行政國家科學委員會專題研究計劃成果報告

藉由線上建立鋸齒角錐尋找一致全域檢查點

Finding Global Checkpoints by On-line Construction of Z-cones

計劃編號：NSC 88-2213-E-009-014

執行期限：87 年 8 月 1 日至 88 年 7 月 31 日

主持人：黃廷祿　　國立交通大學資訊工程學系

E-mail address: tlhuang@csie.nctu.edu.tw

一、中英文摘要

尋找一分散式計算(distributed computation)中的一致全域檢查點(consistent global checkpoints)在許多分散式的應用，如分散式測試(distributed testing)、分散式除錯(distributed debugging)、及容錯計算(fault-tolerant computing)中是一個中心問題。我們考慮的問題是建構所有包含一給定區域檢查點(local checkpoints)集合 S 的一致全域檢查點。我們首先提供一藉由合併 S 中區域檢查點之因果角錐(C-cones)來建構 S 之因果角錐的機制 (所謂 S 之因果角錐乃是所有與 S 無因果關係的區域檢查點所成的集合)。之後，我們提供一個利用因果角錐尋找所有一致全域檢查點的演算法。最後，為簡化建構 S 之因果角錐之工作，我們提供一個線上(on-line)建構區域檢查點之因果角錐的演算法。

關鍵詞：分散式系統、因果關係、因果路徑、鋸齒路徑、一致全域檢查點

Abstract

Finding consistent global checkpoints of a given distributed computation is a central problem in many distributed applications, such as distributed testing, distributed debugging, and fault-tolerant computing. Given a set of local checkpoints S, each from a different process, we consider the problem of constructing all consistent global checkpoints containing S. We first provide a mechanism to generate C-cone(S), the set of checkpoints that are causally unordered with S, by combining the C-cones of individual checkpoints in S. Then, an algorithm that uses C-cones to enumerate all consistent global checkpoints containing S is given. Finally, to facilitate the construction of C-cone(S), we present an algorithm to generate C-cones of local checkpoints on-line.

Keywords: Distributed systems, causality, causal paths, zigzag paths, consistent global checkpoints.

二、緣由與目的

Finding consistent global checkpoints of a given distributed computation is a central problem in distributed systems, such as distributed debugging [9] and fault-tolerant computing [3]. In this paper, we consider the problem of constructing all consistent global checkpoints containing a given set of local checkpoints S. The definition of consistency states that if S can belong to a consistent global checkpoint, then S must contain one checkpoint from each of the n processes and none *happened before* [4] any other in S. However, when $|S| < n$, having no causal path (*C-path* for short) between checkpoints in S is not sufficient to

ensure that S can be extended to a consistent global checkpoint. Netzer and Xu [7] proved the necessary and sufficient condition for an arbitrary set of local checkpoints to belong to a consistent global checkpoint by introducing zigzag paths (*Z-paths* for short). Manivannan, Netzer, and Singhal [6] proved exactly which local checkpoints can be used for constructing consistent global checkpoints. They observed that only those checkpoints in Z-cone(S) [Definition 2] that are not involved in a zigzag cycle can be combined with S to form consistent global checkpoints. These checkpoints are said to be *USEFUL* to S. They also provided an algorithm, which we call **Algorithm MNS97**, to enumerate all consistent global checkpoints containing S.

The time spent on finding *USEFUL* checkpoints will become the performance bottleneck of **Algorithm MNS97** unless an efficient algorithm for finding Z-cones is provided. Unfortunately, such algorithms do not exist currently. Two methods can be considered to improve the performance of **Algorithm MNS97**:

1. To impose some constraint on the checkpoint and communication pattern such that the pattern satisfies RD-trackability [11]. This will cause Z-cone(S) and C-cone(S) to become equivalent. However, extra checkpoints must be created in this method.
2. To select candidates from a broader but *easier acquired* set of local checkpoints, such as C-cones. The drawback of this method is that it is possible to select a *non-USEFUL* checkpoint that cannot be combined with S. In this situation, backtracking must be performed.

Constructing C-cones of local checkpoints on-line is promising because we observe that C-cones of past checkpoints can be obtained while execution progressing. In this paper, we introduce an algorithm that constructs C-cones of local checkpoints on-line. Moreover, we present a variant of **Algorithm MNS97** to enumerate all consistent global checkpoints containing a given set of local checkpoints by using C-cones.

三、結果與討論

The notion of the Z-cone, and of which checkpoints within the Z-cone are *USEFUL*, provides a new understanding of the structure of consistent global checkpoints. However, finding Z-cones based on the R-graph of a given checkpoint and communication pattern is a time-consuming job and will become the performance bottleneck of **Algorithm MNS97**. Since Z-cones are too difficult to be found, we pay our attention to the broader but easier acquired sets -- the C-cones. In this paper, we provide an on-line algorithm for finding C-cones of local checkpoints to facilitate the construction of consistent global checkpoints. In general, it is possible to select a *non-USEFUL* checkpoint in the C-cone and backtracking must be performed in this case. If some constraint, such as RD-trackability, is imposed on the checkpoint and communication pattern, the C-cone and Z-cone of a given set of checkpoints will become equivalent; under such a constraint, our algorithm can construct Z-cones of local checkpoints efficiently.

四、成果自評

目前有碩士生羅健誠已經完成相關論文，尚未發表在期刊上。此研究方向頗具學術價值，可再加強內容對外發表。唯主持人已將相關結果應用在另外一篇論文，發表於 IEEE ICDCS'99 大會上，其題目為："Fast and fair mutual exclusion for shared memory systems", pp. 224~231.

五、參考文獻

[1] Brzezinski, J., Helary, J.M., and Raynal, M. Termination detection in a very general distributed computing model. *Proc. 13th International Conference on Distributed Computing Systems*. May 1993, pp. 374--381.

[2] Huang, S.T. Detecting termination of distributed computations by external agents. *Proc. 9th International Conference on Distributed Computing Systems*. 1989, pp. 79--84.

[3] Koo, R., and Toueg, S. Checkpointing and rollback-recovery for distributed systems. *IEEE Trans. Software Engrg.* **SE-13**, 1 (Jan. 1987), 23--32.

[4] Lamport, L. Time, clocks and the ordering of events in a distributed system. *Comm. ACM* **21**, 7 (July 1978), 558--565.

[5] Lazowska, E.D., Zahorjan, J., Cheriton, D.R., and Zwaenepoel, W. File access performance of diskless workstations. *ACM Trans. Comput. Systems* **4**, 3 (Aug. 1986), 238--268.

[6] Manivannan, D., Netzer, R.H.B., and Singhal, M. Finding consistent global checkpoints in a distributed computation. *IEEE Trans. Parallel Distrib. Systems* **8**, 6 (June 1997), 623--627.

[7] Netzer, R.H.B., and Xu, J. Necessary and sufficient conditions for consistent global snapshots. *IEEE Trans. Parallel Distrib. Systems* **6**, 2 (Feb. 1995), 165--169.

[8] Russel, D.L. State restoration in systems of communicating processes. *IEEE Trans. Software Engrg.* **6**, 2 (Mar. 1980), 183--194.

[9] Venkatesan, S., and Dathan, B. Testing and debugging distributed programs using global predicates. *IEEE Trans. Software Engrg.* **21**, 2 (Feb. 1995), 163--177.

[10] Wang, Y.-M., Lowry, A., and Fuchs, W.K. Consistent global checkpoints based on direct dependency tracking. *Inform. Process. Lett.* **50**, 4 (May 1994), 223--230.

[11] Wang, Y.-M. Consistent global checkpoints that contain a given set of local checkpoints. *IEEE Trans. Comput.* **46**, 4 (Apr. 1997), 456--468.