Motivation: Multi-grain Software and Hardware
- Complex, large-scale parallel applications have inherent granularity
- Large, distributed cluster supercomputers have inherent granularity
- Modern multicore processors such as chip multiprocessors (CMPs)
- Accelerators
  - GPUs
- Cell Broadband Engine
- Hierarchy due to mixed node types, mixed processor types
- Difficult application to hardware mapping
- Current programming systems flatten hierarchy
- Incomplete handling of multiple levels of granularity
- Trial-and-error optimization of important application parameters
- Filter-stream programming framework excellent for multi-grain
- Component-based, for best task compartmentalization
- Data-driven, for easy application development
- Ensures efficient application/hardware granularities

Coarse-grain Development with GPUs
- Coarse-grain CMP/GPU cluster utilized for biomedical image analysis
- Neuroblastoma a childhood cancer
- Prognosis based partially on digitized microscope tissue slides analysis
- Computerized prognosis system needs to analyze up to 30 GB images

Coarse-grain Biomedical Application Results
- Processing nodes consist of:
  - Dual-socket 2.4GHz dual-core AMD Opteron
  - 2 NVIDIA Quadro 5600 GPUs
- Dual GPU time under one minute for largest image (excluding overheads)
- 46.5x speedup 1 GPU vs sequential C++
- 926x speedup 16-node GPU vs 1-node C++
- LARGE image (109,119 x 80,812 pixels) analyzed in under 12 seconds

Fine-grain Filter-stream Framework for the Cell Broadband Engine
- Cell Intercore Messaging Library
- High performance messaging library
- Two-sided communication semantics
- Maintains most of Cell’s bandwidth
- DataCutter-Lite
- Fine-grain component-based filter-stream framework
- Event-based filter-stream programming
- Automatic multi-buffering of data
- Simple multi-threaded, heterogeneous application development

Cell Broadband Engine Intercore Messaging Library (CIML)
- Component-based, filter-stream programming framework
- Define computation as task-graph
- Tasks are filters, which are functions which compute
- Data flows along streams to/from filters along pre-defined paths
- Automatic multi-buffer of data
- Automatic PPE-SPE, inter-SPE communication
- DCL is event-based
- Arrival of stream buffer (quantum of data) triggers filter execution
- Compared against custom IBM SDK version
- 32 1Kx1K image tiles
- Overheads included: DCL takes 23-57% longer
- Overheads excluded: SDK takes 5-26% longer

DataCutter-Lite: Fine-grain Filter-stream Programming Framework and Runtime Engine
- Component-based, filter-stream programming framework
- Define computation as task-graph
- Tasks are filters, which are functions which compute
- Data flows along streams to/from filters along pre-defined paths
- Automatic multi-buffer of data
- Automatic PPE-SPE, inter-SPE communication
- DCL is event-based
- Arrival of stream buffer (quantum of data) triggers filter execution
- Fine-grain component-based filter-stream
- Data flows along streams to/from filters along pre-defined paths
- Automatic multi-buffer of data
- Automatic PPE-SPE, inter-SPE communication
- DCL is event-based
- Arrival of stream buffer (quantum of data) triggers filter execution
-Fine-grain Biomedical Application Results
- Compared against custom IBM SDK version
- 32 1Kx1K image tiles
- Overheads included: DCL takes 23-57% longer
- Overheads excluded: SDK takes 5-26% longer

Multi-grain Biomedical Application Results
- Complex, large-scale parallel applications have inherent granularity
- Large, distributed cluster supercomputers have inherent granularity
- Modern multicore processors such as chip multiprocessors (CMPs)
- Accelerators
  - GPUs
- Cell Broadband Engine
- Hierarchy due to mixed node types, mixed processor types
- Difficult application to hardware mapping
- Current programming systems flatten hierarchy
- Incomplete handling of multiple levels of granularity
- Trial-and-error optimization of important application parameters
- Filter-stream programming framework excellent for multi-grain
- Component-based, for best task compartmentalization
- Data-driven, for easy application development
- Ensures efficient application/hardware granularities

Future Work: Filter-stream Runtime Systems
- Heterogeneity and hierarchy make scheduling problem difficult
- Algorithms needed to deal with task scheduling at all hardware and software granularity levels
- Inter-node, fine-grain tasks affect node performance, network performance, etc.
- Intra-node, coarse-grain tasks affect data availability, network endpoint contention, etc.
- Small memories of Cell, deep-cache hierarchy CMPs introduce new scheduling constraints
- Heart of future runtime systems, programming frameworks will be intelligent task scheduling
- Optimize:
  - Execution time (makespan)
  - Resource utilization

Future Work: Filter-stream Task Scheduling
- Heterogeneity and hierarchy make scheduling problem difficult
- Algorithms needed to deal with task scheduling at all hardware and software granularity levels
- Inter-node, fine-grain tasks affect node performance, network performance, etc.
- Intra-node, coarse-grain tasks affect data availability, network endpoint contention, etc.
- Small memories of Cell, deep-cache hierarchy CMPs introduce new scheduling constraints
- Heart of future runtime systems, programming frameworks will be intelligent task scheduling
- Optimize:
  - Execution time (makespan)
  - Resource utilization

Biomedical Informatics
http://bmi.osu.edu