

Development directions for Elbrus performance libraries

Vadim Loginov, MCST,
10/16/2007

Email: `vlog@mcst.ru`

Authors are fully responsible for the presented material.
If you have questions please refer to Vadim Loginov (`vlog@mcst.ru`)

This presentation can be permanently accessed at MosSIGPLAN web-site <http://www.master-com.ru/sigplan/>

Agenda

- Existing libraries
- Goals, criteria and principles of performance library development
- Library structure
- Current state
- Development directions
- Problems

Existing libraries

Vendor	Mathematical library	Multimedia library
AMD	ACML - AMD Core Math Library	APL – AMD Performance Library
Apple	Velocity Engine	
Compaq	CXML - Compaq extended math library	
HP	MLIB	
IBM	ESSL - Engineering Scientific Subroutine Library	
Intel	MKL – Math Kernel Library	IPP – Integrated Performance Primitives
Sun	PerfLib – Sun Performance Library	MediaLib
MCST	EML – Elbrus Mathematical and Multimedia Library	

Goals

- Speeding-up development
- Performance improvements

Principles

Principle	Meaning
Low overhead	Time to call to IPP is mostly spent operating, not checking/copying
Low structure	Inputs are mostly data arrays, not structures
Common and basic operations	Operation applies across many applications
Atomic	Does one thing
Self-contained	Does not call OS or external libs

Design criteria

reliable	debuggable
useful	consistent
fast	complete
predictable	easy to call

A requirement from user and possibility for optimization are the main criteria to include function into Elbrus math & multimedia library

Library chapters

- Core
- Vector
- Algebra
- Signal
- Image
- Codec

Core

- Version number and status string
- Functions for memory allocation and freeing, pointer aligning

Vector

- Basic operations: arithmetic / logic / filling / copying
- Vector math functions
- Statistics
- Conversions
- Other – norms, dot product, etc.

Algebra

- Basic linear algebra subroutines of 1, 2 and 3 levels (BLAS1/2/3), containing vector-vector (BLAS1), vector-matrix (BLAS2), matrix-matrix operations (BLAS3)
- Standard packet of linear algebra LAPACK – matrix eigenvalue finding, solution to a system of linear equations, decomposition, etc.

Signal

- Fast Fourier Transform
- Signal characteristics computation, convolution
- Filtering and amplification
- Signal & noise generation
- Windowing
- Frequency and sample rate changing

Image

- Image creation, deletion, filling, copying
- Arithmetic and logic operations for images
- 2D FFT
- Conversions
- Statistics
- Auto and cross correlation
- Filtering
- Color conversions
- Geometric conversions

Multimedia(Codec) (plans)

- *I/M/DCT, De/Quantize8x8*
- *ColorConvert, Up/DownSample 420, 422, 444*
- *De/EncodeHuffman8x8*
- *DeblockingFilter*
- *SumAbsDiff, SumDiff*

Current state

Library chapter	Total number of functions	Implemented	Optimized for E90/E3M
Core	5	5	5
Vector	333	333	223/188
Algebra	272	272	0
Signal	97	97	0
Image	94	94	14
Total	801	801	242/207

Some optimization results (1)

Function Performance (clocks per element)	Vector_ Set_8U	Vector_ Copy_8U
Simple version for E90	5.5	5.5
Optimized version for E90	1.0	1.0 / 1.8
Improvement for E90	5.5	5.5 / 3.1
Simple version for E3M	0.9	1.0
Optimized version for E3M	0.09	0.17
Improvement for E3M	10	5.9

Some optimization results (2)

Function Performance (clocks per element)	Vector_ MulShift_ 16S	Vector_ Exp_32F
Simple version for E90	28.7	210.5
Optimized version for E90	17.8	51.5
Improvement for E90	1.6	4.1
Simple version for E3M	3.07	203.34
Optimized version for E3M	0.74	4.26
Improvement for E3M	4.1	47.7

Libraries versions

- linux-x86
- msvs-e3m
- solaris-e90
- linux-e90
- msvs-e90
- *msvs-e3m-stdc*
- *solaris-e90-gcc*
- *linux-e90-gcc*
- *msvs-e90-gcc*

Near term plans

- Optimization of Vector, Signal and Algebra chapters
- Interfaces adding for BLAS and LAPACK
- Checking and compiler bugs elimination

Far term plans

- Optimization for Elbrus-3S
- Optimization for V9 (MCST-4R)
- Parallel (multithreaded) library version (possibly OpenMP based)
- Functionality extension based on user requirements

Problems

- Compiler bugs
- Performance bugs
- VIS intrinsic
- Shortage of well-qualified staff