Genesis Kernel on IXP1200

Stephen T. Chou, Michael E. Kounavis, Andrew T. Campbell and John Vicente
COMET Group, Columbia University, New York, NY 10027, USA
genesis@comet.columbia.edu

There has been growing interest in network processor technologies that are capable of processing network traffic at line rate. Programming the network processors, in order to install new data paths at run time, is a challenge. A good programming model is determined by physical constraints, performance requirements and complexity. Network processors are architecturally different from general-purpose processors. They are designed to maximize packet-processing throughput by exploiting greater hardware parallelism and hiding memory latency. Yet network processors burden the programmers with architectural details and certain inflexibilities. Issues for programming network processors include: headroom limitations, binding methodology, register assignments and allocations, and instruction store limitations. All of these factors impact the performance as well as the complexity of a programming system for network processor-based routers.

Our model for programming the data path is based on the dynamic binding of components. We choose this model for its flexibility, reusability and manageable complexity. We have designed and implemented NetBind, a binding tool for network processor-based routers. Our dynamic binding scheme uses a code morphing technique that modifies instructions on the fly during the binding. NetBind associates values with “symbols” and “exit points” with proper branching offset at run time. The scheme is also taking advantage of the large register files IXP1200 offers by dividing register sets into groups allowing efficient accessing and sharing of states between processing pipelines. NetBind’s low binding overhead allows the programmer to use components of fine granularity. We demonstrate that we can satisfy the delay and throughput requirements of various types of data paths.

We have used NetBind to create new services and new network architectures on demand by dividing network processor resources between virtual routers operating on the same physical router. We have created overlays of interconnected virtual routers operating over the same physical network forming virtual networks. We have incorporated NetBind intro the Genesis Kernel, a network operating system for creating network architectures. Based on the principles of separation, nesting and inheritance, the Genesis Kernel creates and manages “routelets”. Routelets are autonomous programmable virtual routers with their own separate transport, control and management planes. A routelet is created from a profiling script, which is a blue print that contains state, module, and component binding information. Using the process spawning analogy from operating system, a distinct “child” routelet can be spawned from a “parent” routelet.

NetBind is the dynamic binding system of Genesis, where a “data path constructor” accepts a network processor specification and realizes processing pipelines in the network processor. A data path admission control process maintains records of resource usage. Upon request, the admission control process performs best-fit or exhaustive search bin packing to satisfy specific resource requirements.

Likewise, the control plane is constructed from modular components as aggregate objects. These objects control the spawning process, resource allocations, and the data path. During the spawning process, a script parser accepts a profiling script and passes the information to a “control plane binder”. The control plane binder interacts with a loader to fetch components from a component storage and to compose control planes as a set of aggregate objects. Being able to create modular data paths and modular control planes, the Genesis Kernel offers a flexible programming system for spawning new network services and architectures.

References

[1] Kounavis, M. E., Campbell A. T., Chou, S., Modoux F., Vicente, J., and Zhang H., "The Genesis Kernel: A Programming System for Spawning Network Architectures", *IEEE Journal on Selected Areas in Communication*, Vol. 19, No 3, pg. 511-526, March 2001.

[2] Stachtos, V. D., Kounavis, M. E., Campbell, A. T., " Sphere: A Binding Model and Middleware for Routing Protocols", Fourth International Conference on Open Architecture and Network Programming (OPENARCH'01), Anchorage, Alaska, April 2001

[3] Campbell, A. T., Chou, S. T., Kounavis, M. E., Stachtos, V. D., Vicente, J., "NetBind: A Binding Tool for Constructing Data Paths in Network Processor-Based Router", to be published in Fifth International Conference on Open Architecture and Network Programming (OPENARCH'02), New York, New York, June 2002.