**Evaluating Graph Coloring on GPUs**

A.V. Pascal Grosset, Peihong Zhu, Shusen Liu, Suresh Venkatasubramanian, Mary Hall

**Graph Coloring**
- Color a graph using as few colors as possible
- Vertices connected by an edge must have different colors

**Sequential Graph Coloring Algorithm**
- First Fit: $O(n)$
- SDO & LDO: $O(m \log n)$
  - $n$: number of vertices
  - $m$: number of edges

**Graph Coloring Framework**
- Graph Partitioning
- Graph Coloring & Conflict Solving

**Sequential Conflicts Resolution**
- Residual conflicts are solved sequentially

**Graph Partitioning**
- Evenly distribute vertices per thread into subgraphs

**Graph Coloring & Conflict Solving**
- Threads choose vertex to color in their subgraph but check whole graph
- Several passes of coloring and conflict solving are done

**Algorithms I**
- First Fit: Allocate colors randomly
- SDO: Highest Saturation and then Highest degree

**Algorithms II**
- MAX OUT: Most Neighbors outside the subgraph and then highest degree
- MIN OUT: Least Neighbors outside the subgraph and then highest degree

**Results**
- Best Colors: MIN OUT & MAX OUT generally
- Fastest: Parallel First Fit is very fast and provides much better color than the sequential First Fit

**Conclusion**
- Graph coloring on GPU gives few colors at good speed
- Even algorithms like First Fit are better as GPU are more forgiving