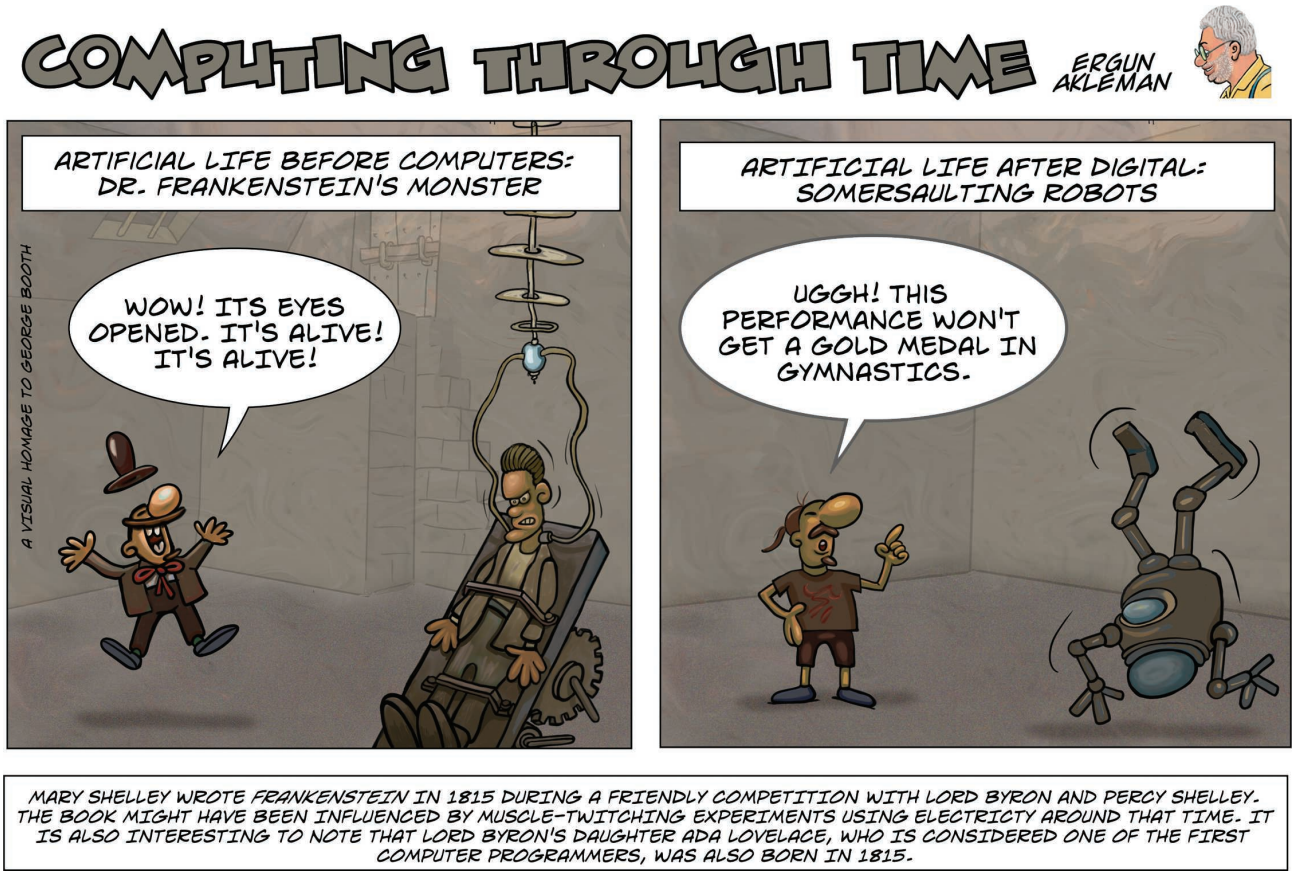

Digital Object Identifier 10.1109/MC.2020.2970820
Date of current version: 9 April 2020

describes and implements many of the issues that also had to be solved again in today's massive computer farms.]

Tuning Memory Performance of Sequential and Parallel Programs (p. 32) "When programmers know where and why memory bottlenecks occur, they can make the appropriate program transformations to enhance performance, with the help of this tool's detailed statistics. ... MemSpy uses cache simulations to gather detailed memory statistics. Since efficiency is a key concern in simulation-based performance monitoring, we have evaluated two performance optimizations—hit bypassing and reference-trace sampling—that reduce the execution time overhead required to gather such information. Together, these techniques reduce simulation time by nearly an order of magnitude. For a simple memory simulator and sequential applications, the time to run a program with MemSpy is 3 to 10 times as long as the time to run the program normally. For parallel applications, the overhead increases to factors of 8 to 25. Our experience in using MemSpy to tune several sequential and parallel applications demonstrates that it effectively profiles memory performance at speeds that make it an attractive alternative to other approaches." (p. 34) "For case study purposes, we ran MemSpy with a program called MatMul, which performs a blocked-matrix multiply. Although MatMul is a simple application, it is also a case where the common intuitions about the code's behavior are incorrect; namely, choosing block sizes to fit into the cache is not necessarily sufficient for good performance." *[Editor's note: The article examines a tool that allows programmers to view and analyze program behavior, both through data- and code-oriented statistics. It helps to better understand the causes of cache misses and supports the programmer in his or her tuning task.]*

Predicting Client/Server Availability (p. 41) "The classic availability models that considered only hardware are no longer relevant, since software, operations, and environmental failures are all at least as important as hardware failures. Scheduled maintenance periods are being eliminated, as customers demand continuous application availability. ... In this article, I present a methodology and data to help design engineers evaluate client/server availability. I evaluate client/server outage data, indicate the most important outage causes in a client/server environment, and discuss methods for improving client/server availability." (p. 42) "Surprisingly, we were able to find very few companies that kept good client/server outage