Graph Prefetching Using Data Structure Knowledge

Sam Ainsworth, Timothy M. Jones

Presented by: Iris Bao, Jiongsheng Cai, Yate Ji, Wenyi Wu

Motivation

- Cache miss rates are high for graph loads
 - Graph access patterns tend to be data-dependent and non-sequential
 - Accesses aren't inherently embarrassingly parallel
- Prefetching graph data for well defined access pattern such as breadth-, depth- first search can improve performance
- We demonstrate software and hardware prefetch for breadth-first search on CSR (Compressed Sparse Row) formatted graphs

CSR: Compressed Sparse Row





(b) CSR breadth-first search

(a) Graph

Software: Cache misses and stalls



Software: Prefetch Edge List



Software: Breadth First Search



Hardware: Configurable Graph Prefetcher



(a) System overview



(b) Prefetcher microarchitecture detail

Scheduling Prefetches

• Basic operation:

Ο

- o visited[edgeList[vertexList[workList[n+o]]]]
- Prefetch distances (offset o):
 - *o* * *work_list_time* = *data_time*
- Make use of runtime data:

$$avg_time_{new} = rac{new_time + (lpha - 1)avg_time_{old}}{lpha}$$

- $\circ \quad \alpha = 8 \text{ for } work_list_time$
- \circ α = 16 for *data_time*



Vertex-Offset Mode

• data_time > work_list_time

• $o = 1 + \frac{k * data_time}{work_list_time}$



Vertex-Offset Mode

Observation	Action
Load from workList[n]	Prefetch workList[n+o]
Prefetch vid = workList[n]	Prefetch vertexList[vid]
Prefetch from vertexList[vid]	Prefetch edgeList[vertexList[vid]] to edgeList[vertexList[vid+1]] (12 lines max)
Prefetch vid = edgeList[eid]	Prefetch visited[vid]

Large-Vertex Mode

- data_time < work_list_time
- *firstLine* = *edgeList[idx* + 14**lineSize]*

Observation	Action
Prefetch vid = workList[n]	Prefetch vertexList[vid]
Prefetch eid = vertexList[vid]	Prefetch edgeList[eid] to edgeList[eid + 8*lineSize - 1]
Load from edgeList[eid] where (eid % (4*lineSize)) == 0	Prefetch edgeList[eid + 4*lineSize] to edgeList[eid + 8*lineSize - 1]
Prefetch vid = edgeList[eid]	Prefetch visited[vid]
Prefetch edgeList[vertexList[vid+1]]	Prefetch workList[n+1]

Large-Vertex Mode

Performance



Performance



Performance

- Can be used to accelerate other traversals on CSR graphs
 - Parallel Breadth-First Search
 - Sequential Iteration Prefetching (PageRank, Sequential Coloring)



Average speedups:

2.2× for bfs-based algorithms2.4× for sequential iteration

access patterns

Questions?