Unikernel support for the deployment of light-weight, self-contained, and latency avoiding services

Ward Jaradat, Alan Dearle, and Jonathan Lewis
School of Computer Science, University of St Andrews
{ward.jaradat, alan.dearle, jon.lewis}@st-andrews.ac.uk

Abstract

We have explored dataflow orchestration approaches that permit distributed systems to be constructed by interconnecting services. In such systems latency is often a problem. For example, large data volumes might have to be communicated across the network if computation cannot be co-located close to data sources. Similarly, high latency may be experienced if computation is geographically situated far from (mobile) users. A solution to this problem is the ability to deploy services in appropriate geolocations and wire them together in order to create distributed ecosystems. Such services should ideally be run within a separate protection domain. Hence, we seek to be able to rapidly deploy services in appropriate network locations and dynamically enact and orchestrate them.

Whilst it is safe to deploy units of code dynamically in remote locations using existing technologies, this task is hindered by the size of the deployment units. Current monolithic-based virtual appliances are very large, thus are potentially slow to deploy as they may need to be transmitted across a network.

In order to create a platform on which such services may be based, we are investigating Unikernel approaches. Employing Unikernels permits the services to be treated as first class computational components that can be composed easily and deployed dynamically. Unikernels are compact library operating systems that enable a single application to be linked against the kernel and to be executed in a single-address-space environment. Due to the absence of a large set of software modules that commonly exist in Unix-like operating systems, Unikernels are relatively smaller in size and are potentially more stable and cost-efficient. This research project presents Stardust – a new specialised Unikernel for the dynamic composition of services.

Stardust is designed to run on the Xen hypervisor and has a small codebase that can be maintained easily. Its codebase consists of the kernel which manages the underlying virtual hardware resources presented by the hypervisor. Moreover, Stardust supports modularity and can be customised by integrating software libraries as necessary to extend its functionality. Its kernel and software libraries are linked statically against a particular application or service to provide an immutable and single-purpose virtual appliance that can be deployed onto virtualisation environments. Stardust is designed to support high-level abstractions based on the Java programming language. It supports pre-emptive threading and includes a minimal set of programming libraries, which are required to execute a light-weight open-source Java virtual machine.

Our principles led us to take the route of re-engineering the standard software stack to create self-contained applications that are less-bloated and consequently much smaller. This research aims to engineer networks of distributed systems whose collective behaviour achieve some high-level objectives. Such networks are characterised by the functionality that resonates from nodal interactions instead of the individual nodes that run the computations. This project seeks to introduce strongly-typed communication channels in Stardust as a uniform mechanism to support communication and synchronisation of heterogeneous distributed system components.

This work was presented at the following workshop:
Third Annual UK System Research Challenges Workshop
March 21-23, 2018
http://sysws.org.uk/workshop/2018