HPEC 2004 Abstract Submission 5/28/2004

R-Stream 3.0: Technologies for High Level Embedded Application Mapping

Richard Lethin, Directing Engineer
Reservoir Labs, Inc.

R-Stream is a High Level Compiler being developed as part of the DARPA IPTO Polymorphous Computer Architecture Program. The compiler is targeted at the problem of mapping high performance embedded signal / knowledge processing applications, as exemplified by the Lincoln Labs Integrated Radar Tracker (IRT) reference, to polymorphous streaming hardware platforms, as exemplified by Stanford's Smart Memories and University of Texas TRIPS projects, and other commercially emerging chips which provide on-chip multiprocessing with distributed memory and explicit memory and communication through DMA operations.

Our 2.0 version, presented as a poster at HPEC last year, will be performing application mapping via the Streaming Virtual Machine interface to low level compilers (LLC). A separate abstract describing the overall software architecture in the PCA Morphware program is being submitted. This presentation will focus on the implementation and architecture of the high level compiler.

In particular, we will be able to present the performance results and insights from our 2.0 version for IRT running simulated on the reference PCA architectures. The performance results will be accompanied by details of the application transformations that 2.0 will be performing, including granularity selection, high level streaming transformations, and the manner in which we integrate the data parallel front end of IRT with the more task/thread parallel back end. Furthermore, we expect to be able to provide insight into the benefits and limitations of some aspects of the morphware software architecture, in particular the phased HLC/LLC compilation structure, the SVM interface, and the abstraction of architectures into the Streaming Machine Model (SMM) and Hierarchical Machine Model (HMM). We may further be able to accompany this with some performance results for the application mapped to commercial architectures that are emerging with similarities to the PCA architecture class.

The second major part of our presentation is to relate the transformations being performed in 2.0 to the more advanced compiler technologies being developed for the 3.0 version of R-Stream. In particular, while the 2.0 compiler will be performing streaming application transformations drawing on classic loop optimization technology, our 3.0 version will be based upon next generation program representations including Affine Partitioning and Dynamic Single Assignment. Such technologies subsume classic loop optimizations and increase their scope of applicability, albeit with substantial challenges in implementation. We expect to be able to comment on how such technologies must be adapted to the specific area of embedded computing, to the array comprehension extensions for Brook, the issues of dynamic application behavior, and to Polymorphous Hardware.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 FEB 2005**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **R-Stream 3.0: Technologies for High Level Embedded Application Mapping**		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Reservoir Labs, Inc.**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM00001742, HPEC-7 Volume 1, Proceedings of the Eighth Annual High Performance Embedded Computing (HPEC) Workshops, 28-30 September 2004 Volume 1., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

R-Stream: Compiler Technology for Next Generation HPEC

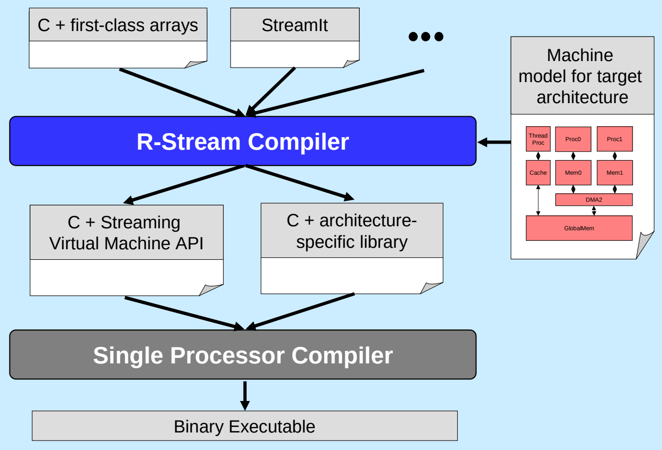
Reservoir Labs Inc.

Abstract

Agenda

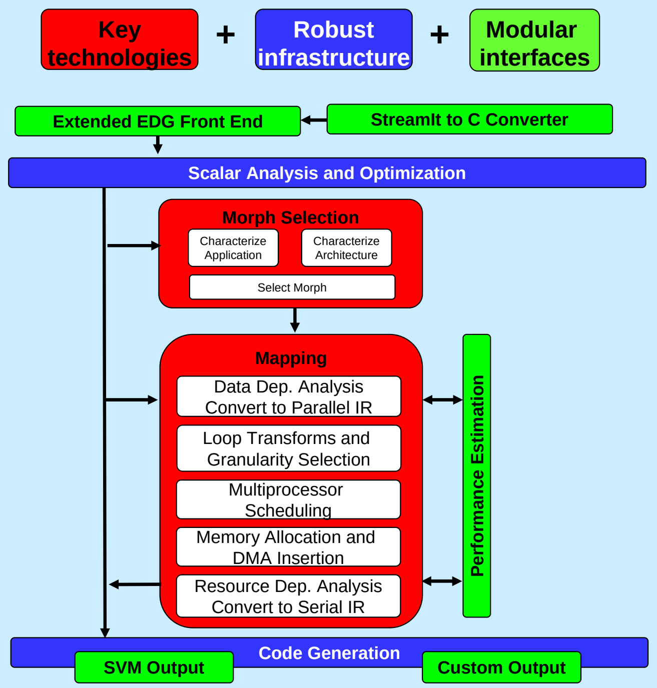
Role in Tool Chain

R-Stream is a source-to-source compiler intended to **augment** an **existing** single processor **tool chain**.



C + first-class arrays
StreamIt
•••
Machine model for target architecture
Thread Proc
Proc0
Proc1
Cache
Mem0
Mem1
DMA2
GlobalMem
R-Stream Compiler
C + Streaming Virtual Machine API
C + architecture-specific library
Single Processor Compiler
Binary Executable

Compiler Structure



Key technologies
+
Robust infrastructure
+
Modular interfaces
Extended EDG Front End
StreamIt to C Converter
Scalar Analysis and Optimization
Morph Selection
Characterize Application
Characterize Architecture
Select Morph
Mapping
Data Dep. Analysis Convert to Parallel IR
Loop Transforms and Granularity Selection
Multiprocessor Scheduling
Memory Allocation and DMA Insertion
Resource Dep. Analysis Convert to Serial IR
Performance Estimation
Code Generation
SVM Output
Custom Output

Compiler Tech. for HPEC

R-Stream compiler technology **automatically maps** applications to HPEC architectures with:

- **Multiple processor** cores
- Distributed on-chip memories w/ **DMA**
- **Reconfigurable** processors and memories

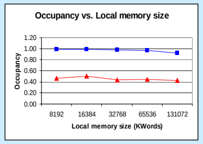
R-Stream **optimizes the whole application**, e.g. reducing memory traffic between kernels, unlike using a library alone.

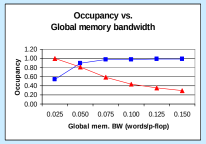
R-Stream maps one C program to multiple targets, for **faster, cheaper, more reliable development** than mapping by hand.

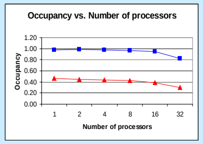
Early Results

Early results show **efficient mappings** over a wide range of architectural parameters:

	Ex. Shown	TRIPS	Smart Mem.	RAW	M-Chip (Not actual)	Imag-ine
Stream Processors	4	4	4	16	8	8
FP ALUs	8	16	2	1	8	6
Frequency	500	1000	500	420	1000	250
Gflops	16.0	64.0	4.0	6.7	64.0	12.0
Local Memory Size (words)	32768	65536	24576	8192	512 (n per proc)	64000
Global Memory BW (bytes/ns)	1.6	0.262	4	1	4	2.3
Global Memory BW (words/p-flop)	0.100	0.001	0.250	0.037	0.016	0.048





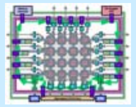


Supports Diverse Architectures

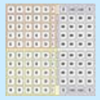
R-Stream prototype supports a **large class of architectures** via a **flexible machine model**, including:

MIT RAW

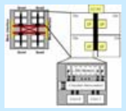
ISI / Raytheon Monarch



UT Austin TRIPS



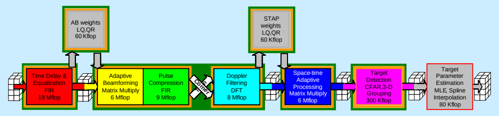
Stanford Smart Memories



Prototype 2.0 Mapper

1. Transform loops for locality, determine granularity

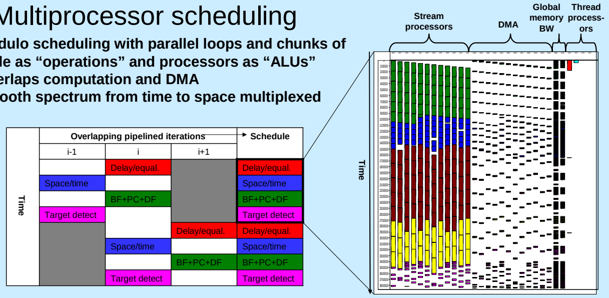
- **Goal is maximizing data that can live in local memory or local memories**
- **Interchange and partially fuse parallel outer loops**
- **Classify communications as local memory, inter-processor, or global memory**
- **Single-processor grains contain local memory communication**
- **Multi-processor grains contain communication between local memories**



AB weights LQ,QR 60 Kflop
STAP weights LQ,QR 60 Kflop
Time Delay & Equalization FIR
Adaptive Beamforming Matrix Multiply
Pulse Compression FIR
Doppler Filtering DFT
Space-time Adaptive Processing Matrix Multiply
Target Detection
Target Parameter Estimation

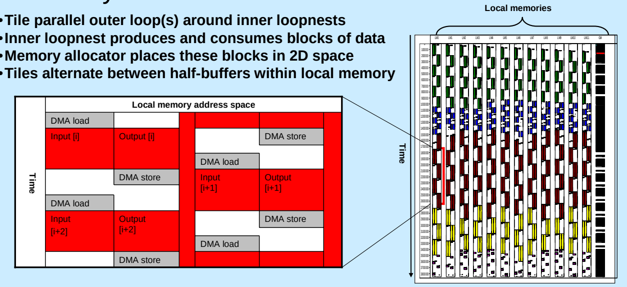
2. Multiprocessor scheduling

- **Modulo scheduling with parallel loops and chunks of code as "operations" and processors as "ALUs"**
- **Overlaps computation and DMA**
- **Smooth spectrum from time to space multiplexed**



3. Memory allocation and DMA insertion

- **Tile parallel outer loop(s) around inner loopnests**
- **Inner loopnest produces and consumes blocks of data**
- **Memory allocator places these blocks in 2D space**
- **Tiles alternate between half-buffers within local memory**



Innovative 3.0 Technology

R-Stream prototype 3.0, currently in development, will produce even **more efficient mappings** for a **wider range of applications** by leveraging:

- SRE-based internal representation to **eliminate false dependences**
- Affine partitioning framework to discover **maximum degrees of parallelism** in application
- Unified/constraint-based mapping to **avoid phase-ordering**.

Poster

Agenda

R-Stream Compiler

Peter Mattson, Richard Lethin, Eric Schweitz,
Allen Leung, Vass Litvinov, Michael Engle,
Charlie Garrett

Reservoir Labs Inc.

Scope of the problem

- Automatically parallelize C programs
  - Source-to-source compilation
  - Initially compile DSP applications for PCA architectures
  - Expand to more general embedded applications and architectures
- Deliver a reliable compiler this year (version 2.0)
  - Classic loop transformations
  - Modulo scheduling
  - Phase ordered optimizations
- Deliver a state of the art compiler next year (version 3.0)
  - Expose maximum parallelism
  - Schedule at the level of statement instances
  - Unify instruction scheduling and data placement into one mapping algorithm

reservoir labs

Demonstration

- High-Level compiler mapping Lincoln Labs' GMTI demonstration to two Polymorphous Computer Architecture chips
- Syntax extensions to C for expressing abstract arrays
- Scheduled and Mapped Code
- Streaming Virtual Machine Output

Questions you should be asking

- How automatic is R-Stream?
  - May always need programmer assistance to avoid pointers, mark reductions, etc.

- How good is the generated code?
  - Too early to say
  - Phase ordered optimizations work for some cases, but have limitations
  - Unified optimizations have theoretical claims of optimality. Will that translate into practice?

- Will R-Stream work for application X and architecture Y?
  - I don’t know, but let’s talk about it

reservoir labs