The Evaluation of GPU-Based Programming Environments for Knowledge Discovery

John Johnson, Randall Frank, and ***Sheila Vaidya***
Lawrence Livermore National Labs
Phone: 925-424-4092
Email Addresses: {jjohnson, fjfrank, vaidya1}@llnl.gov

Introduction

Revolutionary advances in computer graphics technologies, driven by the needs of 3D gaming, have resulted in specialized SIMD floating point rendering engines known as GPUs. These GPUs are programmed via graphics libraries such as OpenGL, but have very general programming architectures. These cards are handily exceeding Moore's law performance predictions and are expected to continue to do so for some time. The size and cost competitive nature of the gaming industry combine to make these systems extremely affordable. Today, GPUs with over 40GF can be bought for around $300 and they are expected to increase to around 1000GF for about that same cost by the 2005 timeframe. These systems form the core of distributed interactive systems but can also be applied to many processes other than rendering and visualization. At present, the non-visualization uses of these systems have been limited to classically streaming or vector floating point bound processes.

We will present early results in the use of these GPU systems to perform computations on alternative types of algorithms that are not traditionally FLOP bound, such as those utilized in video image processing, text processing and semantic graph traversal and analysis. Knowledge discovery based application areas should, minimally, benefit from the extreme memory bandwidths present on GPU systems (over 23GB/sec in current systems), and are in a position to exploit the FLOP rich GPU environment to enhance the fidelity and complexity of their computation. Some of our early studies have already shown orders of magnitude performance speedup for specific applications.

The GPU Based Compute Platform

Two advantages of GPUs are their extremely high memory bandwidth and their unique gather capabilities. We are investigating applications that exploit both of these features. As a key first step we are investigating the mapping of data onto the current GPU architectures via pointer-less indirection techniques and implicit parallel storage techniques. The design is expected to draw from recent work on tiled, paged boundary conditions on GPU systems. The initial targets are temporal image processing algorithms commonly used in the processing of data like surveillance video and facial biometrics. Basic algorithms for filtering and feature detection and tracking are being implemented and demonstrated to apply to large, parallel data streams.

One of the difficulties in the scaling and parallelization of algorithms on GPU systems stems from the very nature of the data structures. Following the image processing work, will focus on non-traditional HPC data structures. In the next several months we expect to investigate the

applicability of GPU system to string and list processing functions. These have been difficult to map onto streaming processing systems, but recent advances in pixel shader technology would suggest that it may be possible to perform hundreds of parallel text searches in parallel in a streaming, multi-pass GPU architecture. We intend to exploit similar advances in texture fetching operations to investigate the use of GPUs in pointer-less list searching and comparison problems. These research advances hold the potential of allowing these systems to be applied to other data mining problems and the processing of transaction orientated data, such as the analysis of web traffic or semantic graphs.

The GPU enhanced system is not a static target and tremendous advances are announced on nearly a bi-annual basis by vendors. Additionally, it will be useful to compare results from these GPU based systems with the results from other architectures that are being developed in parallel (e.g. BlueGene/L and Merrimac). In parallel with the basic algorithmic efforts, we are performing research into the integration of this work with higher level semantic languages with multiple system targets. The integration this work, in particular the non-traditional data structures for strings and lists into streaming languages such as Brook will allow the work to target a number of other real or simulated architectures. As a result, virtual performance comparisons can be made with these architectures. As efforts in this space progress, the model will be adapted to next generation graphics architectures such as upcoming future architectures such as the proposed “Cell” based systems.