ECE5658

Critiques - week9

Student info:
Name: Jaeyoun Nam
Student ID: 2014310198
Email: siisee111@gmail.com

1 READ-LOG-UPDATE

(1) **summary of the paper:** It appears overcoming the shortcomings of Read-Copy-Update (RCU) mechanism. The serious problem with *RCU* is that only single pointer update is allowed. So, there would be cases that reader read inconsistent data.
(2) **strengths/weakness of the paper:** Read-Log-Update can support multiple writers by using concept of clock while providing similar speed of *RCU*.

2 Everything You Always Wanted to Know About Synchronization but Were Afraid to Ask

(1) **summary of the paper:** Scaling software systems to many-core architecture is important challenge, but many programmer has little knowledge about synchronization scheme. This paper explain some study of synchronization.
(2) **strengths/weakness of the paper:** This paper and their proposed *SSYNC* would helps programmer and designer to predict their scheme, and inspect performance behavior.
(3) **any suggestions to improve their idea:** It doesn't treat lock-free data structure.

3 An Analysis of Linux Scalability to Many Cores

(1) **summary of the paper:** As the title represented, It analyze linux kernel and application's scalability issues on many-cores and introduce a new technique named *sloppy counters* that can remove bottleneck while modifying little codes. They examine kernel bottleneck and user application bottleneck either.
(2) **strengths/weakness of the paper:** Linux uses shared counter to manage resources, but they can cause scalability issues. Then this paper shows how shared counter be a bottleneck and introduce new tecqnique called *sloppy counters*.

4 The Scalable Commutativity Rule: Designing Scalable Software for Multicore Processors

(1) **summary of the paper:** This paper aims to detect opportunities for more scalable system call API. This paper introduce *COMMUTER* that helps programmer to apply *"Whenever interface operations commute, they can be implemented in a way that scales"* rule suggested by this paper.
(2) **strengths/weakness of the paper:** Suggested Approach can be adopted to the software interface to examine scalability, so It can be done before implementation.
(3) **any suggestions to improve their idea:** To use commuter, we have to generate test code.