Store Architecture in a Persistent Operating System

DAVID HULSE

Thesis submitted for the degree of Doctor of Philosophy
at the University of Adelaide in the Faculty of Engineering

January 1998
Department of Computer Science

Table of Contents

Abstract

Many systems supporting *orthogonal persistence* have been constructed on top of the abstractions provided by conventional operating systems such as Unix. These abstractions are not ideal for the construction of persistent systems since they do not allow the hardware to be used in the most effective manner. This observation motivated the design of *Grasshopper*, a new operating system intended to support persistence as a basic abstraction.

This thesis describes the construction of a persistent environment in which multiple concurrent computations can execute and manipulate data stored in a shared address space. Experiments related to this work were carried out using the Grasshopper operating system as a framework. The persistent computation environment, known as the PCE, comprises two major facets of work. The first is a log-structured persistent store, which is used to hold stable representations of the state of the PCE. Such states include the persistent data stored within the shared address space and the meta-data describing the execution state of the computations. The result is a fully persistent system in which both data and computations may be recovered in the event of a failure.

As computations exchange information through the shared address space, causal dependencies are created between them. To preserve the consistency of the PCE, these dependencies must be preserved across failure. This is the focus of the second facet of work, which concerns the design of a new optimistic checkpointing strategy that guarantees to preserve causal dependencies in the event of a failure. It allows computations to checkpoint independently and uses an algorithm described by Johnson and Zwaenepoel to identify recoverable sets of checkpoints lazily.