Summary of paper presented on November 16, 2008

Venkatraghavan Ramesh
1200072784

Paper – “Proactive Fault Tolerance for HPC with Xen Virtualization”, A B Nagarajan, F Mueller, C Engelmann, S L Scott, ICS’ 07

Presenter – Stephen Orchowski

The paper proposes a means to achieving proactive fault tolerance by monitoring health of the node and migrating jobs to healthy nodes before failure can occur. Following are some of the strengths and weaknesses of the paper that were discussed during the presentation and by those involved in critiquing the paper –

Strengths –

1. A proactive scheme will greatly help in spreading the stress of workloads across nodes thereby increasing MTBF in large-scale systems, which will improve the life of overall system. The example of how failure of one node required shutdown of an entire 1024 processor midplane in BlueGene illustrates this point.
2. The implementation of proactive approach is straightforward with virtualization. With support for virtualization being available easily from processor vendors, proactive FT is a realistic option.
3. The speedup due to live migration has been established clearly.

Weaknesses –

1. The overhead due to the sensors involved in health monitoring and the Baseboard Management Controller is not discussed in detail (though other works have been quoted) as pointed out by those critiquing. Pravin suggests that even with event triggered mechanism, which the authors claim will offload overhead from application side onto BMC, the cost will not be justified unless the system is large enough. Hence, for the periodic sampling alternative that has been implemented by the authors, the overhead would have been substantial and must have been dealt with in detail.
2. Other individual overheads, possibilities of spurious job migration due to proactive FT and overall feasibility study are also missing.
3. It is not clear how the system will recover from faults incase the faults are not anticipated. There might be factors other than those covered under the spectrum of “system health” that could lead to failures. The implementation cost of reactive scheme would be quite substantial apart from the operating cost. If the system will also implement reactive scheme in conjunction with the proactive scheme, this implementation cost will not be eliminated.
4. The experimental setup used by the authors could have been more broad in its coverage of scenarios such as nodes, migration schemes etc. Further, their method of directly injecting faults rather than using the PFT daemon is debatable as it might not be representative of the way in which the proposed proactive scheme would function.