

Architectures, Languages, and Compilers for the Streaming Domain

Saman Amarasinghe and William Thies
Massachusetts Institute of Technology

PACT 2003
September 27, 2003

Acknowledgements

- Material contributed by
 - Mark Horowitz – Stanford
 - Bill Dally – Stanford
 - Christos Kozyrakis – Berkeley/Stanford
 - Steve Keckler – UT Austin
 - Ed Lee – Berkeley
 - Peter Mattson – Reservoir Labs
 - Bill Mark – UT Austin
 - Robert Bond – MIT Lincoln Labs
 - Vivek Sarkar – IBM Yorktown Heights
 - Jinwoo Suh – USC/ISI

Web Resources

- <http://cag.lcs.mit.edu/streamit>
- <http://cag.lcs.mit.edu/raw>
- <http://cag.lcs.mit.edu/wss03>
- <http://www.morphware.org>

Schedule

1:30-1:40	Overview (Saman)
1:40-2:20	Stream Architectures (Saman)
2:20-3:00	Stream Languages (Bill)
3:00-3:30	Break
3:30-3:55	Stream Compilers (Saman)
3:55-4:20	Domain-specific Optimizations (Saman)
4:20-5:00	Scheduling Algorithms (Bill)

The Streaming Domain

- Widely applicable and increasingly prevalent
 - Embedded systems
 - Cell phones, handheld computers, DSP's
 - Desktop applications
 - Streaming media
 - Software radio
 - Real-time encryption
 - Graphics packages
 - High-performance servers
 - Software routers (Example: Click)
 - Cell phone base stations
 - HDTV editing consoles
 - Radar tracking
 - Streaming databases
- Based on audio, video, or data stream
 - Predominant data types in the current data explosion

What is stream processing?

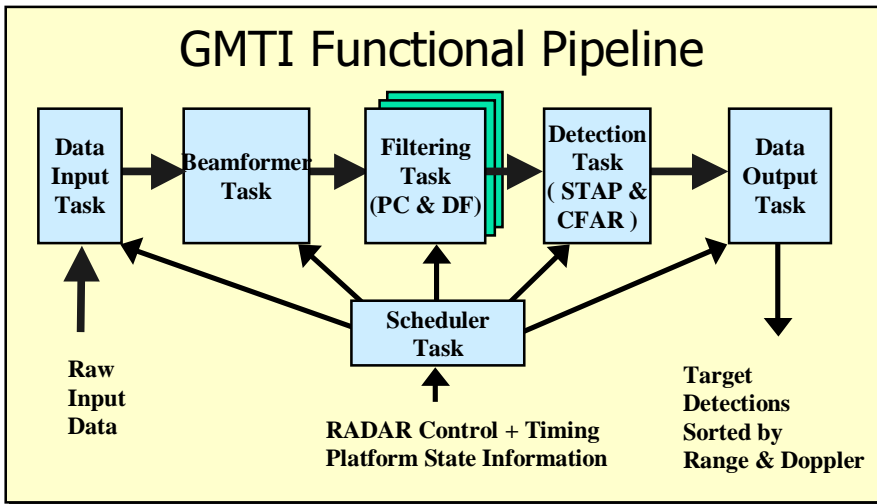
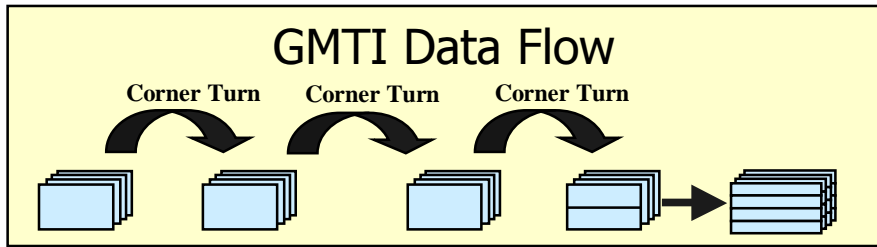
- "A model that uses sequences of data and computation kernels to expose and exploit concurrency and locality for efficiency."

Properties of Stream Programs

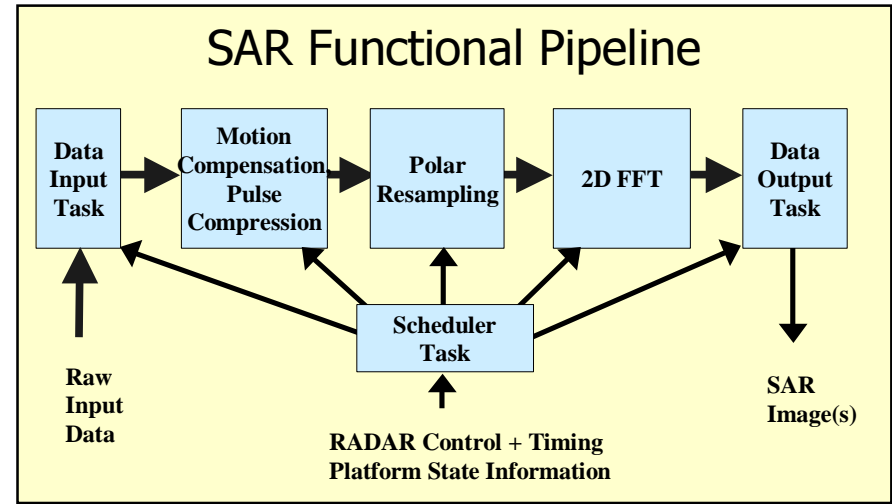
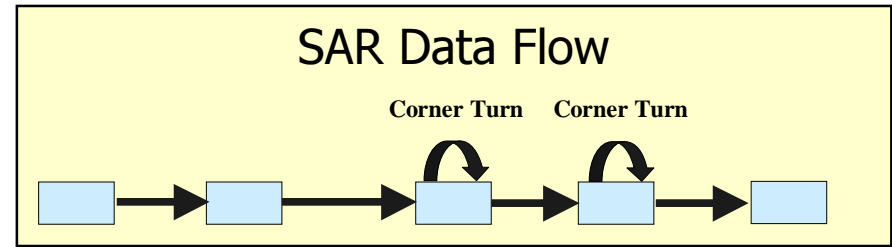
- A large (possibly infinite) amount of data
 - Limited lifetime of each data item
 - Little processing of each data item
- Computation: apply multiple filters to data
 - Each filter takes an input stream, does some processing, and produces an output stream
 - Filters are independent and self-contained
- A regular, static computation pattern
 - Filter graph is relatively constant
 - A lot of opportunities for compiler optimizations

Radar Stream Signal Processing Algorithms

GMTI (STAP) Algorithm



SAR Algorithm

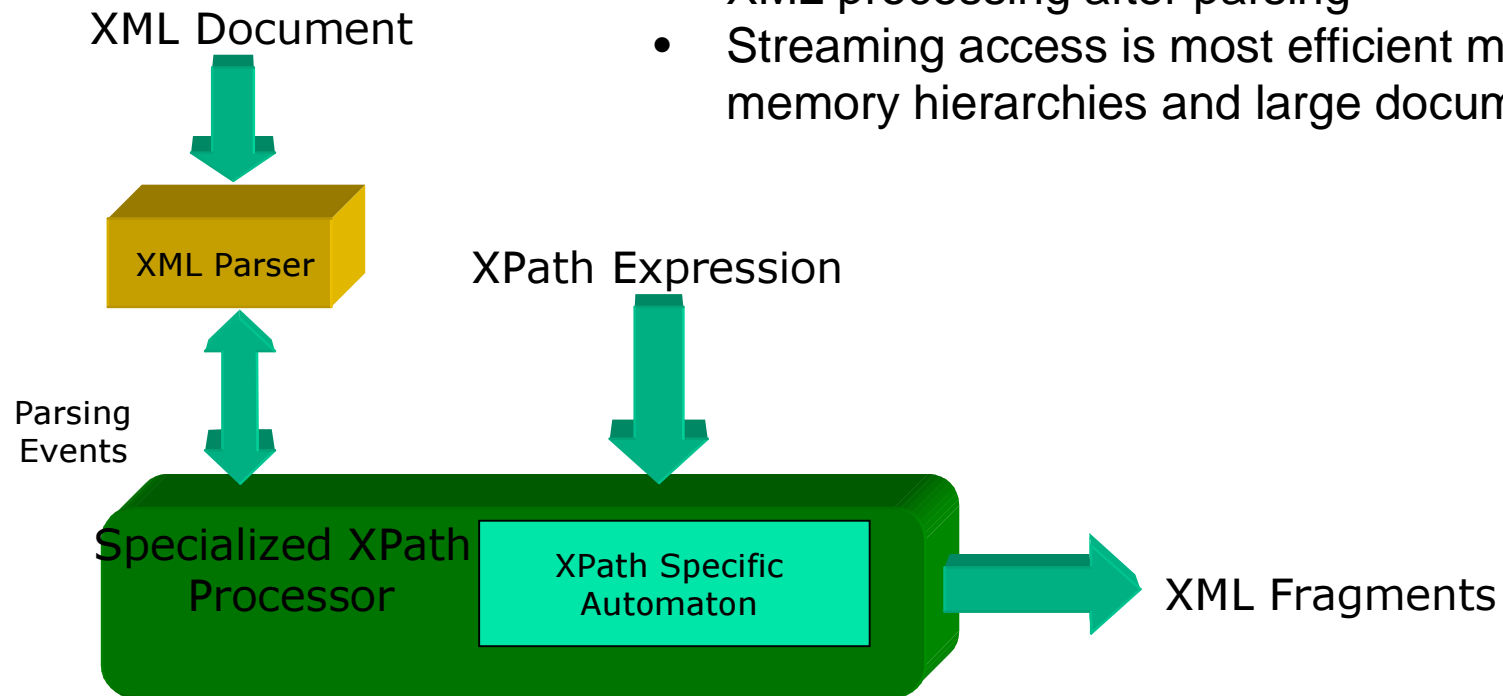


Synthetic Aperture Radar (SAR):	~ 100-1000 GOPS	~ 1-100 s latency
Ground Moving Target Indicator (GMTI)	~ 100-1000 GOPS	~ 0.1-1s latency
Space-Time Adaptive Processing (STAP)	~ 1-100 GOPS	~ 0.01- 0.1s latency

XML Processing: Streaming Approach

Motivation:

- XPath lookup is next major bottleneck in XML processing after parsing
- Streaming access is most efficient mode for memory hierarchies and large documents



XAOS (XPath Analysis and Optimization for Streaming)