Review Report

Title: SHStream: Self-healing Framework for HTTP Video-Streaming
Conference: ACM Multimedia Systems 2013

Summary: This paper presents a self-healing framework for video-streaming web servers that provides load admittance, failure detection, failure prediction, failure diagnosis, model evaluation and proactive recovery. It uses online learning algorithms as the core techniques to predict performance anomalies and uses container-based virtualization to implement failover functionalities. Experimental results showed that high levels of recall and prevision can be achieved for failure prediction.

Importance: Of sufficient interest
Originality: Moderately original
Technical Correctness: Probably correct
Clarity of Presentation: Clear enough
Reference to Prior Work: References adequate

Major Strengths:
1) Several existing types of online algorithm are used for failure prediction to overcome the limitations of traditional batch learning. Better predictive performance is achieved by evaluating multiple models continuously and picking the best model from the constituent models in each iteration. In the learning process, a window of uncertainty is used skillfully to ignore all instances preceding the pre-failure period for learning. Evaluation results showed that a stable and low number of false negatives can be obtained only with a small number of learning instances.
2) A failover mechanism using container-based virtualization can perform online migration within one second to carry on the delivery of video content from the faulty server.

Weaknesses:
1) The performance of failure type classification is so unsatisfactory and unsuccessful. Evaluation results showed that the number of misclassifications is even more than the number of correct classifications. I guess that the problem is in the learning activity. There are four failure types in this paper: CPU, Memory, I/O and Misc. Besides process-related metrics (e.g, CPU and memory consumed by the web server process), one learning instance also contains application-level metrics such as response time, number of network failures, and bytes read from disk. Based on my intuitive judgements, there may be few intrinsic similarities between application-level metrics of two instances associated to the same failure type. Thus, it is a little difficult for online learning algorithms to capture pre-failure patterns associated to specific failure types accurately.
2) Some important issues are missing. When I read Section 5, SHStream Implementation, I have several doubts. What actions will SHStream perform when a

failure is detected by Failure Detection Module? How do the Failure Detection Module and the Failure Prediction Module work cooperatively to avoid performance anomalies? How does the F-measure be updated at each moment? How do the results of failure type classification or failure diagnosis help the subsequent proactive recovery? In addition, there are no useful and further discussion how parameter a in Section 5.3 and the buffer size in Section 5.4 are determined.

3) The experiments are a weak point of the paper. There are no experimental validation for the performance of Load Admittance Module and Recovery Module.

4) Typos:

Page1:

the convergence of TV **with** the Internet → the convergence of TV **and** the Internet

occur server-side→occur **at the** sever-side

checkpointing and recovering of TCP connection states **requires**→ checkpointing and recovering of TCP connection states **require**

Page2:

but **helps** bootstrapping models→ but **help** bootstrapping models

Results **shown** that UBFs **yields** the best results for free physical memory prediction and SVMs **performed** better predicting server response times→Results **showed** that UBFs **yield** the best results for free physical memory prediction and SVMs **perform** better predicting server response times.

uses information about non-fatal events→ **used** information about non-fatal events

Context-aware models **groups** → Context-aware models **group**

Page3:

the problem resumes in incrementally **learn** a classier→ the problem resumes in incrementally **learning** a classier

traditional decision trees algorithms **requires** all data→ traditional decision trees algorithms **require** all data

Page4:

Learn models by giving a positive→ **Learns** models by giving a positive

Learning from a bootstrap→ **Learns** from a bootstrap

it **it** possible to reduce the space→ it **is** possible to reduce the space

a small number of option nodes **is** used→a small number of option nodes **are** used

Page5:

easily adaption→ **easy** adaption

Page6:

Our tests were performed on a **tested** →Our tests were performed on a **testbed**

Page7:

ssh commands that **invokes** the Stress tool→ssh commands that **invoke** the Stress tool

Conclusion:

This paper is more like a work-in-progress paper, which may be better suited for workshops rather than the main conference.