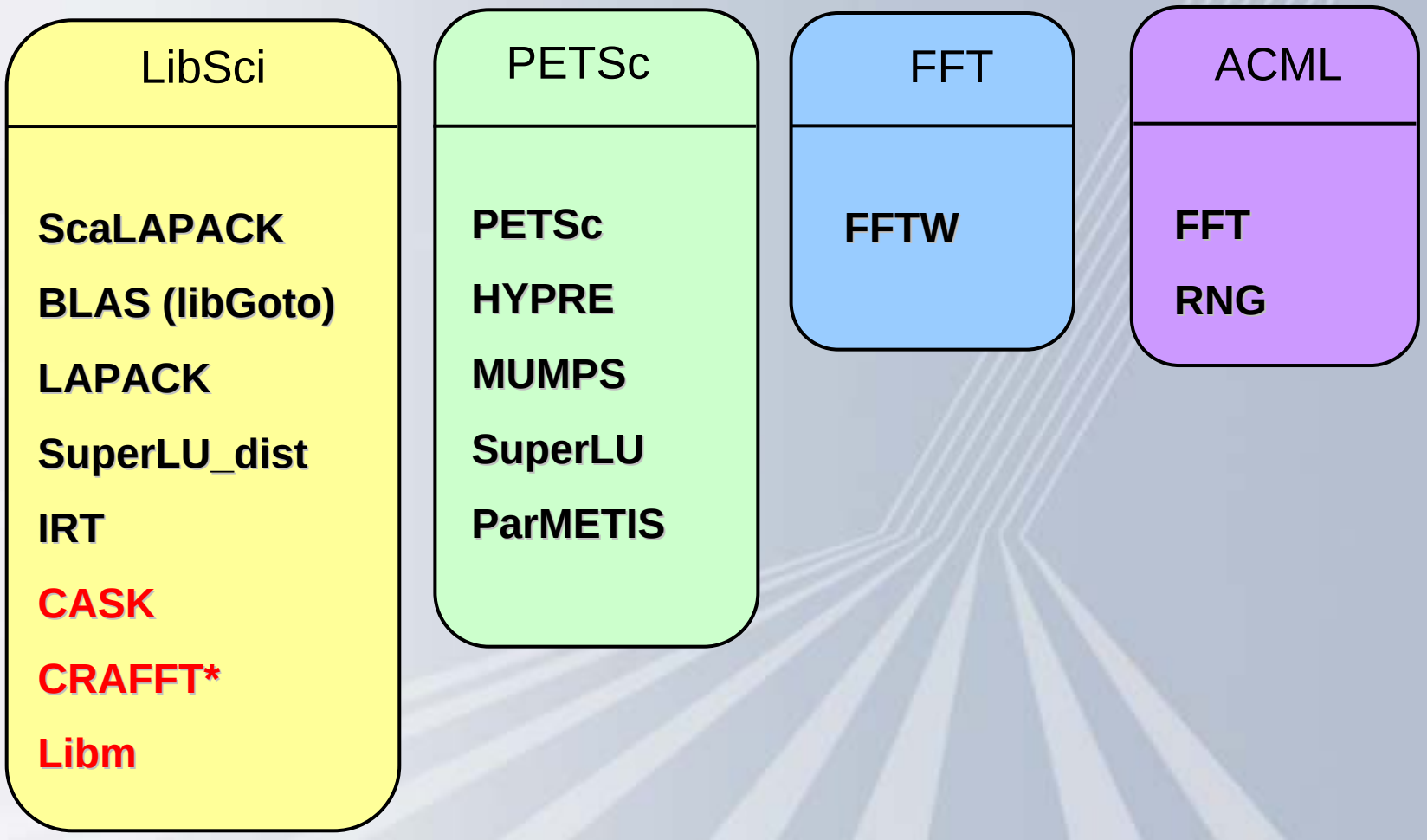
Math Libraries Tuning on the Cray XT

Luiz DeRose
Programming Environment Director
Cray Inc.
ldr@cray.com

University of Bergen
Bergen, Norway
March 11-14, 2008

Current Math Software Stack 1Q08



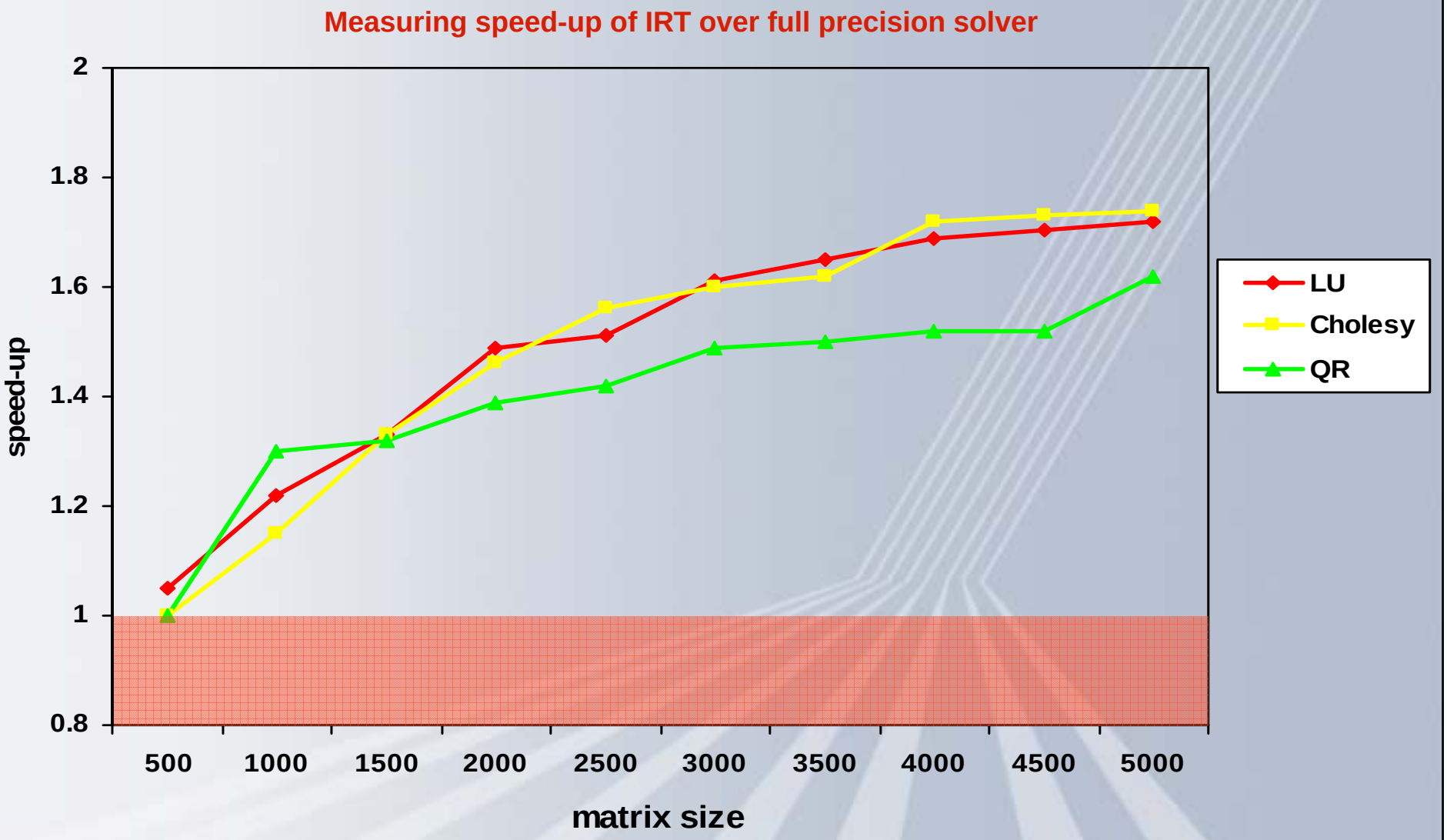
2007 Highlights

- Released LibSci 10.2.0
  - Added Goto + custom BLAS / LAPACK
    - Provided **significant performance improvements** over ACML.
  - LAPACK
  - Mixed mode ScaLAPACK support
    - MPI across sockets (1 BLACS process per socket)
    - Threaded BLAS within sockets
- libsci-10.3.0 will contain considerable performance improvements
  - CASK will improve iterative solver performance by 5-25% (problem dependent)
  - Cray Adaptive FFT
- Released PETSc 2.3.3
  - PETSc + HYPRE, SuperLU, MUMPS, ParMETIS
- Released IRT2.0 automatic interfaces
  - ‘setenv IRT_USE_SOLVERS 1’

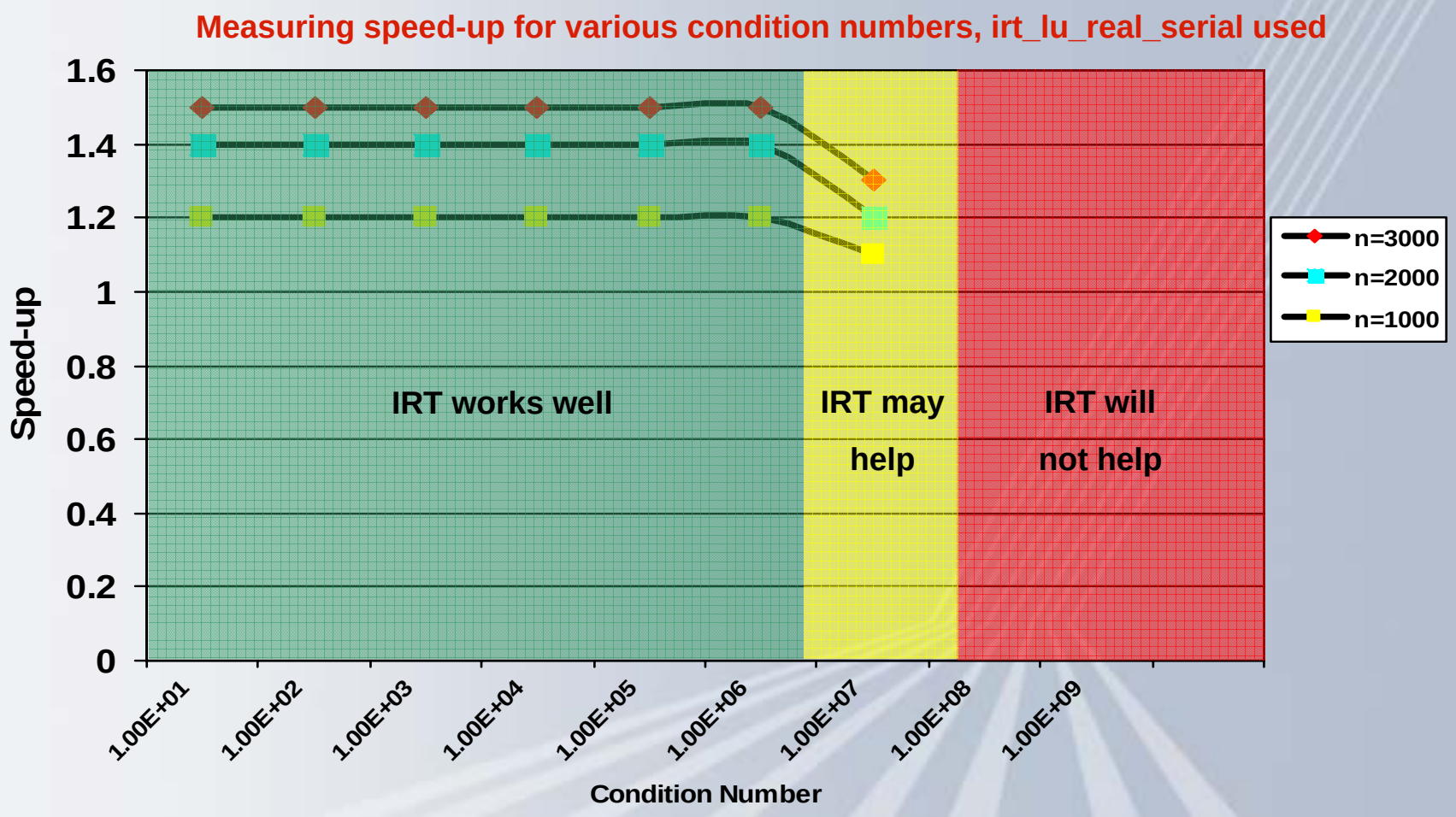
Iterative Refinement Toolkit

- Solves linear systems in single precision whilst obtaining solutions accurate to double precision
  - For well conditioned problems
- Serial and Parallel versions of LU, Cholesky, and QR
- With LibSci-10.2.0, there are now 2 ways to use the library
  1. **IRT Benchmark routines**
     - Uses IRT 'under-the-covers' without changing your code
       - Simply set an environment variable
       - Useful when you just want a quick-and-dirty factor/solve
  2. **Advanced IRT API** (from libsci-10.1.0)
     - If greater control of the iterative refinement process is required
       - Allows
         - » condition number estimation
         - » error bounds return
         - » minimization of either forward or backward error
         - » 'fall back' to full precision if the condition number is too high or IRT fails
         - » max number of iterations can be altered by users

IRT2.0 performance (serial)



IRT on XT4 (Condition vs. performance)



Math SW Emphasis in 2008

- **Auto-tuning:** use code generator and automatic tester to develop codes
  - Cray Adaptive Sparse Kernels (CASK)
- **Adaptivity:** make runtime decisions to choose best kernel/library/routine
  - Cray Adaptive FFT (CRAFFT)
  - CASK
- **Performance:**
  - Iterative Solver Performance
  - FFT performance
  - Quad-core optimization
  - Fast libm

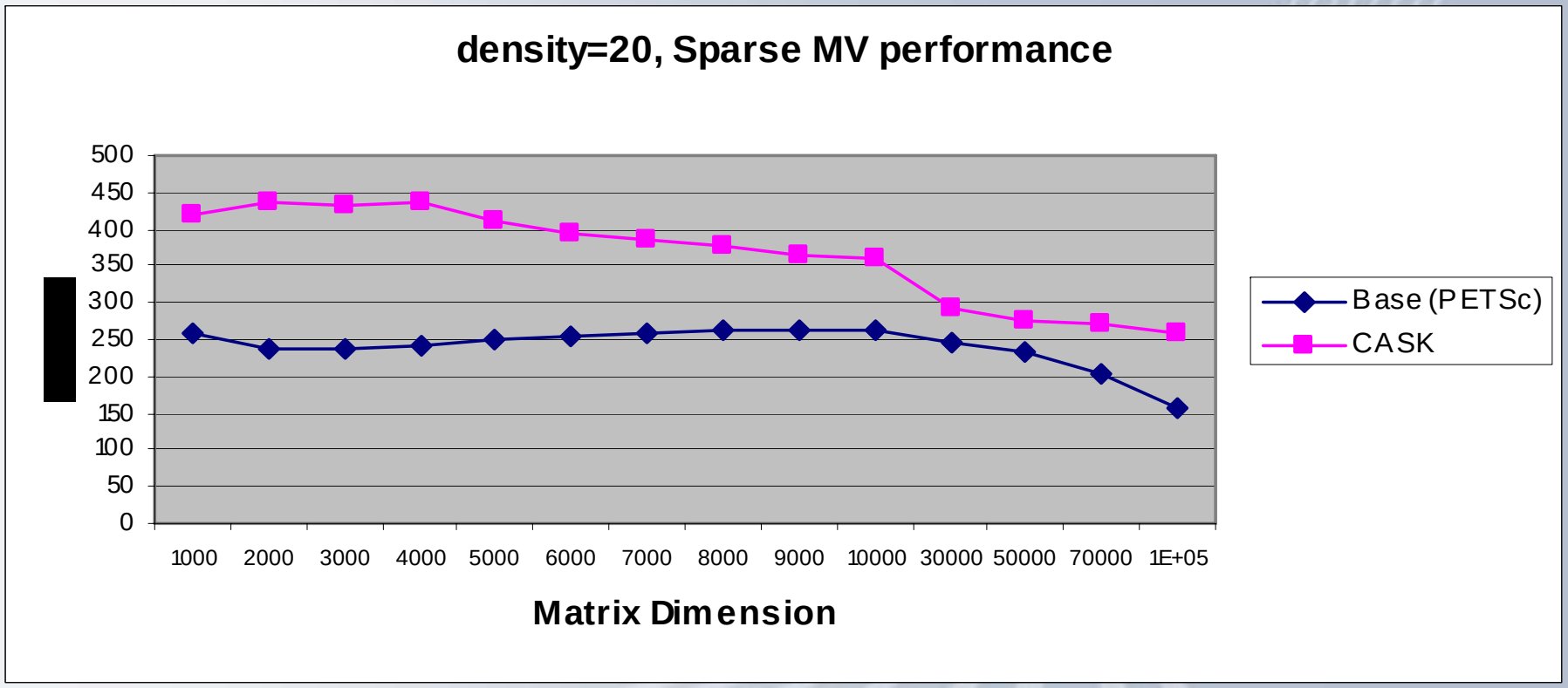
Adaptive LibSci

- Old model
  - Tuned general purpose codes
    - only good for dense
    - not problem sensitive
    - not architecture sensitive
- Adaptive Model
    - Runtime analysis allows best library/kernel to be used dynamically
    - Extensive offline testing gives allows library to make decisions or remove the need for those decisions
    - Adaptively for architectures
    - Adaptively for problem characteristics
    - Auto-tuning provides many more parameters to be studied

E.g. Iterative Solver Problem

- Most important mathematical area for scaling HPC apps
- 25-75% of time in iterative solver is in the sparse MV kernel
- Unlike dense solvers, sparse solvers do not exhibit predictable performance with respect to different matrix types
  - Performance directly relates to sparsity characteristics
  - There is no such thing as a ‘general purpose tuned kernel’
  - What we require are kernels tuned for a specific matrix
  - Any compiler optimized code will remain useful only for a specific matrix category
- It is even worse than that…
  - We can make no reasonable assumptions about the interplay of optimizations with one another

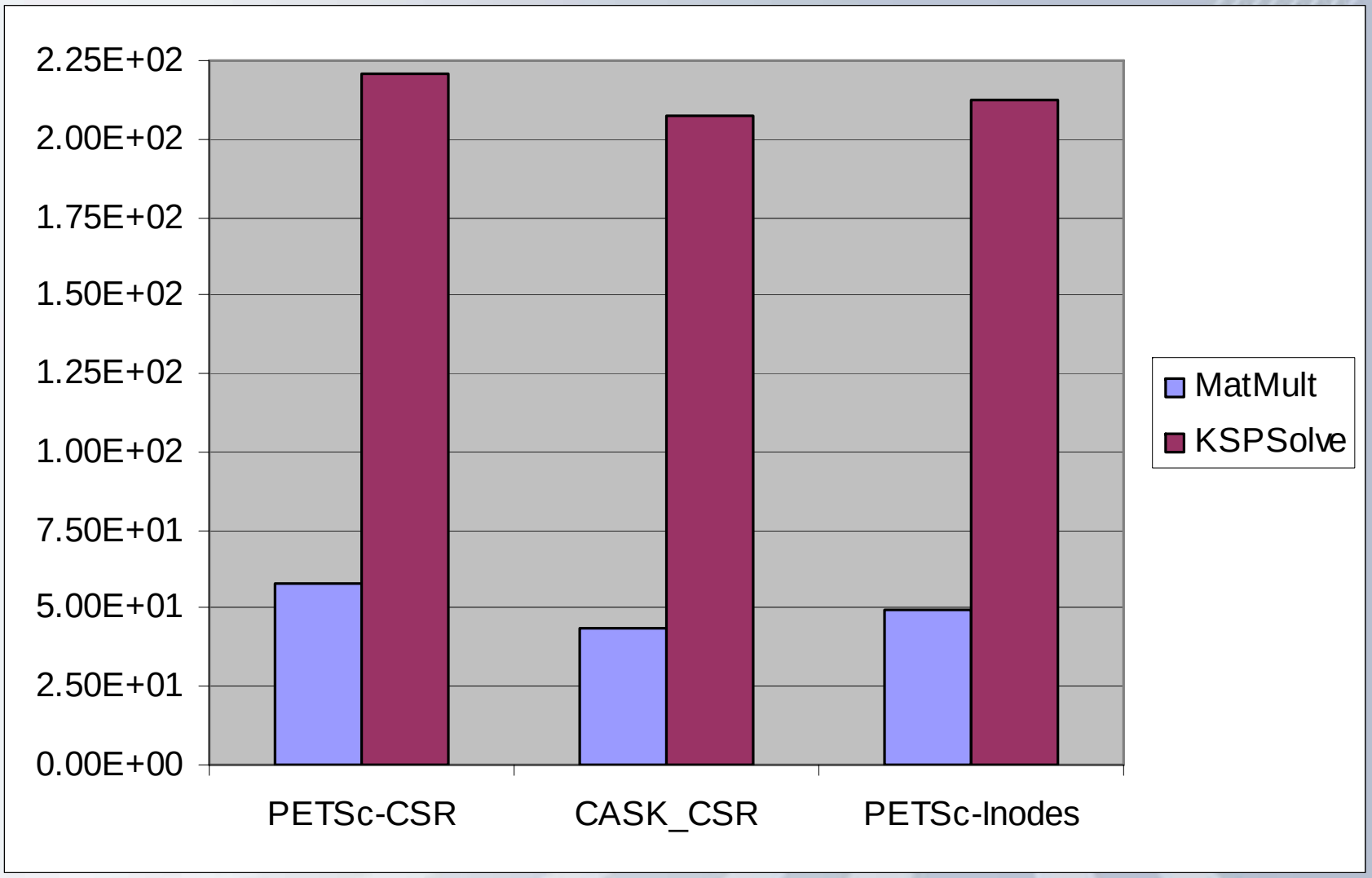
CASK Preliminary Performance



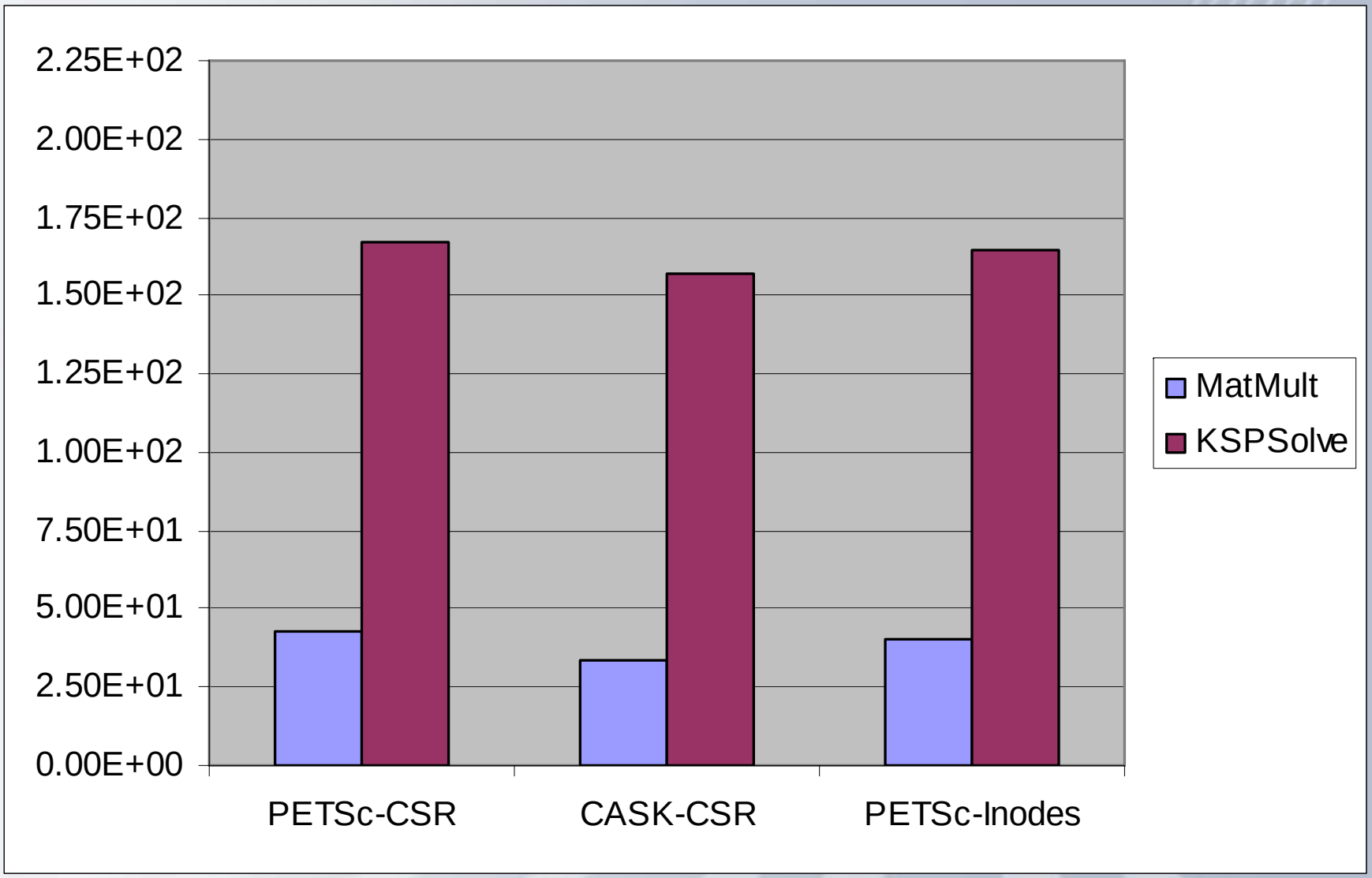
What to expect

- CASK version 1 (May 1 2008)
  - Generic CSR, single RHS
    - Real
    - Complex
  - Support for banded systems
  - Will work silently underneath PETSc
  - External API also available
- Version 2.0 (~Sept08)
  - FBSR versions
    - Compatible with PETSc inode strategy
  - Compiler optimization strategy
    - How to find a set of optimal compiler flags for a kernel
      - Widely studied discrete optimization problem

PETSc with CASK preliminary (SC)



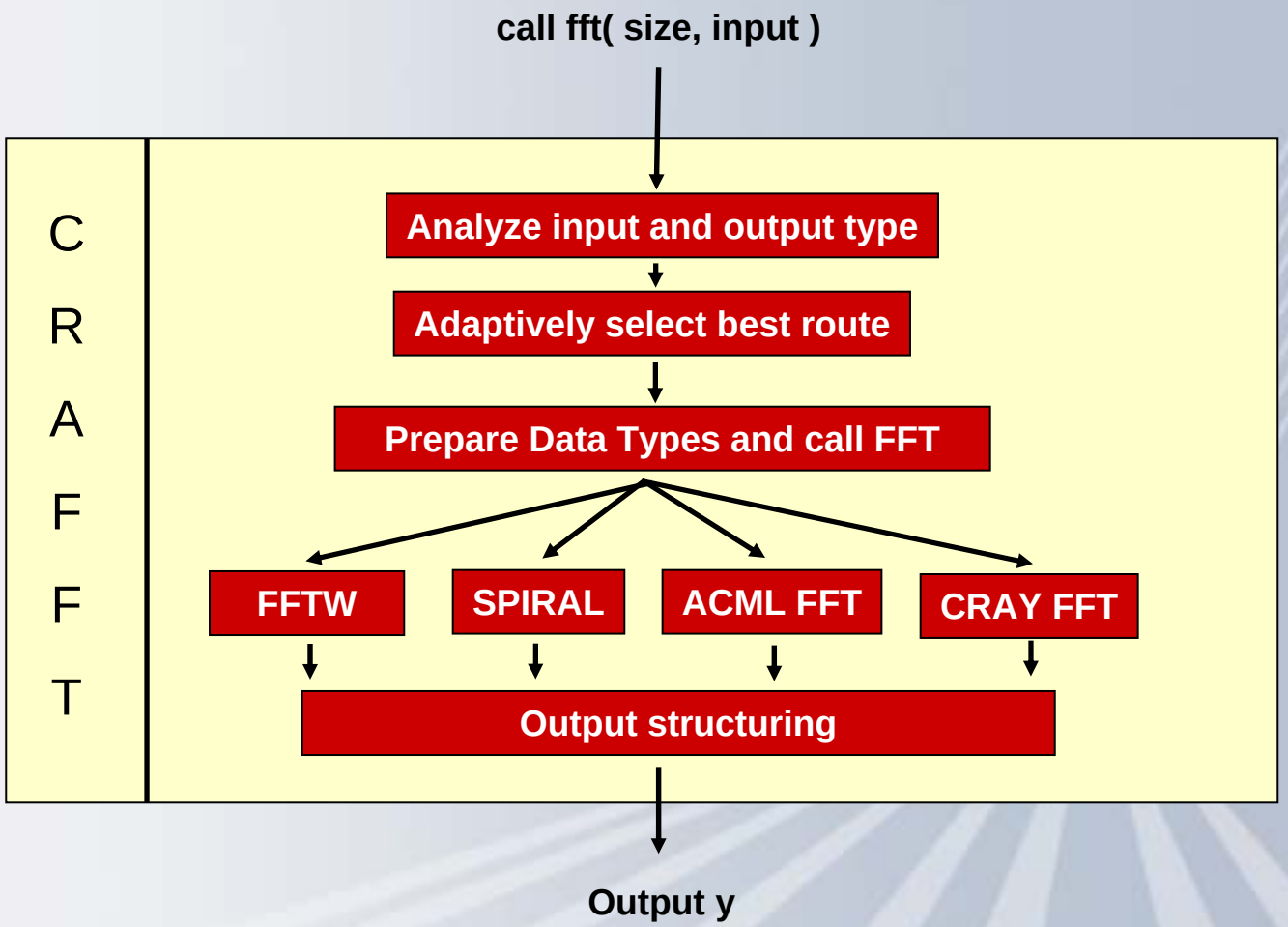
PETSc with CASK preliminary (QC)



CRAFFT overview

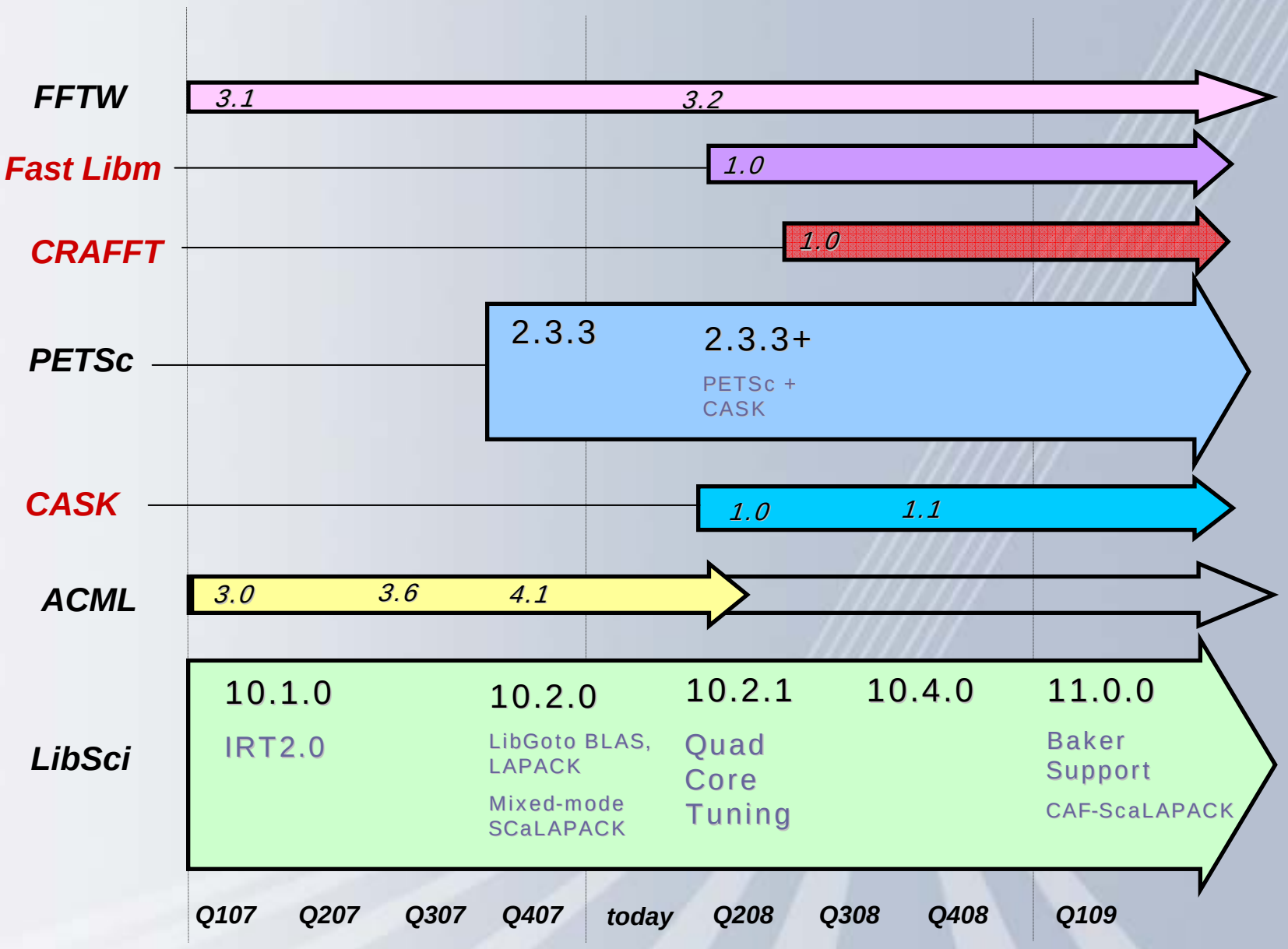
- Many different FFT options available
  - FFTW, ACML, CrayFFT….
- Some interfaces are very hard to use
  - E.g. FFTW
- CRAFFT uses very simple interface for all FFTs
- Extensive offline testing allows library to make decisions for the user
  - Which FFT library
  - No expensive plan stages
- Library adaptively selects the best FFT program to use based on online testing, can use FFT from any available library (e.g FFTW, ACML, Custom FFT)

Cray Adaptive FFT user perspective



CRAFFT* version 1.0alpha

- Largely FFTW centric
- Includes FFTW wisdom for removal of expensive plan stage
- Allows simple interface into advanced FFTW functionality
- Will also consider ACML and custom FFT choices dynamically


FFTW
3.1
3.2
Fast Libm
1.0
CRAFFT
1.0
PETSc
2.3.3
2.3.3+
PETSc +
CASK
CASK
1.0
1.1
ACML
3.0
3.6
4.1
LibSci
10.1.0
IRT2.0
10.2.0
LibGoto BLAS,
LAPACK
Mixed-mode
SCaLAPACK
10.2.1
Quad
Core
Tuning
10.4.0
11.0.0
Baker
Support
CAF-ScaLAPACK
Q107
Q207
Q307
Q407
today
Q208
Q308
Q408
Q109

Math Libraries Tuning on the Cray XT

Questions / Comments

Thank You!

University of Bergen
Bergen, Norway
March 11-14, 2008

Luiz DeRose (ldr@cray.com)

CRAY
THE SUPERCOMPUTER COMPANY