

# HAL: Towards Operator-Proof Systems Management

Fábio Oliveira (PhD student), Ricardo Bianchini, Richard P. Martin, Thu D. Nguyen

{fabiool, ricardob, rmartin, tdnguyen}@cs.rutgers.edu

Department of Computer Science, Rutgers University, Piscataway, NJ 08854

The complexity of today's computer systems poses a challenge to system administrators. Current systems comprise a multitude of inter-related software components running on many servers. To make matters worse, as computers permeate all aspects of our lives, higher demands are placed on the availability and correct operation of the services provided by the computer systems. Unfortunately, few current systems can legitimately claim to be highly available, and many studies over the years have observed that human mistakes are an important source of unavailability in complex systems.

Responses from an extensive survey we conducted of professional network administrators and DBAs (Database Administrators) further substantiate this problem. Among common failures reported by network administrators, 43% of them can be attributed to operator actions, with software and hardware being the other prominent causes. The results show that mistakes happen due to a variety of reasons, including lack of understanding of system functioning by operators, complexity of the maintenance task, operator inattentiveness, and lack of appropriate tools to help in operations. Moreover, according to the surveyed DBAs, DBA mistakes are responsible (entirely or in part) for roughly 80% of the database administration problems they reported.

These findings suggest the need for effective system support for mistake-free system maintenance. Previously, we proposed an approach for human operators to check the correctness of their actions in a virtual sandbox environment that extends the online system [1]. We experimented with two techniques within a prototype implementation of the environment for a multi-tier online Internet service. The prototype caught two-thirds of all mistakes we observed in human-factor experiments using volunteer operators. Our techniques relied on comparing the behavior under test with a known correct behavior, in the form of traces (trace-based validation) or an online replica (replica-based validation). Afterwards, we adapted the same techniques to database administration and proposed model-based validation as an approach to test DBAs' actions in the absence of an instance of correct behavior for comparison, by checking a set of assertions that define system correctness [2].

Although trace and replica-based validation can hide operator mistakes through virtualization, requiring the operator to validate every action might become cumbersome, specially in large systems, and be unnecessary for harmless actions. In addition, validation does not provide protection against actions that directly affect the online system, nor is it applicable to actions that change the behavior of the affected component(s) and are not covered by the assertion model.

In this work, we propose the idea of operator-proof systems, a radically different approach to dealing with human operator mistakes. In these systems, an omnipresent management infrastructure enables the managed system to defend itself against operator mistakes. In a reference to "2001: A Space Odyssey", we call this infrastructure HAL. HAL keeps monitoring the operator actions and the system state to decide whether or not the system should react to what the operator is doing. The reaction would prevent an observed (or potential) mistake from compromising the whole system. For instance, while the operator is editing a critical configuration file on a Web server, HAL may decide to preventively make the same configuration file immutable on all other Web servers until it is certain that the operator is not performing any harmful action. In deciding whether a reaction is needed and what reaction is the most appropriate, HAL predicts what the operator is trying to accomplish and evaluates whether to react to it based on the probability of mistake for the predicted task, the cost of all applicable reactions, and the cost of not reacting at all.

We have designed a prototype of HAL that can be applied to multi-tier Internet services. In particular, we have a monitoring infrastructure that collects information about what the operator is doing on all servers. We have also designed and implemented the task prediction module of HAL, which performs recursive Bayesian estimation to predict the task that the operator is executing based on the information collected by the monitoring infrastructure. We carried out a preliminary evaluation of the accuracy of the task prediction model by applying it to the traces of operator commands we collected during our previous human-factor study [1]. HAL was able to correctly identify with high confidence the operator intent in all 43 collected traces. Currently, we are defining a set of reactions that can be applied to multi-tier services, along with the model for the cost-sensitive analysis HAL performs to decide on whether it should react to the operator action and what reaction should be taken. The cost-sensitive analysis will consider factors, such as, the impact of the predicted operator intent, the probability of mistake, the difficulty to recover the system from potential mistakes during the predicted task, and the impact of the reactions on the system performance and their estimated runtime.

Lying at the intersection of systems and machine learning, this work poses some questions that we shall answer. How to represent the reactions and the models to generalize our approach to other systems? How much should the system engineer tell HAL upfront? How much should HAL learn by itself from experience? How effective at providing high system availability will HAL be? What are all the possible approaches to model task prediction, mistakes, performance impact of operator actions, and cost of reactions? How do different modeling approaches compare to each other?

## References

- [1] K. Nagaraja, F. Oliveira, R. Bianchini, R. P. Martin, and T. D. Nguyen. Understanding and Dealing with Operator Mistakes in Internet Services. In *Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI'04)*, Dec. 2004.
- [2] F. Oliveira, K. Nagaraja, R. Bachwani, R. Bianchini, R. P. Martin, and T. D. Nguyen. Understanding and Validating Database System Administration. In *Proceedings of the 2006 USENIX Annual Technical Conference*, June 2006.