Abstract

The API interfaces provided by CUDA help programmers to get high performance CUDA applications in GPU, but they cannot support most I/O operations in device codes. The characteristics of CUDA’s mapped memory are used here to create a dynamic polling service model in the host which can satisfy most I/O functions such as read/write file and "printf". The technique to implement these I/O functions has some influence on the performance of the original applications. These functions quickly respond to the users’ I/O requirements with the "printf" performance better than CUDA’s. An easy and effective real-time method is given for users to debug their programs using the I/O
functions. These functions improve productivity of converting legacy C/C++ codes to CUDA and broaden CUDA’s functions.

References


Index Terms

Computer Science

Circuits And Systems

Keywords
CUDA's Introduction  Architecture  Input-output Functions  Mapping Of Memory.