Future Generation Computer Systems 62 (2016) 1–16

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Distributed scheduling with probabilistic and fuzzy classifications of processes

Susmit Bagchi

Gyeongsang National University, Department of Aerospace and Software Engineering (Informatics), 660-701 Jinju, Republic of Korea

HIGHLIGHTS

- Distributed scheduling with process classification.
- Distributed scheduling with probabilistic and fuzzy classified processes.
- Distributed scheduling and scheduler activation with process classification.

ARTICLE INFO

Article history:
Received 2 January 2016
Received in revised form
26 February 2016
Accepted 6 March 2016
Available online 17 March 2016

Keywords:
Distributed systems
Scheduler
CPU-bound
IO-bound
Fuzzy logic
Kernel

ABSTRACT

The classification of various types of processes is an important factor in large-scale distributed systems such as, grid and cloud platforms. Moreover, the coordination and control of distributed processes are research challenges in presence of unpredictable network partitioning and distributed semaphores. The process classification is important in order to allocate and schedule distributed processes enhancing overall resource utilization and throughput. The schedulers employ patterns of resource affinities of concurrent processes in order to make scheduling decisions affecting overall resource utilization in a system, where resource affinity patterns of a process may not be static. This paper proposes an estimation model and a classifier algorithm to queuing processes based on respective resource affinities. The kernel-level software architecture is designed to control scheduling of distributed processes based on classification for enhanced throughput. The classifier algorithm tracks the resource affinities of processes based on execution traces and the control algorithm performs process scheduling. Experimental results indicate that the classifier algorithm successfully manages process queues based on resource affinities of processes and, the control algorithm successfully monitors scheduler activation for a set of processes.

1. Introduction

The applications of distributed systems are pervasive ranging from grid/cloud systems to distributed embedded sensing applications. The large-scale distributed systems such as, grid and cloud platforms, are comprised of heterogeneous processes having varying resource affinities. Moreover, the distributed computing applications need to adapt to changing execution environments needing distributed inter-process coordination [1,2]. The heterogeneous large-scale distributed systems need low-level (system level) knowledge about coordination and control of state of processes to achieve optimal resource utilization and synchronization [3–5]. The designing of appropriate scheduling of process-load as well as control of execution is important to achieve power consumption optimizations and providing quality of services (QoS) [6].

E-mail address: susmitbagchi@yahoo.co.uk.

http://dx.doi.org/10.1016/j.future.2016.03.001

Appropriate classification of distributed processes is necessary to design optimal resource allocation and scheduling of distributed processes at nodes. Researchers have proposed a middleware based approach to realize resource monitoring and coordination of isolated long-running processes in grid [7]. However, the assumptions about process isolation break down in presence of synchrony. Although middleware based distributed resource monitoring is widely adapted, however kernel-level monitoring is more effective in the presence of QoS requirements of applications [5,8].

In general, heterogeneous distributed processes can be classified into two broad categories namely, CPU-bound processes and IO-bound processes. A CPU-bound process exhibits frequent and long CPU bursts with fewer IO calls. On the contrary, an IO-bound process rests in the wait-state often with short CPU-bursts and frequent IO bursts. In general, the stubbing mechanism and heuristics are used to determine resource affinity of a process with approximate measure [9]. In order to measure the resource consumption patterns of transactional processes under concurrent executions, the resource modeling and performance analysis tools are designed [10,11]. However, a concurrently executing process may not exhibit static as well as deterministic locality of execution through out its lifetime uniformly. In addition, the resource allocation frequency and affinity may change from time to time depending on the application logic and dynamics of execution environments.

This paper proposes a novel estimation model based on probability and fuzzy logic to estimate resource affinity pattern of a process. The corresponding process classifier and scheduling-control algorithms are designed to monitor process queues and making scheduler activations. The system level software architecture is designed to realize distributed scheduling and monitoring of processes. The scheduling-control algorithm employs process classifier to monitor and control processes according to their affinity patterns. The experimental results illustrate that the classifier and control algorithms accurately classify the processes into different process queues as well as implements better schedule.

1.1. Motivation

Distributed computing applications are heterogeneous in nature ranging from large scale cloud/grid computing systems to distributed embedded systems involving sensor devices [12,13]. These distributed and pervasive heterogeneous applications require optimal resource allocation and, strong coordination mechanisms between processes as well as adaptive scheduling [1]. The existing scheduling and coordination mechanisms employ deterministic application-logic, which makes the distributed systems architecturally rigid and hardly scalable [1]. In the large-scale distributed computing systems, variations of heterogeneous hardware/software resources affect overall performance of the systems and, accordingly various scheduling algorithms are designed depending upon resource requirements of distributed processes [14]. In general, the parameter based process classifiers are employed to design computing systems offering enhanced performance. The classifications of processes are used to realize distributed scheduling and, to design real-time algorithms, where a process can have transition to wait-state due to disk IO without releasing associated semaphore [15]. Furthermore, the virtualized computing systems monitor resource affinity of processes to analyze overall system performance [16]. The fine tuning of a system parameter to enhance resource utilization and, avoidance of livelock are difficult without appropriate classification of processes based on their resource affinities. The modeling of IO affinity and interference in data-intensive distributed applications are required to avoid degradation of system performance due to existence of multiplexed IO [17,18]. The task classification schemes can be employed to distribute tasks at several nodes based on resource requirements to save power and reduce mutual interference [19].

The online transactional and concurrent processes often employ training-data points to classify resource affinity of processes [10]. However, the static profiling of processes is inadequate and cannot be employed in large-scale distributed computing systems comprised of heterogeneous concurrent processes [15,9,17,10,11]. Thus, an online estimation of resource affinity of processes and probabilistic classification of processes with high accuracy are necessary steps to consider. The effective control of scheduling of distributed processes in a node requires the classifier output. Moreover, the unpredictable network partitioning, delays and distributed semaphores would lead to execution failures, process blocking and restart due to coordination failure. The better coordination and resource management between distributed processes can be achieved if the remote process monitoring and scheduling control architecture can be realized in systems level [5,20]. This is because, the systems level or low-level knowledge about state of distributed processes helps in realizing better synchrony and reliability [20].

This paper proposes a novel online process classifier based on probabilistic and fuzzy estimation of resource affinities of processes depending on execution and scheduling traces. The fuzzy logical refinements are applied to detect transitory affinity patterns. The kernel-level software architecture and scheduler control algorithm are designed based on the output of process classifier. The main contributions of this paper are as follows.

- A probabilistic estimation model is designed to estimate resource affinity patterns of processes.
- Designing of kernel-level software architecture to control of local scheduling of distributed processes based on process classifier algorithm.
- Application of fuzzy logical refinement to model fine-grained dynamics of processes.
- Designing of algorithm to control local scheduling of distributed processes based on classifier output.

Rest of the paper is organized as follows. Section 2 describes related work. Section 3 describes the probabilistic and fuzzy estimation models. Section 4 describes designed software architecture and scheduler activation mechanism. Section 5 presents process classifier and scheduling-control algorithms. Section 6 describes experimental evaluations in details. Section 7 presents comparative analysis of different algorithmic performances. Lastly, Section 8 concludes the paper.

2. Related work

The distributed and pervasive applications require strong coordination between executing processes as well as adaptive scheduling [1,12]. In the distributed systems, the coordinating entities are the distributed processes having instantaneous state of execution, allocated resources and, assigned scheduling parameters [21]. In general, the distributed monitoring and adaptation mechanisms employ centralized local control or distributed control-loops within systems [22].

On the other hand, the classification of processes in a distributed computing system is employed to determine the resource consumption and performance as well as scheduling of concurrent processes at different nodes. Researchers have proposed the designing of task classifiers based on static allocation and task migration between nodes in real-time systems [19]. A real-time scheduling algorithm is designed based on classification of CPU-bound and IO-bound processes in a system [15]. The algorithm considers a ratio-based estimation of CPU affinity of tasks and, accordingly classifies other IO-bound tasks. However, the direct ratio-based classification may not be appropriate in highly dynamic execution environments. In the large scale distributed computing systems, such as cloud/grid platforms, the application-parameter based scheduling of jobs are important to maintain QoS. It is proposed that distributed job scheduling algorithms can classify processes or jobs based on various parameters such as, energy requirements and, execution speed requirements [6]. However, the scheduling algorithm is considered to be not optimistic in nature and does not consider other classifying parameters based on dynamic execution traces.

The execution localities and resource affinities of concurrent processes having data dependencies may change unpredictably and cannot be determined statically. The virtual machine based computing systems employ resource affinity estimation of processes in order to monitor and analyze overall performance [16]. It is observed in the virtual machine profiling that, the CPU-bound processes and IO-bound processes control the overall system performance very differently [16,23]. The load of CPU-bound processes executing on virtual machines does not scale linearly with