

Peer-to-Peer Checkpointing Arrangement for Mobile Grid Computing Systems

Paul J. Darby III

Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, Louisiana 70504
1-337-482-5741

pauldarby@aol.com

Nian-Feng Tzeng

Center for Advanced Computer Studies
University of Louisiana at Lafayette
Lafayette, Louisiana 70504
1-337-482-6304

tzeng@cacs.louisiana.edu

ABSTRACT

This paper deals with a novel, distributed, QoS-aware, peer-to-peer checkpointing arrangement component for mobile Grid (MoG) computing systems middleware. Checkpointing is more crucial in MoG systems than in their wired counterparts due to node mobility and less reliable wireless links resulting in frequent and dynamic connections and disconnections. Having determined the globally optimal checkpoint arrangement to be NP-complete, we consider ReD, our **Reliability Driven (ReD)** protocol, employing QoS-aware heuristics, for constructing superior peer-to-peer checkpointing arrangements efficiently.

Categories and Subject Descriptors

D.4.7 [Operating Systems]: Organization and Design – *Distributed systems*.

General Terms

Algorithms, Measurement, Performance, Design, Reliability.

Keywords

Peer-to-Peer wireless checkpointing arrangement, computational Grids, collaborative job execution, Mobile Grids, Checkpointing

1. INTRODUCTION

Grid computing systems have seen their widespread adoption lately in not only academia but also industry. While most existing Grids refer to clusters of computing and storage resources which are wire-interconnected for offering utility services collaboratively, mobile Grids (MoGs) have started to receive growing attention and they are expected to become an integral and critical part of a future computational Grid involving mobile hosts that facilitate user access to the Grid and also offer computing resources for applications [1]. A MoG can be comprised of a number of mobile hosts (MHs), i.e., laptop computers, PDAs, wearable computing gear, or even sensors, having wireless interconnections among one another or to access points. Current trends toward powerful multicore processors, efficient and small flash memory devices, and wireless technologies, such as IEEE 802.16 WiMAX (having a capacity of 10 Mbps or more for distances over a few miles), are seen as the

technology enablers of the practical MoG, while models for compensation, accounting, and regulation of these systems are under development. For example, some authors have proposed a fair pricing strategy and optimal job allocation scheme for MoG computing by drawing upon the Nash Bargaining solution to maximize Grid revenue from the viewpoint of the user [2]. Other researchers have proposed a middleware framework utilizing mobile agents for secure MoG services while addressing the heterogeneity issues of concern with this technology and providing compatible interfaces to the Globus Toolkit [3]. The distributed applications likely to run on these MoGs are not traditional long-running heavy computing jobs commonly found in their wired counterparts. Instead, what is envisioned are scenarios involving distributed applets with moderate execution durations (such as address book synchronization), interactive games, and distributed microcode sensor applications, requiring data from disparate locations. Likewise, MoGs are suitable for distributed scientific or engineering applications in remote areas where access to a wired Grid is infeasible or where the highly mobile characteristics of the task make wired Grid access impractical and the MoG must operate with autonomy. In these scenarios, a MoG is seen to offer collaborative computation, practically, not just to add user access and mobility to the wired Grid considered earlier [1]. This article explores the details of our **Reliability Driven (ReD)** protocol, a QoS-aware middleware, functional mechanism, employing heuristics which seek continually, to set up and maintain optimal peer-to-peer checkpointing arrangements for a MoG even under dynamic conditions. In these systems, MoGs don't merely access Grid computing services, but offer their resources, collaboratively to provide services. In both simulations and actual testbed implementation, ReD has been demonstrated to yield favorable recovery probabilities in comparison to a heuristically neutral baseline protocol. Section 2 outlines related work pertinent to checkpointing in wire-connected systems and to wireless systems with designated base stations BSs. Section 3 discusses the theory and methodology of our proposed Reliability Driven (ReD) protocol for peer-to-peer checkpointing arrangement, in a dynamic MoG, a novel and needed area of research, not currently being considered elsewhere. Section 4 lists select references.

2. RELATED WORK

Checkpointing in wired Grid computing systems has been investigated earlier with various methodologies. Such systems generally assume that the constituent nodes are all connected by high-speed wired links with small latency and low link and node

failure rates and that the computation interval and the checkpoint overhead are much smaller than the mean time between failures (MTBF) [4]. Thus, in wired systems, only relatively long-term applications really need be checkpointed. However, with the correspondingly low MTBF for wireless links in the MoG, these assumptions clearly do not hold. In such systems even relatively short distributed applications require checkpointing if they are to guarantee any practical quality of service level.

Wireless System Checkpointing with BSs

Earlier work on checkpointing in wireless computing systems has relied on the wire-connected network and Grid to provide a stable checkpoint storage platform. With this methodology, checkpointed data has to be stored on stable, safe storage (normally at a computer server or PC on a wired network, known as a base station, BS) [5]. Checkpointing wireless MHs to BSs has its own drawbacks, however, when not all MHs are adjacent to BSs or when BSs do not exist (like the MoG at hand). This is because mobility itself presents major impediments to moving checkpoint data among the MHs and to the BSs often over multiple unreliable hops. Intermittent disconnections and low bandwidth become problematic in this methodology resulting in excessive retransmissions and the associated heavy traffic delays. This leads to our investigation into the use of MHs themselves for keeping checkpointed data. Our approach is especially attractive if link bandwidth and reliability are of key concern.

3. PEER-TO-PEER CHECKPOINTING

The MoG of our interest does not utilize or require a BS, and its constituent MHs are not adjacent to any BS. Thus, our checkpointing strategy for the MoG aims to keep checkpointed data at immediate neighboring MHs. In order to limit the use of relatively unreliable wireless links, while minimizing the consumption of wireless node memory resources and energy, each MH sends its checkpointed data to *one and only one* neighboring MH, and also serves to take checkpointed data from exactly one neighboring MH, realizing peer-to-peer checkpointing. When a given peer MH, say A sends its checkpointed data to another peer MH, say B, for safe storage, A is called the “consumer” and B is the “provider” of checkpointing services. We symbolize this relationship via $A \rightarrow B$. The central mechanism of our MoG middleware component is our Reliability Driven (ReD) protocol, which is aware of the reliabilities of links among MHs within the MoG, a significant indicator of the service quality (i.e., QoS) a distributed application will receive. Defining the term “connectivity” to mean the parallel reliability of all links from a given node to its neighbors, ReD makes use of these link reliability values to determine the best possible checkpointing arrangement dynamically. We seek to maximize $R_A = \max \prod_{i=0}^{n-1} [1 - (1 - C_i)(1 - P_j L_{ij})]$, $i \neq j$, where R_A is the arrangement reliability we seek to maximize, C_i is the connectivity of consumer i , P_j is the connectivity of provider j , and L_{ij} is the reliability of the wireless link from C_i to P_j . Because we determined that the problem of finding the optimum checkpointing arrangement to be NP-Complete, ReD utilizes a heuristic algorithm. To ensure convergence within a reasonable time, the global MoG is partitioned into clusters. Because ReD operates within clusters, and is localized, it often arrives at suboptimal checkpoint arrangements.

ReD Operation

Upon initiation or a refresh, if a prospective *consumer* MH does not have a designated provider, that consumer begins to look for a checkpoint provider. In doing so, it examines and compares the $P_j L_{ij}$ product of each of its neighboring nodes. The prospective consumer sorts the products in decreasing order. It then transmits a checkpoint request first to the list top node (as prospective provider, since its product is the greatest). If a positive acknowledgement is received, then this node sets its checkpoint “to pointer” to the positively responding provider. If, on the other hand, there is a negative reply or no reply after 5 tries or if a *Break* message received, then the consumer abandon’s its attempt with the top provider and sends a checkpoint request to the next provider on the sorted list, and so on. If it finds no provider it goes to sleep until the next refresh period.

From the perspective *provider* node’s point of view, if a request is received from a prospective consumer and there is no existing consumer of record, then it sends an positive acknowledgement to the requesting consumer and sets its “from pointer” to point to that consumer. If, on the other hand, the prospective provider already has a consumer of record, then if the requesting consumer’s $C_i \parallel L_{ij}$ calculated reliability is less than the same for the consumer of record, the requesting consumer is sent a positive acknowledgement, the consumer of record is sent a Break message, and the prospective provider’s “from pointer” is set to point at the new consumer. Otherwise, the requesting consumer is sent a negative acknowledgement. ReD, being QoS-aware, efficiently and quickly converges in a dynamic MoG environment to produce superior checkpointing arrangements.

ACKNOWLEDGEMENT

This work is support in part by the U.S. Department of Energy (DOE) under Award Number DE-FG02-04ER46136 and by the Board of Regents, State of Louisiana, under Contract Number DOE/LEQSF(2004-07)-ULL.

REFERENCES

- [1] Wesner, S. *et al.*, “Mobile Collaborative Business Grids – A Short Overview of the Akogrimo Project.” White Paper, Akogrimo Consortium, 2006.
- [2] Ghosh, P., Roy, N., Das, S. and Basu, K. “A Game Theory Based Pricing Strategy for Job Allocation in Mobile Grids.” *Proceedings of 18th Intl. Parallel and Distributed Processing Symposium (IPDPS '04)*, April 2004, pp. 82 - 91.
- [3] Wong, S. and Ng, K. “A Middleware Framework for Secure Mobile Grid Services.” In *Proceedings of the 6th IEEE Intl. Symposium on Cluster Computing and the Grid Workshops (CCGRIDW '06)*, November 2006, pp. 2 - 12.
- [4] Wang, L. *et al.*, “Modeling Coordinated Checkpointing for Large-Scale Supercomputers.” In *Proceedings of the Intl. Conference on Dependable Systems and Networks*, July 2005, pp. 812 - 821.
- [5] Lin, C. Kuo, S., and Huang Y., “A Checkpointing Tool for Palm Operating System.” In *Proceedings of Intl. Conference on Dependable Systems and Networks*, July 2001, pp. 71 - 76.