DEVELOPMENT RELIABILITY NODE FAULT-TOLERANT COMPUTING SYSTEMS BY PARALLEL TECHNICS.

Kamil Iehab Abduljabbar Kamil, M.B. Abrosimov

Reliability and availability have become increasingly important in today's computer-dependent world. In many applications where computers are used, outages or malfunction can be expensive, or even disastrous. Which includes high availability, improved throughput and response time. Due to the complexity of such systems, it is impossible to improve computer performance using a single processor, the system must be able to operate correctly despite the presence of certain faults. Fault-tolerance the important technique used to the ability of a system to continue performing its intended function, in spite of faults maintains dependability in these systems. In a broad sense, fault tolerance is associated with reliability. The theory of fault tolerance in graphs as a mathematical model for fault tolerance in computers was introduced in a seminal paper by Hayes [1]. In fault-tolerant systems, spare components are needed so that when some basic component fails, their tasks are dynamically transferred to the spare component. The problem here is to minimize the number of spare components that are needed for the network to become fault-tolerant and deciding whether graphs are isomorphic. In general, the implementation of a serial based graph algorithm is time-consuming as the number of vertex and edge in graph increases as the size of the graph increases [2]. Recently there have been increasing graph applications with large sizes because of increasing big data provided [3, 4]. Due to these large data, the parallel implementation of graph algorithms is more effective than serial-based implementation. we need to parallel algorithm and special approach for contemporary massively parallel computers that run efficiently and general technique for generating families of combinatorial objects without isomorphs. In our theses, we used random graphs from vertices of small size to up to 9 vertices. The experimental results show that the proposed approach brings a considerable improvement in performance and efficiency compared to the CPU-based results. Our result also shows high performance, especially on large graphs.

References

1. Hayes J.P. A graph model for fault-tolerant computing system // IEEE Transactions on Computers. – 1976. – Vol. C-25, no. 9. – P. 875–884. – DOI: 10.1109/TC.1976.1674712.

2. Ullmann J.R. An algorithm for subgraph isomorphism // Journal of the ACM (JACM). – 1976. – Vol. 23. – P. 31-42.

3. A (sub)graph isomorphism algorithm for matching large graphs / L.P. Cordella [et al.] // Pattern Analysis and Machine Intelligence. – 2004. – Vol. 26. – P. 1367–1372.

4. An Improved Algorithm for Matching Large Graphs / L.P. Cordella [et al.] // 3rd IAPR-TC15 workshop on graph-based representations in pattern recognition. – 2001. – P. 149–159.