Distributing Dynamic Divisible Loads

Ming Zeng and Viktoria Fodor
Department of Network and Systems Engineering
School of Electrical Engineering, KTH Royal Institute of Technology
Stockholm, Sweden
{mzeng, vfodor}@kth.se

***Abstract*—With the emergence of computing infrastructures in the cloud or at the network edge we need to address the question of how to utilize these shared resources when computational tasks are generated dynamically. While small computing tasks may be satisfied with the computing capacity of a single resource, large tasks may want to utilize multiple computing nodes and perform parallel processing to shorten the task completion time. In this paper we evaluate how additional overhead in such divisible load systems affect the efficiency of parallel processing - from the point of view of the task itself, and for the entire resource sharing system. We show that the preference of a single task may be in conflict with the allocation needed for a social optimum, which in turn depends heavily on the load as well as on the system size.**

I. Introduction

Envisioned smart city, intelligent transportation and mobile cloud computing applications often consider shared computing resources that are available for the applications' computational tasks [1]. These resources can be provided by an infrastructure, like road side units in vehicular area networks [2], or by the community of computationally powerful mobile nodes [3]. As the resources are available for many applications or users, the question of optimal resource sharing emerges.

If computational tasks are not divisible, then the question is which resource among the many to use. However, many of the complex tasks form divisible computational loads, like visual analytics [4] or large scale graph processing [5]. The possibility of parallel processing leads to a large body of research on optimal task scheduling [6][7][8], considering the instantaneously available computing resources. Clearly, if the division of the task does not lead to any overhead, each task, as well as the community benefits from using as many resources as possible. There seems to be no common understanding however on how much resources a single task should use if dividing the load has some additional cost.

In this work we evaluate the efficiency of load division when computational tasks arrive dynamically, and the division of a computational task requires additional overhead, such as the preprocessing of the input data, or the exchange of partial results. We aim at reaching a basic understanding of the following question: what is the optimal resource allocation, if the average performance of the distributed computation system needs to be optimized? How much resource would a single tasks request if it were aiming at minimizing its own task completion time? What would be the resulting system performance in this case? How do the optimal operating points depend on the overhead of load distribution? Since these distributed computing systems are likely to serve large communities, we are also interested in the asymptotic system behaviour at constant system utilization level.

ISBN 978-3-901882-94-4 © 2017 IFIP

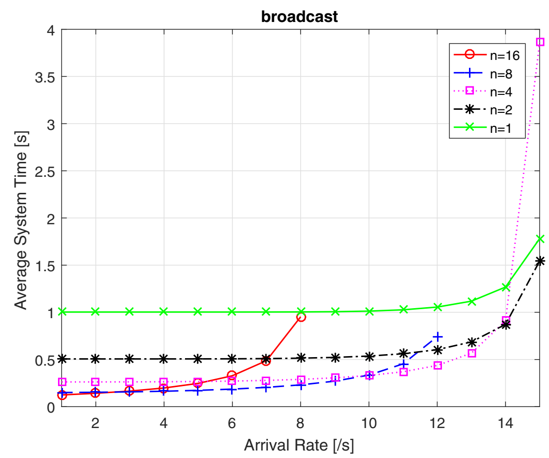

Fig. 1. Average system time, broadcast model. $L = 1$, $p = 1$, $\sigma = 0.003$, $m = 16$.

II. System description and problem formulation

We consider a computational resource sharing system, where computational tasks arrive dynamically, and share a number of processing nodes. When all nodes are busy, the tasks wait in a shared virtual queue. The tasks are divisible, but parallel processing may require overhead. Specifically, we consider three types of overheads:

- **C**onstant overhead: for computational tasks with a start-up phase that needs to be performed at each node, but is independent from the number of utilized nodes.
- **B**roadcast communication overhead: for tasks, where information exchange is required among the nodes, or to a central unit, one node can transmit at a time, but the information reaches all relevant nodes.
- **U**nicast communication overhead: for similar tasks, but considering the case when broadcast communication is not possible, and information needs to be transmitted from all nodes to all other nodes separately.

We start by considering a simplified model of a computational resource sharing system, that allows tractable mathematical analysis while gives insights in the challenges of the optimal resource allocation. We consider that the computational

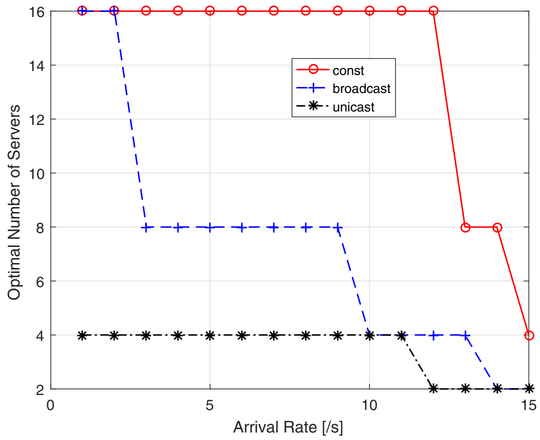

Fig. 2. Optimal number of servers for three overhead models.

system has m processing nodes, each with computational power p instructions per time unit. Tasks wait in a FIFO queue, and when at the head of the queue, they access n out of the m servers. The size of the tasks is exponentially distributed with a mean of L instructions, and we consider an overhead proportional to L with parameter δ. The computational requirement of the task is known, and the load is divided ensuring that the sub-tasks are completed at the same time. Consequently, the computational time at a server is still Exponentially distributed with a mean of $\overline{x}_C(n) = \frac{L}{p}\left(\frac{1}{n} + \sigma\right)$, $\overline{x}_B(n) = \frac{L}{p}\left(\frac{1}{n} + n\sigma\right)$ and $\overline{x}_U(n) = \frac{L}{p}\left(\frac{1}{n} + (n)(n-1)\sigma\right)$ for the three types of overhead. As the task occupies a block of n servers, $\lfloor m/n \rfloor$ tasks can be processed at a time, which results in a M/M/$\lfloor m/n \rfloor$ system model

We address two scenarios: in the first scenario the number of allocated servers n is selected such that the average task completion time is minimized; in the second scenario we consider greedy users that want to minimize the completion time of their own tasks, that is, once at the head of the queue, tasks occupy a number of servers, such that the computational time $x(n)$ is minimized.

III. Numerical results

Fig. 1 shows the average system time for the broadcast overhead model, as the load of the computing system, tuned by the task arrival intensity λ, is increased, and for different n values. The figure shows that under low load the system benefits from allocating many servers for the tasks, but this changes as the load gets higher, and at very high loads tasks would end up using only a few servers to avoid the overhead that increases the system utilization. We compare the optimal number of allocated servers, that minimises the system time, on Fig. 2 for all three models. As we see, the optimal value depends significantly on the load as well as on the type of the overhead.

Fig. 3 compares the optimal number of servers with that allocated by greedy users, as the system size m increases. Constant overhead allows the increase of n together with m, however, in the broadcast and uniform scenarios it becomes

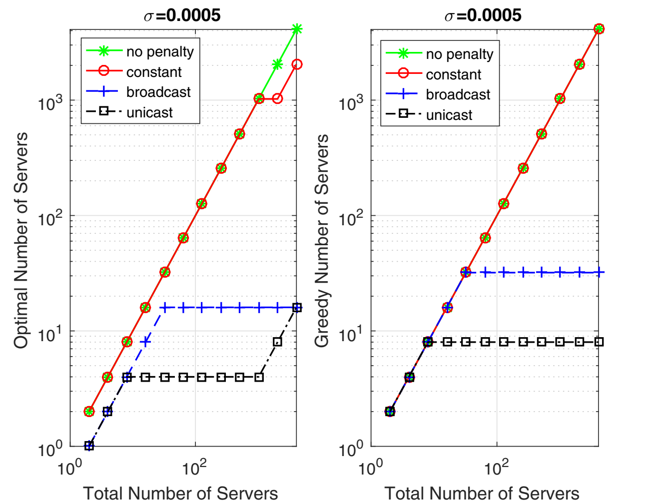

Fig. 3. Optimal and greedy number of servers under increasing m, with constant, broadcast and unicast overhead, for $L = 1, p = 1, \lambda/m = 0.25$.

limited by the introduced overhead. Note that a greedy system divides the load into too many servers, increasing this way the average system time.

IV. Conclusions

We have evaluated the optimal level of load division in dynamic resource sharing systems with divisible loads and different types of overheads. We can conclude that the optimal level of load depends on the system size, the system load and the type of overhead. Moreover, we see that minimizing the service time of the tasks does not lead to minimized task completion time in dynamic resource sharing systems, and thus additional system level control is required to achieve a social optimum.

References

[1] S. Vatsikas, G. Kalogridis, T. Lewis, and M. Sooriyabandara, "The experience of using the ies cities citizen-centric iot platform," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 40–47, February 2017.

[2] J. Barrachina, P. Garrido, M. Fogue, F. J. Martinez, J. C. Cano, C. T. Calafate, and P. Manzoni, "Road side unit deployment: A density-based approach," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 30–39, Fall 2013.

[3] I. Yaqoob, E. Ahmed, A. Gani, S. Mokhtar, M. Imran, and S. Guizani, "Mobile ad hoc cloud: A survey," *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2572–2589, 2016.

[4] A. Redondi, M. Cesana, M. Tagliasacchi, I. Filippini, G. DÃąn, and V. Fodor, "Cooperative image analysis in visual sensor networks," *Ad Hoc Networks*, vol. 28, pp. 38 – 51, 2015.

[5] Y. Zhang, Q. Gao, L. Gao, and C. Wang, "Maiter: An asynchronous graph processing framework for delta-based accumulative iterative computation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2091–2100, Aug 2014.

[6] Y. H. Kao, B. Krishnamachari, M. R. Ra, and F. Bai, "Hermes: Latency optimal task assignment for resource-constrained mobile computing," *IEEE Transactions on Mobile Computing*, 2017.

[7] G. Kesidis, Y. Shan, B. Urgaonkar, and J. Liebeherr, "Network calculus for parallel processing," *SIGMETRICS Perform. Eval. Rev.*, vol. 43, no. 2, pp. 48–50, 2015.

[8] E. Eriksson, G. Dan, and V. Fodor, "Radio and computational resource management for fog computing enabled wireless camera networks," in *IEEE Globecom Workshops (GC Wkshps)*, Dec 2016.