A Scalable and Reconfigurable Shared-Memory Graphics Architecture

Michael Manzke*
Trinity College Dublin

Ross Brennan
Trinity College Dublin

Keith O'Conor
Trinity College Dublin

John Dingliana
Trinity College Dublin

Carol O'Sullivan
Trinity College Dublin

Current scalable high-performance graphics systems are either constructed using special purpose graphics acceleration hardware or built as a cluster of commodity components with a software infrastructure that exploits multiple graphics cards [Humphreys et al. 2002]. Both these solutions are used in application domains where computational demand cannot be met by a single commodity graphics card e.g., large-scale scientific visualisation. The former approach tends to provide the highest performance but is expensive because it requires frequent redesign of the special purpose graphics acceleration hardware in order to maintain a performance advantage over the commodity graphics hardware used in the cluster approach. The latter approach, while more affordable and scalable, has intrinsic performance drawbacks due to computationally expensive communication between the individual graphics pipelines.
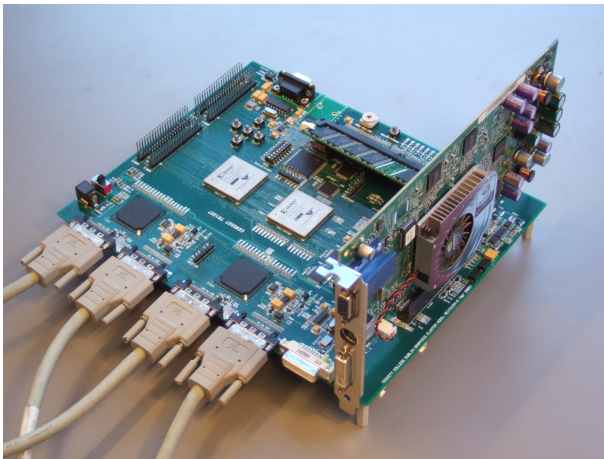

Figure 1: The first prototype of the custom-built high-performance graphics cluster node. The figure shows how a commodity graphics card interfaces the cluster node. It also depicts the four SCI cables that should interconnect the custom-built GPU interface boards and the PC cluster via a 2D torus topology.

In this sketch we propose a scalable tightly coupled cluster of custom-built boards that provide an AGP interface for commodity graphics accelerators. This hybrid solution aims to bridge the gap between both of the current solutions, offering a minimal custom-built hardware component together with a novel and efficient shared memory infrastructure that exploits cutting-edge consumer graphics hardware. The boards are supplied with rendering instructions by a cluster of commodity PCs that execute OpenGL graphics applications. All the commodity PCs and custom-built boards are interconnected with an implementation of the IEEE 1596-1992 Scalable Coherent Interface (SCI) standard. This technology provides the system with a high bandwidth, low latency, point to point interconnect. Our design allows for the implementation of a 2D torus topology with good scalability properties and excellent suitability for parallel rendering. Most importantly the interconnect implements a Distributed Shared Memory (DSM) architecture in hard-

ware. Figure 2 shows how local memories on the custom-built boards and the PCs become part of the system wide DSM through the SCI interconnect. Figure 2 also depicts Field Programmable Gate Arrays (FPGAs) on the custom-built boards. These reconfigurable components assist the SCI implementation and provide substantial additional computational resources that may be used to control the commodity graphics accelerators and to perform operations associated with a parallel rendering infrastructure. Beyond the previously mentioned application of the FPGAs we envision other graphics application related computation e.g., ray tracing. These reconfigurable components are an integral part of the scalable shared-memory graphics cluster and consequently increase the programmability of the parallel rendering system, just like vertex and pixel shaders increased the programmability of graphics pipelines.
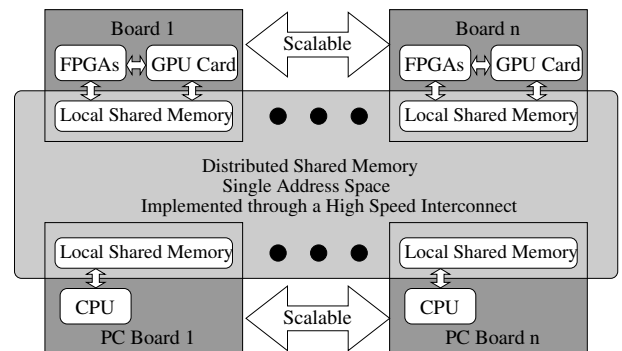

Figure 2: Shared memory system.

In this sketch, we describe the design of a tightly coupled scalable Non-Uniform Memory Access (NUMA) architecture of distributed FPGAs, GPUs and memory that may be constructed with a limited amount of custom-built hardware. A first prototype of the custom-built boards, seen in Figure 1, was manufactured and is currently debugged. A second revision will resolve outstanding problems. We expect that this hardware DSM cluster communicates data at 500Mbytes/s with low latencies ($< 1.5~\mu$s). This hard real-time capable parallel rendering cluster will be connected with the same high speed interconnect to a commodity PC cluster that will execute the graphics application. We have introduced this novel architecture and estimate, based on the arguments presented, that this solution could out-perform pure commodity implementations without increased hardware cost and yet maintain its adaptablity to the most recent generation of commodity graphics accelerators and target applications. Later prototypes will incorporate PCI Express to be compatible with the latest commodity graphics accelerators.

References

Humphreys, G., Houston, M., Ng, R., Frank, R., Ahern, S., Kirchner, P. D., and Klosowski, J. T. 2002. Chromium: a stream-processing framework for interactive rendering on clusters. In *SIGGRAPH*, 693–702.

*e-mail:Michael.Manzke@cs.tcd.ie