Introduction

Related work

Datatype Engine:

1. Commit time optimization
2. Master’s internal datatype representation
3. Iovec datatype representation

Datatype representation pros and cons:

1. Master
   1. Very efficient when datatype can be optimized (vector graph)
   2. When optimization does not work, performance can fall drastically (comparison graph)
   3. Very hard to perform prefetches since traverse datatype tree takes a lot coding effort
2. Iovec
   1. Stable performance (comparison graph)
   2. Better performance for scattered irregular memory patterns (datatype that cannot be optimized) (random datatype graph)
   3. large representation size lead to low performance for optimized datatypes (vector grpah)
   4. easy to prefetch since each prefetch can be issued for each address in iovec

Prefetch:

* prefetch performed in iovec representation
  + prefetch is issued for every iovec
  + prefetch is issued only once for every cacheline
* reduces TLB miss for large stride memory layout large size buffers thus leads to better performance

Performance graphs:

* comparison graph (optimization need to be better to recognize where to combine contiguous memcpys)
* vector graph (stride varied with papi TLB miss counters)
* matrix graph (diagonal and sparse matrix)