil. But the fixed coil outlet temperature (COT) is usually adopted in practice at present. Hence, it is promising to get greater profits by using the optimal COTs.

The optimization problem that is studied in this paper is to get the maximal total ethylene yield by manipulating the COTs in different time horizons. The optimization model is described as following problem:

$$J = \max \sum_{m=1}^{M} y_m \tag{1}$$

$$s.t. \quad \begin{cases} [y_m, Tw_m, \Delta P_m] = f_m(\theta, z_m, \delta_{C,m}, \sigma) \\ \delta_{C,m+1} = h(y_m, z_m) \\ g(z_m, Tw_m, \Delta P_m) \le 0 \\ \delta_{C,1} = 0 \\ m = 1,2,....., M \end{cases}$$

Here y_m is the yield of ethylene during the mth quasi-steady states, Tw_m is the external tube skin temperature, ΔP_m is the pressure drop over the reactor, θ is the rate of feed flow, $\delta_{C,m}$ is the coke layer thickness, z_m represents COT and σ represents other variables.

In the constraint equations, $h(\bullet)$ represents the rate equation of coke formation. The material equation that is used here is:

$$r_c = 1.8 \times 10^{15} \exp\left(\frac{222.00}{RT}\right) \cdot C_a^{2.05} \tag{2}$$

$f(\bullet)$ is the reactor mechanism model equations. It can be described by a group of ordinary differential equations (ODEs). (Heyndericks, 1998)

$$\frac{dN_m}{dL} = S \sum_i \upsilon_{im} \cdot r_i \tag{3}$$

$$\frac{dP}{dL} = -\frac{9.89067 f E G^2}{19.62 \rho D_i} \tag{4}$$

$$\frac{dT}{dL} = \frac{k\pi D_o (T_w - T) - \sum_m \Delta H_{fm}^0 \cdot \frac{dN_m}{dL}}{\sum_m Cp_m \cdot N_m + C_{pH_2O} \cdot N_{H_2O}} \tag{5}$$

$g(\bullet)$ represents the inequalities which are composed of lower and upper bounds constraints.

$$\begin{aligned} z^L \le z_m \le z^U \\ Tw_m \le Tw^U \\ \Delta P_m \le \Delta P^U \end{aligned} \tag{6}$$

Bottleneck problem of the optimization

For this practical and complex process, it is difficult to solve the optimization problem denoted by Eq. (1) directly. Thus, a SQP method based on infeasible path strategy (Biegler, 1982) has been used in this case. The original model is divided into two parts, optimization module and process simulation module.

Optimization module:

$$J = \max \sum_{m=1}^{M} y_m \tag{7}$$

s.t.

$$\begin{cases} g(z_m, Tw_m, \Delta P_m) \le 0 \\ m = 1,2,.....,M \end{cases}$$

Process simulation module:

$$\begin{cases} [y_m, Tw_m, \Delta P_m] = f_m(\theta, z_m, \delta_{C,m}, \sigma) \\ \delta_{C,m+1} = h(y_m, z_m) \\ \delta_{C,1} = 0 \\ m = 1,2,.....,M \end{cases} \tag{8}$$

The scale of optimization model is reduced. But, a serious bottleneck still exists. Because the model described above is very complex, the gradient cannot be calculated by analytic solution and the method of perturbation is used to get the gradient values. Since the coke layer thickness may affect the heat transfer and the pressure drop of the subsequent time horizons, the process simulation for one quasi-steady state (Eq. (8)) will be aggregately executed for *M(M+1)/2* times to accomplish a gradient calculation. For instance, the running period of an actual GK-V cracking coil that is studied in this paper is 42 days. If the length of the time interval is set to 3 days, *M=14*. In this case, 105 simulations must be run to accomplish a gradient calculation. Therefore, the total optimization process is greatly slowed down and it is very necessary to find a method to eliminate the bottleneck of computation.

Deserialization of the bottleneck

The parallelism exists in the excessive simulations. Because all perturbations to manipulated variables (COTs) are independent, the corresponding simulations are simultaneously realizable. This means that the columns of Jacobian matrix can be calculated in parallel. Each column of Jacobian matrix corresponds to a series of simulation calculation. For example, if the slight perturbation is added to z_i , after the corresponding simulations have been completed, the *i*th column of Jacobian matrix (Eq. (9)) can be acquired immediately.

$$\begin{pmatrix} \frac{\partial J_1}{\partial z_1} & \frac{\partial J_1}{\partial z_2} & \cdots & \frac{\partial J_1}{\partial z_M} \\ \frac{\partial J_2}{\partial z_1} & \frac{\partial J_2}{\partial z_2} & \cdots & \frac{\partial J_2}{\partial z_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial J_N}{\partial z_1} & \frac{\partial J_N}{\partial z_2} & \cdots & \frac{\partial J_N}{\partial z_M} \end{pmatrix} \tag{9}$$

It is obvious that the calculation of the 1st column of Jacobian matrix (Eq. (9)) is the most complicated, because here *M* simulations must be accomplished. So, at most the granularity of the tasks that can be refined to is *M* (simulations). Theoretically, through deserialization, the gradient computation duty may approximately be reduced to $\frac{1}{(M+1)/2}$ of the serial case. Thus, the proper distribution of gradient computation duty is the key factor which may greatly affect the performance of the parallel optimization algorithm.

Results and discussion

In our work, the running period of the practical GK-V cracking coil is 42 days. The increment of the time horizon is 3 days, thus the whole period is divided into 14 quasi-steady states. The lower and upper bounds of some variables are given here: z^L=1097K, z^U=1111K, Tw^U=1373K, ΔP^U=130KPa.

To exert the most power of parallel computing, we distributed the computational duties (Number of simulations, NS) to each computational node as averagely as possible. In fact, the total 105 simulations of quasi-steady state were distributed to each node averagely.

Table 1 lists the running time for solving the problem denoted by Eq. (1) with different numbers of computer nodes. As seen from Table 1, the total computation time for optimization problem is reduced distinctly.

Table 1. Computation time for optimization

Node Number	1	2	3	4	5	6	7
Running time (h)	7.11	3.74	2.45	1.88	1.50	1.29	1.09

The achievable speed-up and parallel efficiency is shown in Figure 1 and 2. As seen from Figure 1, an approximate linear speed-up is acquired through this parallel optimization scheme. Because the computational duties assigned to each node are not equal absolutely, the values of parallel efficiency in Figure 2 are less than a hundred percent. In addition, the communication and synchronization overheads in this case also slightly decrease the speed-up and parallel efficiency.

To study the influence of the computational load imbalance on the parallel performance, the computational duties are distributed to four nodes in different combinations as listed in Table 2.

According to the distributions listed in Table 2, The CPUmax, CPUavg, load imbalance, speed-up and parallel efficiency in different cases are listed in Table 3. It is clear that the load imbalance is the decisive factor for the performance of this parallel optimization scheme. While the load imbalance increases, the speed-up and parallel efficiency descend. As seen from Table 3 and Figure 2, if

the load imbalance is large, the speed-up of four nodes even is fewer than that of three nodes.

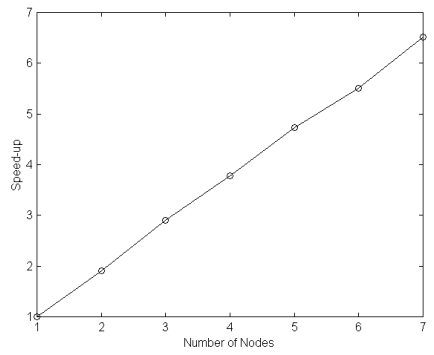

Figure 1. Approximate linear speed-up

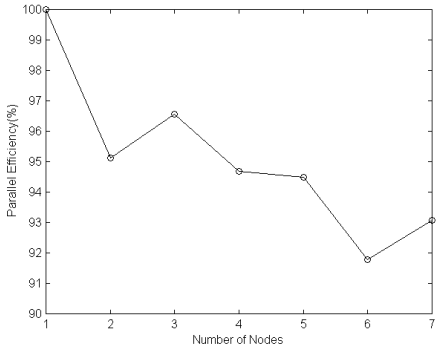

Figure 2. Relevant parallel efficiency

Table 2. Distributions of computational duties

	NS on node 1	NS on node 2	NS on node 3	NS on node 4
Case 1	27	26	26	26
Case 2	31	20	28	26
Case 3	35	18	26	26
Case 4	39	20	22	24
Case 5	45	21	19	20

Table 3. Influence of load imbalance on parallel performance

	CPU_{max} (h)	CPU_{avg} (h)	Load imbalance (%)	Speed up	Parallel efficiency (%)
1	1.580	1.456	8.47	3.79	94.68
2	1.756	1.451	21.01	3.53	88.34
3	1.944	1.455	33.65	3.24	80.91
4	2.172	1.455	49.24	2.93	73.27
5	2.573	1.462	76.07	2.51	62.86

Conclusion

Optimization for industrial-scale chemical processes on distributed memory multicomputers is feasible and can offer great savings in total execution time. A parallel optimization scheme for industrial steam cracking process was implemented on a cluster of workstations successfully.

The performance of parallel computing strategies is mainly limited by the parallelism of the system, the computational load balance, the message-passing and synchronization overheads. In spite of these overheads, it is still effective to solve the time-consuming simulation and optimization problems for its computational efficiency and flexibility. Cost-effective methods can be implemented on existing reliable networked workstations using sophisticated parallel software packages such as PVM.

Acknowledgements

This work was supported by National "863" program of China under grant number 2001AA413220.

References

Biegler, L. T., Hughes, R. R. (1982). Infeasible Path Optimization with Sequential Modular Simulators. *AIChE J.*, **28**, 6, 994-1002.

Froment, G. F., Bischoff, K. B. (1990). *Chemical Reactor Analysis and Design*; John Wiley & Sons: New York.

Heyndericks, G. J., Froment, G. F. (1998). Simulation and Comparison of the Run Length of an Ethane Cracking Furnace with Reactor Tubes of Circular and Elliptical Cross Sections, *Ind. Eng. Chem. Res.*, **37**, 914-922.

Kumar, P., Kunzru, D. (1985a). Modeling of Naphtha Pyrolysis, *Ind. Eng. Chem. Process Des. Dev.*, **24**, 3, 774-782.

Kumar, P., Kunzru, D. (1985b). Kinetics of Coke Deposition in Naphtha Pyrolysis. *Can. J. Chem. Eng.*, **63**, 8, 598-603.

Mallya, J. U., Stadtherr, M. A. (1997a). A Multifrontal Approach for Simulating Eqilibrium-stage Processes on Supercomputers. *Ind. Eng. Chem. Res.*, **36**, 1, 144-151.

Mallya, J. U., Zitney, S. E., Choudhary, S. (1997b). Parallel Frontal Solver for Large-scale Process Simulation and Optimization. *AIChE J.*, **43**, 4, 1032-1040.

Schools, E. M., Froment, G. F. (1997). Simulation of Decoking of Thermal Cracking Coils by Steam/Air-mixtures. *AIChE J.*, **43**, 1, 118-126.

Xu, Q., Chen, B. Z., He, X. R. A Fast Simulation Algorithm for Industrial Cracking Furnaces, *Hydrocarbon Processing*, accepted for publication.

Zitney, S. E., Brull, L., Lang, L., Zeller, R. (1995). Plantwide Dynamic Simulation on Supercomputers: Modeling a Bayer Distillation Process. *AIChE Symp. Ser.,* **91**, 304, 313-316.