

Atualmente, o uso, para processamento de propósito geral, das chamadas Graphics Processing Units (GPUs), presentes nas placas de vídeo modernas, vem chamando a atenção de vários pesquisadores. Entretanto, pouca atenção tem sido dada em relação a comparações de desempenho entre APIs tradicionais de processamento paralelo com APIs que exploram os elementos de processamento das GPUs, considerando, além de métricas convencionais de desempenho, métricas pertencentes ao contexto de sistemas embarcados. Com o intuito de preencher esta lacuna, adotou-se, como metodologia de desenvolvimento, a pesquisa de diversas aplicações reais, relevantes para o contexto de sistemas embarcados e com granularidade grossa, ou seja, com grande quantidade de computação em comparação com comunicação, e o estudo de seis APIs, três de cada uma das duas classes citadas (APIs convencionais e APIs para GPUs). A partir de uma análise preliminar, com base em quesitos como programabilidade (e suas componentes, entre elas, baixo tempo de desenvolvimento e suporte a diferentes níveis de abstração) e portabilidade, foram escolhidas representantes promissoras para cada uma das classes, visando uma comparação detalhada usando uma das aplicações reais investigadas. Espera-se que, até o final deste trabalho, sejam encontradas respostas a duas perguntas fundamentais: mesmo com o overhead do processo de escrita e leitura na memória da GPU, será mais eficiente utilizá-la para processamento de propósito geral massivo ou o aumento na quantidade de núcleos em processadores multicore será a melhor alternativa para melhorar o desempenho em aplicações onde questões de consumo energético e de memória, por exemplo, são importantes? Haverá uma maneira de isolar a GPU, de modo que ela não dependa de uma CPU, permitindo assim o seu emprego como suporte de hardware para as atuais e complexas aplicações embarcadas?