A Parallel Streaming Framework for Multi-Processing-Unit Systems

Presenter: Huy Vo

Huy Vo, Brian Summa, Valerio Pascucci, Claudio Silva
SCI Institute, University of Utah, USA

Daniel Osmari, Joao Comba
UFRGS, Brazil

Motivation

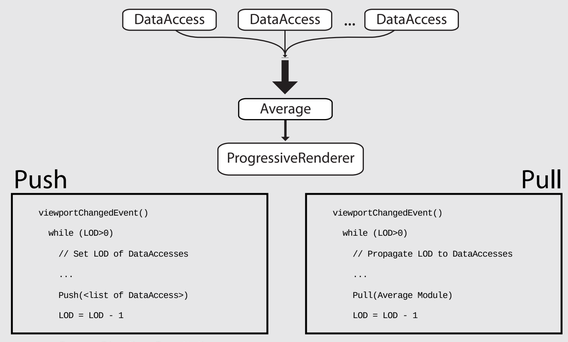
Current dataflow system problems (with VTK, AVS, Data Explorer,...):

- API "polluted" over the years (many designs in 80s-90s)
- Do not map well to modern highly-parallel architecture: Multi-Core, GPUs, ...
- Large datasets not well supported

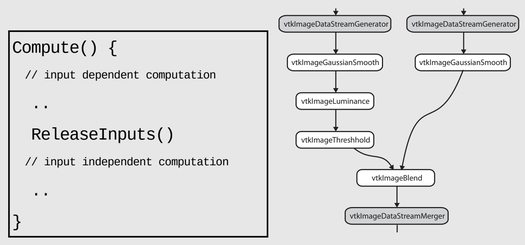
Need a unified dataflow architecture supporting:

- Streaming
- Heterogeneous architecture
- Efficiency
- Concurrency

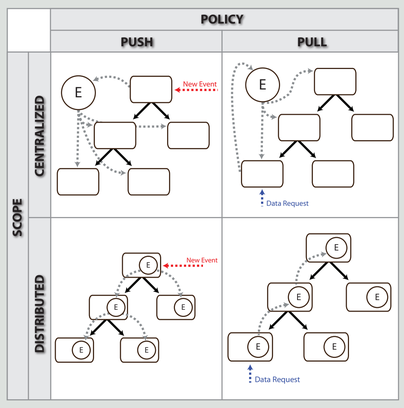
Pull(), Push() and ReleaseInputs()

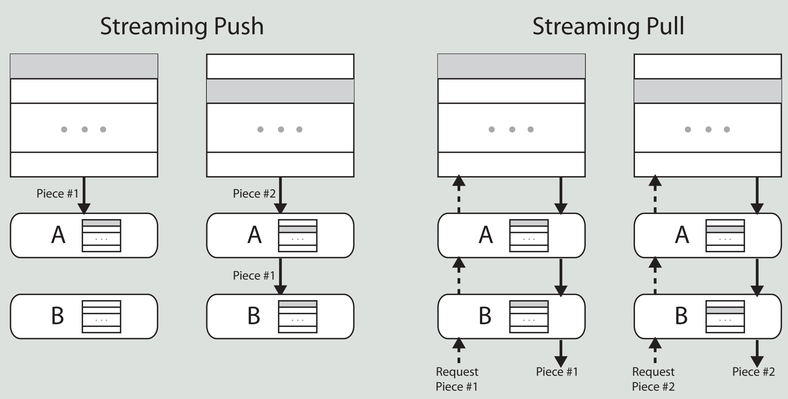


- Better Pipeline Parallelism
- Simpler Interface

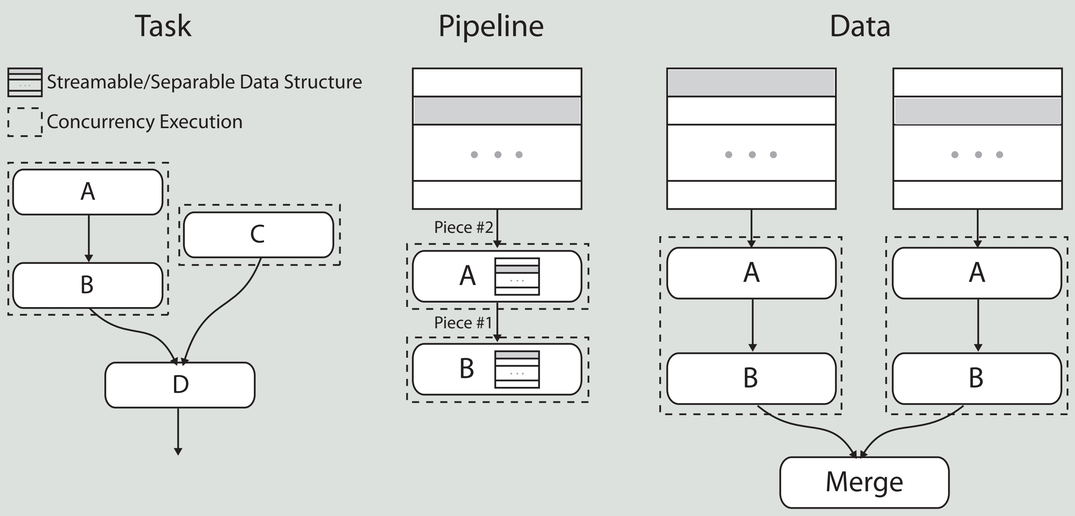


Classification of Dataflow Streaming Model

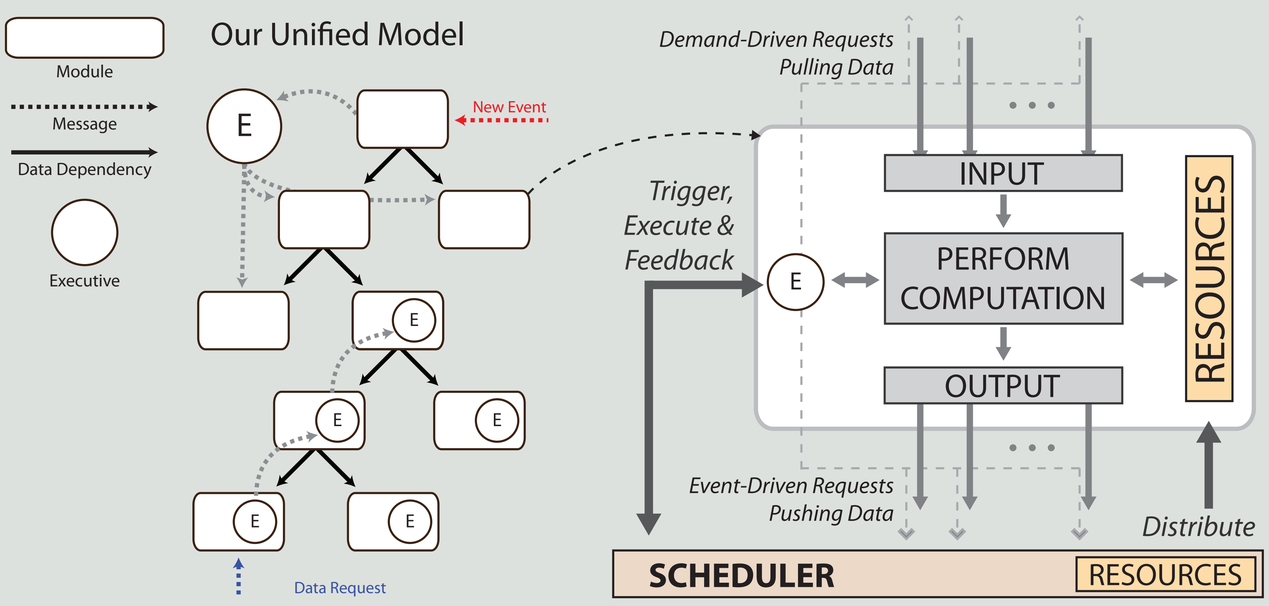




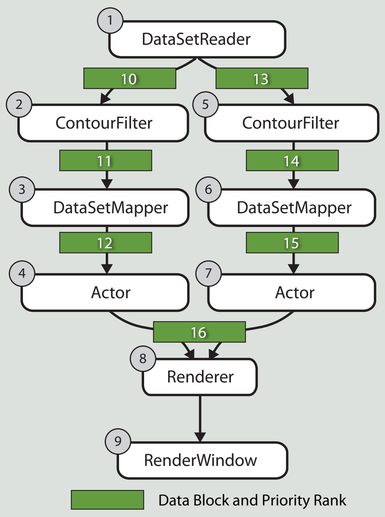
Execution Parallelism



System Architecture



Task Scheduler & Resource Distribution



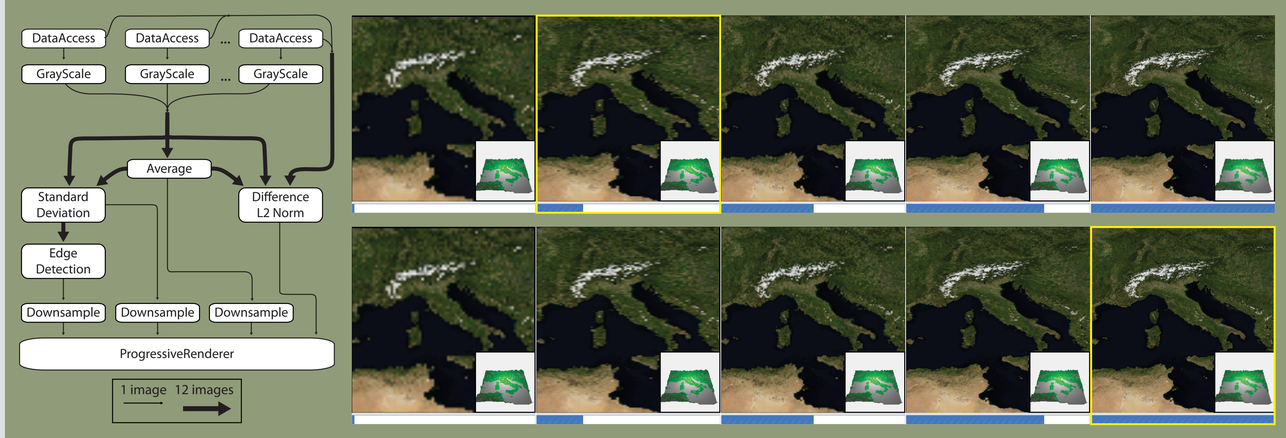
- Use a priority queue with topological order of the modules as priority ranks
- Include streaming piece numbers to priority ranks
- Resources distributing strategy is based on execution time
- Maximize concurrency but still taking data dependency into account

Preliminary Results

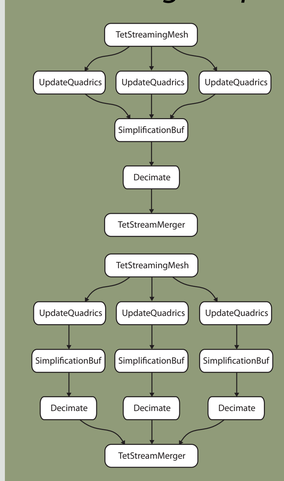
Complete Implementation for CPU Thread Resources

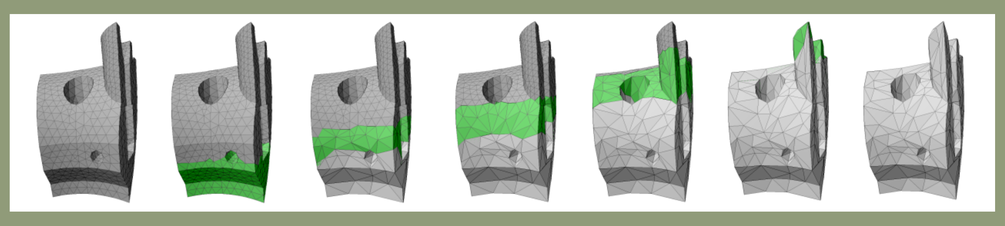
Multi-View Visualization

~3x speed up on 4-core machine



Streaming Simplification of Tetrahedral Meshes



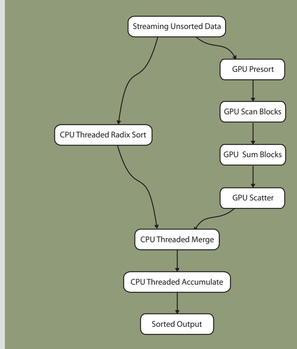


Streaming Simplication of Tetrahedral Meshes					
Models (Tets)		Original	Streaming	Quadric Inst.	Pipeline Inst.
Torso	1.0M	8.5s	8.4s	7.9s	3.0s
Fighter	1.4M	10.1s	9.5s	8.4s	3.6s
Rbl	3.8M	37.4s	35.6s	31.5s	8.9s
Mito	5.5M	52.0s	53.1s	44.9s	10.2s

Basic Concrete Implementation of CUDA execution for GPU Resources

- Connections can direct data transfer to appropriate cudaMemcpy() calls
- A custom scheduling strategy to minimize host-device data transfer

CPU/GPU Sort



- Pipelining GPU branch with separate CUDA devices
- Load-balancing between CPU/GPU is converged over streaming time
- Achieved a throughput of 250M pairs/s on (2) Xeon w5580 and a Quadro FX5800 gives
- Achieved a throughput over 350M pairs/s on (2) Xeon e5550 and (2) Tesla C1060s

Ongoing Progress

- Dynamic Load-Balancing for GPU kernels
- Concrete implementation for MPI to run on cluster
- Incorporate TBB/OpenMP for Task Scheduling
- A generalized streaming data structure

References

"Parallel Dataflow Scheme for Streaming (Un)Structured Data" by H. Vo, D. Osmari, B. Summa, J. Comba, V. Pascucci and C. Silva, SCI Institute, University of Utah, Technical Report #UUSCI-2009-004, 2009.

"Multi-Threaded Streaming Pipeline For VTK" by H. Vo and C. Silva, SCI Institute, University of Utah, Technical Report #UUSCI-2009-005, 2009.