VIRTUALIZATION BASED SCHEDULING ALGORITHM FOR PARALLEL DISTRIBUTED COMPUTING

K.LAKSHMAJI
Dept. of IT
Sir C.R.Reddy college of Engg,
Eluru,India
Email: kotlalakshmaji@gmail.com

A.M.K.KANNA BABU
Dept. of IT
Sir C.R.Reddy college of Engg,
Eluru, India
Email: kumarkanna1@gmail.com

N.PRASAD
Dept. of IT
Sir C.R.Reddy college of Engg,
Eluru, India
Email: nprasad9999@gmail.com

ABSTRACT

In this paper, we design a new virtual queue-based back-pressure scheduling algorithm (VBR) for achieving significant delay reduction in parallel distributed computing. Our algorithm design comes from an observation that classical back-pressure scheduling algorithm usually needs a long period of time to form a queue backlog-based gradient in a network, which decreases towards the sink in the network, before achieving stable packet delivery performance. To address this issue, VBR is designed to pre-build proper virtual queue-based gradient at nodes in a parallel distributed computing, which is chosen to be a function of traffic arrival rate, link rate, and distance to sink, in order to be adaptive to different network and application environments while achieving high network performance. Moreover, the queue backlog differential between each pair of neighbor nodes is decided by their actual queue lengths and also their virtual queue lengths (gradient values). We prove that VBR can maintain back-pressure scheduling's throughput optimality. Simulation result shows that VBR can obtain significant performance improvement in terms of packet delivery ratio, average end-to-end delay, and average queue length as compared with existing work.

Keywords: *Back-pressure scheduling Delay reduction parallel distributed computing*

1. INTRODUCTION

The popularity of the Internet and the availability of powerful computers and high-speed networks as low-cost commodity components are changing the way we use computers today. These technical opportunities have led to the possibility of using geographically distributed and multi-owner resources to solve large-scale problems in science, engineering, and commerce. Recent research on these topics has led to the emergence of a new paradigm known as parallel distributed computing[6],.To achieve the promising potentials of tremendous distributed resources, effective and efficient scheduling algorithms are fundamentally important. Unfortunately, scheduling algorithms in traditional parallel and distributed systems, which usually run on homogeneous and dedicated resources, e.g. computer clusters, cannot work well in the new circumstances. In this paper, the state of current research on scheduling algorithms for 2 the new generation of computational environments will be surveyed and open problems will be discussed. The remainder of this paper is organized as follows. An overview of the parallel distributed scheduling problem is presented in Section 2 with a generalized scheduling architecture. In Section 3, the progress made to date in the design and analysis of scheduling algorithms for Grid computing is reviewed. A summary and some research opportunities are offered in Section

$$Area = \int_a^b f(x)\,dx = \lim_{N \to infinity} \frac{1}{N} \sum_{i=1}^{N} f(r)\,(b-a)$$

This procedure efficiently calculates a one-variable integral using the above formula where r is a random input to f.

2. PARALLEL DISTRIBUTED COMPUTING

A parallel implementation adds the following code to split the problem over all available nodes and send the partial results back to node 0. Note that here the head node, 0, is used to accumulate the results, and does not actually do any computations.

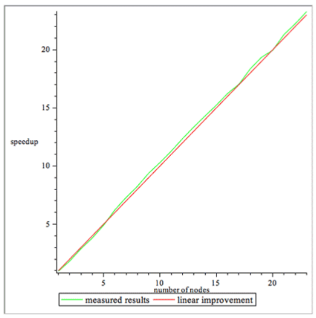
```
parallelApproxint := proc( expr, lim::name=numeric..numeric, { numSamples::integer := 1000 }
 )
   uses Grid;
   local me, numNodes, r, n;

   me := MyNode();
   numNodes := NumNodes();
   n := trunc(numSamples/numNodes);
   r := approxint( expr, lim, numsamples = n );
   printf("Node %d computed result %a using %d samples\n", me, r, n );

   if me = 0 then
      r := (r + add( Receive(), i=1..numNodes-1 )) / numNodes;
   else
      Send(0,r):
   end if;
end proc:
```

Execution times are summarized as follows. Computations were executed on a 3-blade cluster with 6 quad-core AMD Opteron 2378/2.4GHz processors and 8GB of memory per pair of CPUs, running Windows HPC Server 2008 and Maple 15.

Number of Compute Nodes	Real Time to Compute Solution	Performance
1 (using serialized code)	356.25 s	speedup is a measure of $\frac{T_1}{T_p}$ where T_1 is the execution time of the sequential algorithm and T_p is the execution time of the parallel algorithm using p processes.
1 (using grid)	369.18	
2	194.63	
3	123.12	
4	93.57	
5	72.67	
6	57.22	
7	48.80	
8	43.15	
9	37.96	
10	34.60	
11	31.58	
12	28.83	
13	26.66	
14	24.90	
15	23.38	
16	21.93	
17	20.92	
18	19.41	
19	18.41	
20	17.85	
21	16.71	
22	16.02	
23	15.30 s	



3. VIRTUAL BASED SCHEDULING ALGORITHMS

In this section, we first introduce the scheduling mechanism in classical back-pressure algorithm. Then, we present the observation which motivates this work. Finally, we propose the algorithm design of VBR.

3.1 Classical back-pressure scheduling

The back-pressure scheduling algorithm was proposed in [1], and it works as follows. Assume time is slotted. At the beginning of a time slot *t*, for each link (*m*,*n*) in the network, the weight associated with the link (i.e., link weight) is assigned as the maximum flow weight (i.e., the maximum backlog differential of all the flows passing through the link, ties broken arbitrarily):

$$\begin{array}{@{}rcl@{}} W_{mn} (t)=\max\nolimits_{f:(m,n)}\left[U_{m}^{f} (t)-{U_{n}^{f}} (t)\right]. \end{array}$$

(2)

Packets belonging to flow *f* will be transmitted over link (*m*,*n*) if the link (*m*,*n*) is to be activated under a schedule $\pi(t)$, which is derived based on the following optimization function:

$$\begin{array}{@{}rcl@{}} \pi(t)=\arg\max\nolimits_{\pi\in\Gamma}{\sum\nolimits}_{(m,n)}W_{mn} (t)r, \end{array}$$

(3)

where Γ represents the set of all feasible schedules according to link interference model. Specially, since in this paper, we assume that all links have the same rates, and thus, the chosen schedule maximizes the sum of link weights among all schedules.

In back-pressure routing, when a packet leaves the current node where it stays and to which neighbor it will depart depends on the following: a) ranks of the queue length differences between the two ends of the outgoing links of the current node in the entire network; b) link rate; and c) whether there exists a link having been activated by the scheduling process in the interference range of a particular outgoing link. Moreover, all these factors may change with time (slots).

3.2 Algorithm design and analysis

In this subsection, we propose VBR. In VBR, the calculation of virtual queue-based gradient considers flow arrival rate, link capacity, and distance to sink. Furthermore, the pre-established gradient is expected to have the following property: The farther a network node is away from the sink, the higher the gradient difference between neighbor nodes with different distances to sink is. In this way, packets remote to the sink are expected to take the direction of shortest paths while packets in the vicinity of the sink may take alternate paths due to the reduced gradient difference.

Based on the above considerations, in VBR, the gradient for a node $m\in V(G)$ is calculated as follows.

$$\begin{array}{@{}rcl@{}} {Q_{m}^{f}}=a\times b^{\lambda/{H_{m}^{f}}}\times c^{{H_{m}^{f}}} \times r, \end{array}$$

(6)

where *a*, *b*, and *c* are network parameters and needed to be tuned via experiments. Next, we explain each component in (6) as follows: $b^{\lambda /{H_{m}^{f}}}$ is an indicator for measuring the influence of packet arrival rate at a node *m* with ${H_{m}^{f}}$ hop distance to sink. Here, the main reason that we divide the arrival rate by hop distance to sink is to increase the impact of arrival rate at nodes closer to the sink and especially increase the gradient difference between sink's one-hop neighbor nodes and the sink itself, for which the reason is as follows, as we know, the vicinity of sink node in a WSN is often the bottleneck of the entire network. A large gradient difference between sink's one-hop neighbor nodes and the sink itself can encourage sink's neighbors to transmit to the sink so as to better use the limited capacity in sink's neighborhood. The item $c^{{H_{m}^{f}}}$ enables the gradients at nodes increases exponentially with hop distance to sink. The item *r* is the link rate, and its introduction in (4) is to enable good adaptability to networks with different link rates. In addition, we have also tried many other functions for the virtual queue-based gradient calculation. The function given in (4) exhibits the best performance.

Next, we theoretically prove that VBR is throughput optimal as do classical back-pressure algorithm and also the EDR algorithm (proof for the latter can be found in [5]). Since our algorithm shares the same scheduling paradigm with EDR, i.e., (2) and (3), thus, the following lemma is given for proving the throughput optimality of VBR.

4. CONCLUSION AND FUTURE SCOPE

The parallel distributed computing model enables users to use services as utilities. It allows users to access these services over the Internet and make payments according to the usage and level of quality requested by the user. In this research work, we have proposed a score based deadline constraint virtualization scheduling algorithm that achieves less execution time along with manageable execution cost under the user specified deadline constraint. We also compared it with without score deadline constraint scheduling algorithm. We are also trying to extend our work to support real time.

4. REFERENCES

1. L Tassiulas, A Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Trans. Autom. Control. 37, 1936–1948 (1992).

2. A Eryilmaz, R Srikant, in *INFOCOM'05: Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societie*s. Fair resource allocation in wireless networks using queue-length based scheduling and congestion control (IEEEMiami, FL, 2005), pp. 1794–1803**.**

3. MJ Neely, E Modiano, C Li, in *INFOCOM'05: Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societi*es. Fairness and optimal stochastic control for heterogeneous networks (IEEE,Miami, FL, 2005), pp. 1723–173**4.**

4. A Dvir, AV Vasilakos, in *SIGCOMM'10: Proceedings of the ACM SIGCOMM 2010 Conference*. Backpressure-based routing protocol for DTNs (ACM,New Delhi, INDIA, 2010), pp. 405–406**.**

5. L Georgiadis, MJ Neely, L Tassiulas, *Resource Allocation and Cross-layer Control in Wireless Networks* (Now Publishers Inc, Boston, 2006**).**

6. Gottlieb, Allan; Almasi, George S. (1989). *Highly parallel computing*. Redwood City, Calif.: Benjamin/Cummings. ISBN 0-8053-0177-1.

Mr. Lakshmaji Kotla is working as an Asst. Professor, in IT Department, Sir C. R. Reddy College of Engg, Eluru, A.P., India. He has received his M.Sc(CS) from Gowri P.G College Visakhapatnam and M.Tech (CSE) from Vasavi Engineering college Tadepalligudem (JNTUK University), Kakinada, A.P., INDIA. His research interests include Image Processing, Data Mining, CryptoGraphy & Network security, Biometrics, Cloud Computing and Software Engineering..

Mr.A.M.K.KANNA BABU is working as an Asst.Professor, in IT Department, Sir C. R.Reddy College of Engg, Eluru, A.P., and India. He has received his B.Tech(CSE) from Sir C R Reddy College of Engineering, Eluru, and M.Tech. (SE) from Avanthi Institute of Engineering and Technological. Makavarapupalem.A.P., INDIA. His research interests include Image Processing, Data Mining, Networks Security, Web security, Software Engineering, Computer Networks and Wireless Networks.

Mr. N. Prasad is working as an Asst. Professor, in IT Department, Sir C. R. Reddy College of Engg, Eluru, A.P., and India. He has received his B.Tech (CSE) from SVH College of Engineering, Machilipatnam and M.Tech (CSE) from Jawaharlal Nehru Technological College of Engg.Hyderabad.(JNT University),Hyderabad, A.P., INDIA. His research interests include Data Mining, Networks security, Web security, WSNs and Computer Networks.