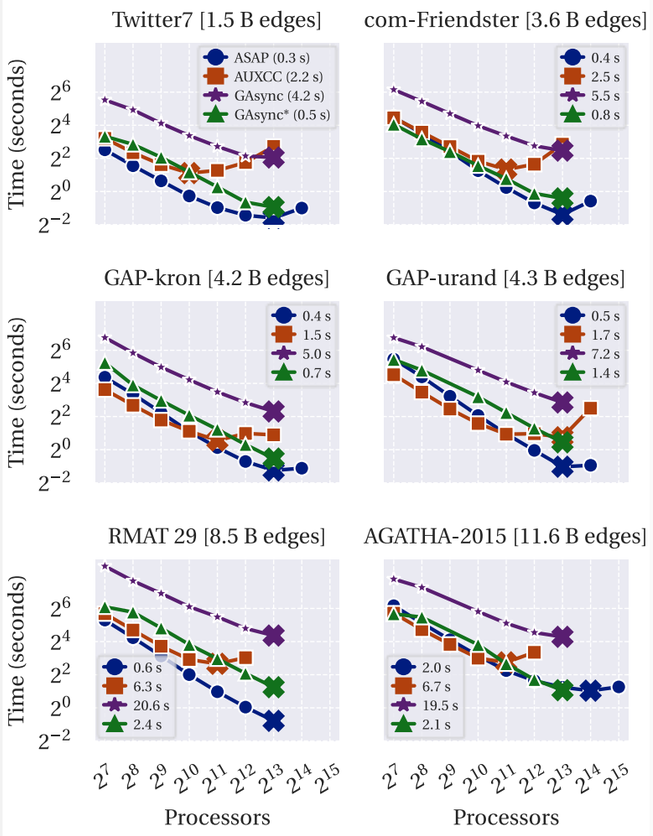
Fast to Write. Faster to Run.

A Fused Collective for Irregular Distributed-Memory Applications

Graph Analysis
Connected Components

1.8x faster than two-sided AUXCC and one-sided GluonAsync (GAsync*).

8.8× faster than GluonAsync end-to-end by avoiding graph partitioning.

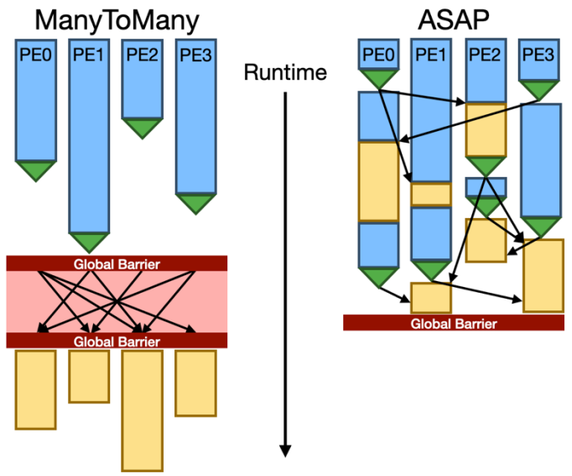


Bonus: Full k-core decomposition of hyperlink2012 graph: **1.7×** faster than optimized MPI baseline.

ASAP: AllScatterApply



4× faster than optimized two-sided MPI collectives.



Up to 74× faster than raw one-sided HClib Actor.

k-mer counting using ManyToMany (left) and ASAP (right)

```python
def KmerCounting(R(n,m), k, Batch):
  SendBufVec = [[] for _ in range(P)]
  RecvBuf = []; N = 0
  for i in range(n):
    for j in range(k, m):
      kmer = GenerateKMER(R[i][j], k)
      SendBufVec[OwnerPE(kmer,P)].add(kmer)
      if (N:= N+1) % Batch == 0:
        WAIT() # Barrier for last ManyToMany to finish
        ExchangeKMER(SendBufVec, RecvBuf)
  WAIT()
  return Compact(RecvBuf, Merge())

def ExchangeKMER(SendBufVec, RecvBuf):
  SendBuf = []
  for i in range(P):
    Compact(SendBufVec[i], Merge())
    SendBuf.add(SendBufVec[i])
  RecvBuf = NonBlocking-ManyToMany(SendBuf)
```

```python
def KmerCounting(R(n,m), k):
  RecvBuf = []
  CommObj = ASAP(RecvBuf, Merge(), Receive(), OwnerPE())
  for i in range(n):
    for j in range(k, m):
      kmer = GenerateKMER(R[i][j], k)
      CommObj.Push(Pkt(key=kmer, val=1))
  CommObj.Done()
  GLOBAL BARRIER
  return CommObj.Compact(RecvBuf, Merge())

struct Pkt:
  key  # k-mer in integer encoding
  val  # partial count of key

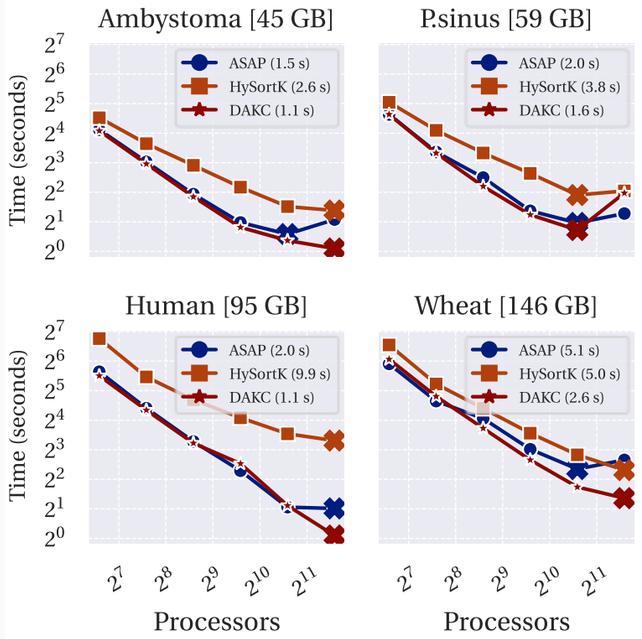
def Receive(Pkt A, Buffer):
  Buffer.add(A)

def Merge(Pkt A, Pkt B):
  if A.key == B.key: A.val += B.val
  return A
```

Genomics
k-mer Counting

2× faster than HySortK (OpenMP + non-blocking MPI collectives).

Matches the performance of hand-tuned DAKC (HClib Actor)



Souvadra Hati
Richard Vuduc
Georgia Institute of Technology

Acknowledgement: We thank Kalsuda Lapborisuth and Maxim Zhulin for their respective work on the development and evaluation of BSP CC baselines. This material is based in part upon work supported by the U.S. National Science Foundation (NSF) Award Number 2316177. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.