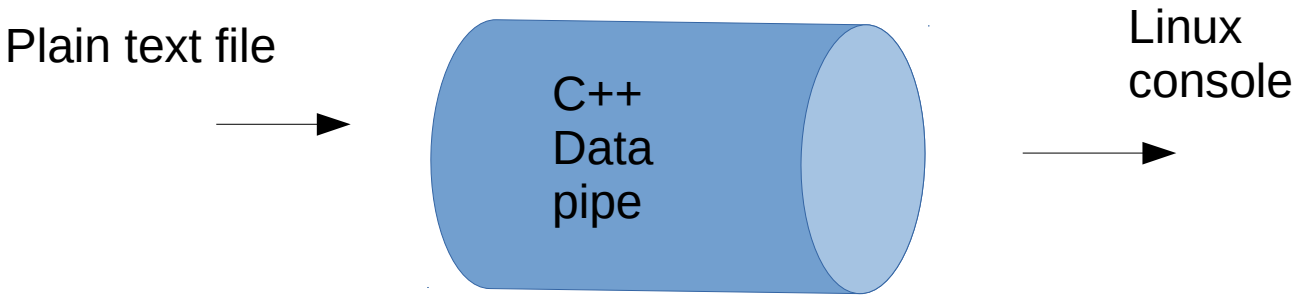

Plain text file
C++
Data
pipe
Linux
console

Possibility to improve the pipeling:

*add stages OO
*Use threads
*Classic coding optimization
*Debugging and testing

*Identify the machine: local? Server? System level DSP?

*Explore the state of the art

Available online at www.sciencedirect.com
ScienceDirect
Electronic Notes in Theoretical Computer Science
ELSEVIER
Electronic Notes in Theoretical Computer Science 238 (2009) 13–23
www.elsevier.com/locate/entcs

Policies of System Level Pipeline Modeling

Ed Harcourt[1]

Department of Mathematics, Computer Science, and Statistics
St. Lawrence University
Canton, NY USA

Abstract

Pipelining is a well understood and often used implementation technique for increasing the performance of a hardware system. We develop several SystemC/C++ modeling techniques that allow us to quickly model, simulate, and evaluate pipelines. We employ a small domain specific language (DSL) based on resource usage patterns that automates the drudgery of boilerplate code needed to configure connectivity in simulation models. The DSL is embedded directly in the host modeling language SystemC/C++. Additionally we develop several techniques for parameterizing a pipeline's behavior based on *policies* of function, communication, and timing (performance modeling).

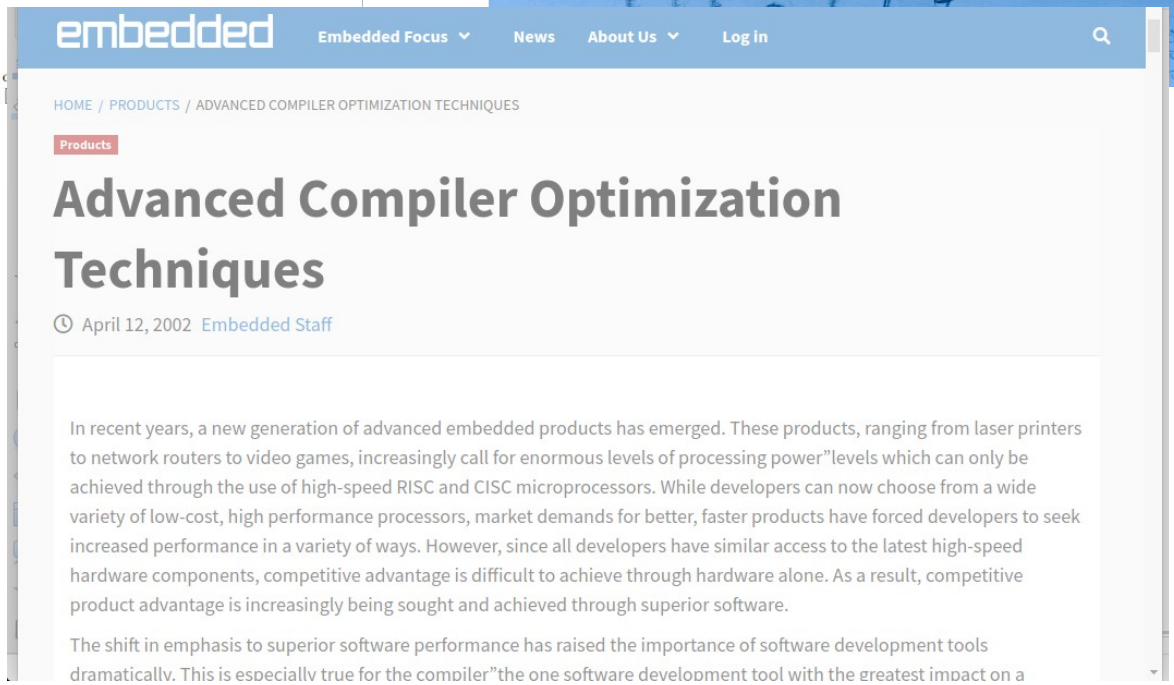
Keywords: pipeline, system level design, discrete-event simulation, generic programming, hardware modeling, policies, SystemC

1 Introduction

Pipelining is a well understood and
creasing the performance of hardware

How to optimize C and C++ code in 2018

Iurii Krasnoshchok Follow
Nov 30, 2018 · 20 min read



embedded Embedded Focus News About Us Log in

HOME / PRODUCTS / ADVANCED COMPILER OPTIMIZATION TECHNIQUES

Products

Advanced Compiler Optimization Techniques

April 12, 2002 Embedded Staff

In recent years, a new generation of advanced embedded products has emerged. These products, ranging from laser printers to network routers to video games, increasingly call for enormous levels of processing power"levels which can only be achieved through the use of high-speed RISC and CISC microprocessors. While developers can now choose from a wide variety of low-cost, high performance processors, market demands for better, faster products have forced developers to seek increased performance in a variety of ways. However, since all developers have similar access to the latest high-speed hardware components, competitive advantage is difficult to achieve through hardware alone. As a result, competitive product advantage is increasingly being sought and achieved through superior software.

The shift in emphasis to superior software performance has raised the importance of software development tools dramatically. This is especially true for the compiler"the one software development tool with the greatest impact on a