International Journal of Recent Trends in Engineering & Research
ISSN(ONLINE) : 2455-1457
IMPACT FACTOR : 3.344

A DISTRIBUTED MODIFIED BIN BACKING FOR COMPUTING INFRASTRUCTURE USING SMARTPHONES

Ms.Rajaambika[1], P.M Anusree[2], P. Dinesh Kumar[3] and J. Santhosh Kumar[4]
[1,2,3,4]Department of CSE, Kathir College of Engineering,

Abstract-Daily night , so many smartphones are connected into a power socket for recharging the battery. Given the increasing computing efficiency of smartphones, these idle phones constitute a sizeable computing infrastructure which says that, an enterprise can tackle the combined computing power of such smartphones, to construct a distributed computing infrastructure. Such an infrastructure has the potential to reduce both the capital expenses and energy costs incurred by enterprises. Introduce a new framework CWC, which stands for computing while charging. CWC uses a server that connected to the Internet, for job scheduling on the smartphones and collecting the outputs from the computations. The scheduling algorithms on the server are lightweight, and thus, a rudimentary low cost PC will suffice. The scheduling algorithm should have the capacity to maintain the job working set which utilizes all resources available. The root of this scheduling issue is a d-capacity bin-packing problem where every system resource is represented by a capacity in the bin and the requirements of each waiting job are denoted by the d-capacities of an item in the input list. Implement and evaluate the model of CWC that handle a scheduling algorithm to minimize the make span of a set of computing tasks. Our CWC's scheduler produces a makespan which is 1.6x faster than other scheduling approaches by evaluating 8 android phones.
Keywords*:*Distributed Computing, Mobile computing, Bin Packing Algorithm, Distributed Task Scheduling,

I. INTRODUCTION

Recent survey reports that 66 percent of surveyed organizations supply their employees with smartphones for various reasons. For example, Novartis handed out smartphones to its employees to manage e-mails, calendars and health related information; Lowe's did so for its employees to have real-time access to key product information and to allow managers to handle administrative tasks. In this work argue that in such settings, an enterprise can harness the aggregate computing power of such smartphones, to construct a distributed computing infrastructure. Such an infrastructure has the potential to reduce both the capital expenses and energy costs incurred by enterprises. Due to recent advancements in embedded processor design, smartphone CPUs can now compete with desktop CPUs for raw computational power . If enterprises could exploit the smartphones handed out to their employees to run some portion of their workload, it is conceivable that the cost of their computing infrastructure could have been reduced. More importantly, enterprises could save energy (by shutting down their servers) when they offload certain tasks to smartphones, since smartphone CPUs consume much less power than server CPUs [16-17].

In fact, there are already proposals for ARM-based data centers to harness the energy efficiency of embedded processors. Besides its potential benefits, realizing such a smartphone computing infrastructure faces a number of challenges.

Seek to articulate these challenges and build a framework to make such a platform viable. In particular, the biggest obstacles in using smartphones for computing are the battery life and bandwidth. If a smartphone is utilized for heavy computing when in use by its owner, the battery may drain and render the phone unusable. Further, sending large volumes of computing data to the phone using cellular is not practical since data usage is typically capped by carriers. Thus envision using smartphones when they are being charged at night, when active use by phone owners is not likely. Moreover, the phones will be stationary and will likely connect to Wi-Fi in owners' homes; this will reduce bandwidth fluctuations and allow the transfer of computing data to/from the phone at no cost.

II. PROPOSED SYSTEM

We propose new Job Scheduling algorithm (Bin Packing Algorithm) for CWC: Distributed Computing Using Smartphones. The methodology Proposed as follows

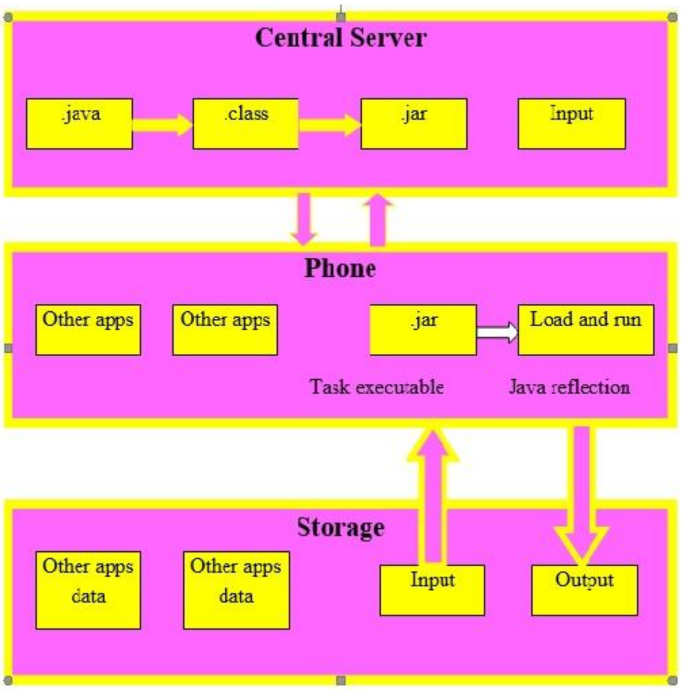
Methodology

The list scanning process produces the key algorithm structure for new multi- capacity aware bin-packing algorithms. The main differences between the new algorithms and the FF algorithm are the criteria used to select the next item to be packed into the current bin. Whereas FF requires only that the item fits into the current bin, the multi-capacity aware algorithms, uses heuristics to select items which attempts to correct a capacity imbalance in the current bin. A capacity imbalance is defined as the condition; in the current bin essentially, at least one capacity is fuller than the other capacities. The general theme is that if the capacities are all kept balanced, then more items will likely t into the bin. A simple heuristic algorithm accompanies from this theme. Consider a bin capacity vector in which is the component which is currently filled to a lower capacity than all other components. A lowest-capacity aware packing algorithm searches the list L looking for an item which fits in bin and in which Xij is the largest resource requirement in. Adding item to bin heuristically lessens the capacity imbalance at component. The lowest-capacity aware packing algorithm can be concluded where the algorithm looks at the w; lowest capacities and searches for an item which has the similar w corresponding largest component requirements. The parameter w is a window into the current bin state. This is the common windowed multi-capacity bin-packing heuristic.

Permutation Pack: Permutation Pack (PP) attempts to find items in which the largest w components are ordered exactly as same as the ordering of the corresponding smallest elements in the current bin. For example, consider the case,d = 5 and the capacities of the current bin are ordered as follows:

Choose Pack: The Choose Pack (CP) algorithm is a further relaxation of the PP algorithm. CP also attempts to match the w smallest bin capacities with items in which the corresponding w components are the largest. The key difference is that CP does not enforce an ordering between these w components. As an example, consider the case where w = 2 and the same bin state exists as in the previous example. CP would search for an item in which but would not enforce any particular ordering between Xi1 and Xi3. This heuristic partitions the input list into logical sub lists thus reducing the time complexity by over PP.

III. SYSTEM FLOW DIAGRAM



IV. ADVANTAGES

- Will reduce bandwidth fluctuations and allow the transfer of computing data to/from the phone at no cost.
- Scheduling algorithm is to minimize the make span of a set of computing tasks.

V. FUTURE WORK

Blindly executing tasks for extended durations on a smartphone being charged, can prolong the time taken for the phone to fully charge. Show that intensive use of a phone's CPU can delay a full charge by 35 percent. In future here apply a new task sleeping mechanism by using ant colony optimization methods which ensures that task executions do not impact the charging times.

VI. CONCLUSION

In this paper, envision building a distributed computing infrastructure using smartphones for enterprises. This vision is based on several compelling observations including (i) enterprises gives their staffs with smartphones in many situations, (ii) Mostly the phones are idle when being charged, and (iii) in that case, infrastructure could yield significant cost benefits to the companies. During scheduling of task, the parallel system is represented by a bin with d capacities corresponding the multiple resources available in the system. The job wait queue is represented by an item list where each item is described by a d-capacity requirements vector. A scheduling epoch consists of packing jobs from the job queue into the currently available resources in the parallel system. The information available in the additional capacity requirements for each job is used to guide the scheduling process. Contribution is to provide multi-capacity aware bin-packing algorithms which uses the information in the additional capacities to guide item selection in the packing process. Past research in multi-capacity bin-packing has focused on extending the single capacity bin-packing to deal with the multiple capacities, and on providing performance bounds on these simple algorithms. Repeat the same simulation, this time with migration enabled in CWC (labeled CWC-MIG). With migration, if a phone fails during a sub-task, that particular sub-task's state is saved and only the remaining input file of the sub-task needs to be re-scheduled. Have a prototype implementation of CWC on a testbed of 8 Android phones. Using this implementation, demonstrate both the viability and efficacy of various components within CWC.

REFERENCE

1. F. B€usching, S. Schildt, and L. Wolf, "DroidCluster: Towards smartphone cluster computing—The streets are paved with potential computer clusters," in Proc. 32nd Int. Conf. Distrib. Comput. Syst. Workshop, Jun. 2012, pp. 114–117.
2. S. Schildt, F. B€usching, E. J€orns, and L. Wolf, "CANDIS: Heterogeneous mobile cloud framework and energy cost-aware scheduling," in Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things and IEEE Cyber, Phys. Social Comput., Aug. 2013, pp. 1986–1991.
3. P. R. Elespuru, S. Shakya, and S. Mishra, "MapReduce system over heterogeneous mobile devices," in Proc. 7th IFIPWG10.2 Int. Workshop Softw. Technol. Embedded Ubiquitous Syst., 2009, pp. 168–179.
4. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in Proc. 6th Conf. Symp. Operating Syst. Des. Implementation, 2004, p. 10.
5. [Online]. Available: http://research.cs.wisc.edu/condor/, 2014.
6. D. Datla, X. Chen, T. Tsou, S. Raghunandan, S. M. Hasan, J. H. Reed, B. Fette, C. B. Dietrich, J.-H Kim, and T. Bose, "Wireless distributed computing: A survey of research challenges," IEEE Commun. Mag., vol. 50, no. 1, pp. 144–152, Jan. 2012.
7. M. Conti, S. Giordano, M. May, and A. Passarella, "From opportunistic networks to opportunistic computing," IEEE Commun. Mag., vol. 48, no. 9, pp. 126–139, Sep. 2010.
8. D. Datla, H. I. Volos, S. M. Hasan, J. H. Reed, and T. Bose, "Task allocation and scheduling in wireless distributed computing networks," Analog Integr. Circuits Signal Process., vol. 69, nos. 2/3 pp. 341–353, Dec. 2011.
9. D. Datla, H. I. Volos, S. M. Hasan, J. H. Reed, and T. Bose, "Wireless distributed computing in cognitive radio networks," Ad Hoc Netw. Special Issue Cognitive Radio Ad Hoc Netw., vol. 10, no. 5, pp. 845–857, Jul. 2012.
10. T. Das, P. Mohan, V. N. Padmanabhan, R. Ramjee, and A. Sharma, "PRISM: Platform for remote sensing using smartphones," in Proc. 8th Int. Conf. Mobile Syst., Appl. Services, 2010, pp. 63–76.
11. D. Estrin, "Participatory sensing: Applications and architecture," in Proc. 8th Int. Conf. Mobile Syst., Appl. Services, 2010, pp. 3–4.
12. Z. Dong, L. Kong, P. Cheng, L. He, Y. Gu, L. Fang, T. Zhu, and C. Liu, "REPC: Reliable and efficient participatory computing for mobile devices," presented at the IEEE Int. Conf. Sensing, Communication and Networking, Singapore, 2014.
13. E. Oliver, "The challenges in large-scale smartphone user studies," in Proc. 2nd ACM Int. Workshop Hot Topics Planet-Scale Meas., 2010, pp. 5:1–5:5.
14. H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin, "A First Look at Traffic on Smartphones," in Proc. 10th ACM SIGCOMM Conf. Internet Meas., 2010, pp. 281–287.
15. A. Shye, B. Scholbrock, G. Memik, and P. A. Dinda, "Characterizing and modeling user activity on smartphones: Summary," in Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst., 2010, pp. 375–376.
16. E. Oliver, "Diversity in smartphone energy consumption," in Proc. ACM Workshop Wireless Students, by the students, for the Students, 2010, pp. 25–28.
17. D. Ferreira, A. K. Dey, and V. Kostakos, "Understanding humansmartphone concerns: A study of battery life," in Proc. 9th Int. Conf. Pervasive Comput., 2011, pp. 19–33.