Energy and Buildings 76 (2014) 92–101

Contents lists available at ScienceDirect

Energy and Buildings

journal homepage: www.elsevier.com/locate/enbuild

High throughput computing based distributed genetic algorithm for building energy consumption optimization

Chunfeng Yang [a,b,c], Haijiang Li [b,c,*], Yacine Rezgui [b,c], Ioan Petri [b,c], Baris Yuce [b,c], Biaosong Chen [a], Bejay Jayan [b,c]

[a] State Key Laboratory of Structural Analysis for Industrial Equipments, Department of Engineering Mechanics, Faculty of Vehicle Engineering and Mechanics, Dalian University of Technology, Dalian 116024, China
[b] School of Engineering, Cardiff University, UK
[c] BRE Institute of Sustainable Engineering, Cardiff University, UK

ARTICLE INFO

Article history:
Received 29 October 2013
Received in revised form 31 January 2014
Accepted 14 February 2014
Available online 4 March 2014

Keywords:
Simulation-based optimization
Building energy optimization
EnergyPlus
GA
Parallel
Distribute
HTCondor
SiPESC.OPT

ABSTRACT

Simulation based energy consumption optimization problems of complicated building, solved by stochastic algorithms, are generally time-consuming. This paper presents a web-based parallel GA optimization framework based on high-throughput distributed computation environment to reduce the computation time of complex building energy optimization applications. The optimization framework has been utilized in an EU FP7 project – SportE2 (Energy Efficiency for Sport Facilities) to conduct large scale buildings energy consumption optimizations. The optimization results achieved for a testing building, KUBIK in Spain, showed a significant computation time deduction while still acquired acceptable optimal results.

1. Introduction

The last decades have witnessed a substantial body of research directed towards simulation based building energy consumption optimization [1–12], however, such optimization applications are generally time consuming. The heavy time load is mainly caused by two factors, one, the lagging of software behind the computer hardware advancements; the other, the complexity of the building simulation models. It typically takes software 5–7 years to catch up after the new hardware launches [13]. The gap between what hardware made possible and what software achieved has been actually widening [14]. In spite of the computer hardware processing power improves at roughly exponential rates [15], sometimes, one run of a complicated building model, utilizing the high-power personal computer, still takes a few minutes or even several hours [2]. The advances of computational science and technology have enabled researchers to build and test simulation models of climate change, human activities, HVAC system performance and complex building's internal environments. The first generation building simulation software is based on analytical formulations with many simplifying assumption. The current generation simulates buildings' multi aspect nonlinear performances based on numerical methods of particle integrations [3]. The building models are discontinuous with respect to some facility parameters [6,16], for simulation software, such as EnergyPlus and TRSYS, uses adaptive algorithms and condition logic. Moreover, some of the parameters are discrete [17]. Because of the tolerance to discontinue functions, stochastic optimization algorithms, such as GA (genetic algorithm) [5–8] and PSO (Particle Swarm Optimization) [9,10], are widely used to solve optimization problems with such building simulation models. However, stochastic algorithms frequently require hundreds and thousands of runs of the simulation in order to reach optima result. Thus a large amount of time, sometimes days and weeks, is consumed to solve the simulation based building optimization problems [6,8,2,18].

Several techniques have been presented to reduce the computation time of solving simulation based building optimization problems with the stochastic algorithms. This paper only considered GAs which are a kind of stochastic search optimization algorithms based on principles of biologic evolution. In the optimization application with GAs, a population of candidate solutions

* Corresponding author at: School of Engineering, Cardiff University, UK.
E-mail addresses: yangchunfeng@mail.dlut.edu.cn (C. Yang), lih@cardiff.ac.uk (H. Li).

http://dx.doi.org/10.1016/j.enbuild.2014.02.053
0378-7788/

of the optimization problem is evaluated, according to its fitness value, and the best solutions are selected to generate new generation. Over a number of iterations, it is expected that good traits would dominate the population, thus optimal solutions would be found.

The first method is reducing population size and/or generation number of GAs [9]. Both of these parameters influence whether GAs can find optimal solutions. If the individuals are insufficient or the generation number is small, it would be difficult for GAs to generate good solutions.

The second method is using simplified simulation models instead of complete full scale simulation models [18]. This method raises a risk of over-simplification of modeling complicate building phenomenas. The third method is using surrogate models of the complex building simulation models [6]. This method has the same disadvantages as the second method has. The fourth method is using centralized mainframe or supercomputer. However, high capital costs and heavily oversubscription preclude access to these high-end computing resources.

Parallelization is another approach to reduce the computation time of the simulation based building optimization applications. The basic idea underlying the parallel computation is divide-and-conquer, which means dividing a complex computation task into chunks and solving the chunks simultaneously using multiple processors [19]. As to building optimization applications solved by GAs, hundreds of runs of building model simulations consume most of the their computation time. These simulations are independent with each other, and there is no need for communication during these processes. Thus it is naturally suit to be applied to parallelization.

This paper presents a web-based parallel GA optimization framework based on high-throughput distributed computation environment to reduce the computation time of complex building energy optimization applications. As the advances of computer science and information technology, thousands and millions of cheap personal computers are in use worldwide. The computational power of a school of personal computers as a whole exceeds that of a large centralized mainframe or supercomputer. Such distributed computation resources is an efficient and cost-effective platform for simulation based optimization applications with the paralleled GAs. The parallel optimization application has been enveloped as a reusable web service which is a promising service-oriented computing (SOC) technology. The SOC paradigm assembles loosely coupled services to support the development of flexible, dynamic business processes and agile distributive applications. The optimization service can be accessed worldwide through Internet, and it can also be integrated into other applications. The case study showed that this framework can reduce the computation time of building energy optimization process.

2. Related work

2.1. Parallel genetic algorithms

Optimization algorithms can be grouped into two categories: gradient-based algorithms and stochastic algorithms. The gradient-based optimization algorithms require the objective function of the optimization model to be continue. They are not suit to deal with optimization problems which have discontinuous objective functions and/or discrete variables. The stochastic algorithms do not have such smoothness requirements on the optimization models, and that's the main reason why they are widely utilized in solving simulation-based building energy consumption optimization problems.

Genetic algorithms (GAs), premised on theory of natural evolutions and gene, are stochastic search optimization algorithms. In optimization application with GAs, a population of candidate solutions of the optimization problem is evaluated. Every individual of the population has a fitness value which is the indicator of the goodness of the candidate solution. Generally, the fitness value is the objective of the optimization model. The best solutions are selected, according to their fitness values, to generate a new generation of candidate solutions. Over a number of generations, it is expected that good traits would dominate the population, thus optimal solutions would be found.

Although GAs can find a near optimal solution in a reasonable time, one disadvantage of the GAs, as well as other stochastic algorithms, is that they often require hundreds times of evaluations of individuals' fitness values to find adequate solutions for complex optimization problems [6]. In regard to simulation based building optimization applications where the fitness value is evaluated through the time-consuming building simulation, the large amount of time consumed by hundreds of runs of complex building simulation model is a unbearable burden. However, the fitness values of the individuals are independent with each other in GAS, and there is no communication during the fitness evaluation process. In simulation based optimization applications, the time-consuming building simulation can run simultaneously in multiple processors, and a large amount of time would be saved. Thus, parallelization is a promising method to reduce the computation time of optimization applications with the GAs.

Parallelization can be applied to GAs in several different ways. According to the population's topology, parallel GAs can be classified into three categories [19]: global single population master-slave GAs, single population fine-grained GAs and multiple populations coarse-grained GAs. The schemes of these parallel GAs are shown in Fig. 1. The master-slave parallelization method inherits the GAs' behaviors, while the other two methods do not have this advantage. In the master-slave category, a parallel genetic algorithm is synchronous when its master process stops to wait fitness values of all individual slave processes before proceeding the next generation, as in Fig. 2, otherwise it is an asynchronous parallel GA.

2.2. High-through computation environment

The advances of computational science enable researchers to explore complex problems that are otherwise impractical or impossible to address [20]. In many research and engineering projects, a computational environment that provides huge amount of computational power during a long period of time is needed [21]. Years ago, such computational powers were provided by large, centralized mainframes or supercomputers. This kind of computational environment is not convenience to use because of its high utilization costs and heavily oversubscription. According to the Moore's law [15], the capabilities of computer hardware double approximately every two years; computers have become faster, cheaper and smaller. Personal computers today have more processing power and storage space than the supercomputers of last century have. Billions of personal computers are in use around the world. The personal computer is slower than the centralized machine, but total computational power of a school of personal computers and workstations as a whole exceeds that of a large centralized computer.

HTCondor is an open-source high throughput computing (HTC) workload management software framework for a cluster of distributed computer resources [21]. It consists of a set of software tools which implement and deploy high throughput computing on distribute computers. Distribute computing powers can seamlessly integrate through HTCondor into one computing environment for