Bringing Computing Power to the People

Honors Seminar
Fall 2005
Arun Chauhan
Collaborators

Rice
Bradley Broom
Jason Eckhardt
Rob Fowler
Chuck Koelbel
John Garvin
Richard Hanson
Ken Kennedy
Cheryl McCosh
John Mellor-Crummey

Indiana
Joshua Hursey
Andrew Lumsdaine
Pooja Malpani
Youngsang Shin
Craig Shue
Roshan James
Daniel McFarlin

Ohio-State
Stan Ahalt
Ashok Krishnamurthy
P. Sadayappan

Honors Seminar, Fall 2005
Computing Power to the People
Interface
Interacting with Computers

OH NO. IT'S A TECHNICAL GLITCH THAT I DON'T KNOW HOW TO FIX.

GAA! NOW I MUST HUMBLE MYSELF TO SOME CONDESCENDING ENGINEER AND ASK FOR HELP!

AND HOW DID I TEACH YOU TO ASK?

I'M SORRY THAT I SPENT MY COLLEGE YEARS DRINKING BEER AND STUDYING ENGLISH LITERATURE.
In a Perfect World

- Talk to computers in a natural language
- Solve problems using high-level description
  - “plan this itinerary”
  - “simulate this model on these inputs”
- Seamless data access
Problem Solving with Computers in Real World

Programming Languages

Scheme

Machine

Honors Seminar, Fall 2005
Problem Solving with Computers in Real World

Programming Languages

- Scheme, Scripting Languages
- C++, Java
- C, Fortran
- Assembly Machine
The Productivity Connection

- Users love high-level languages
- High-level languages deliver inadequate performance for problem solving
- Users are forced to rewrite their applications in lower-level languages
The Performance Gap

jakes: Type–specialized FORTRAN vs MATLAB

- Sun SPARC 336MHz
- SGI Origin
- Apple PowerBook G4 667MHz

MATLAB 6.x vs FORTRAN
About MATLAB

• High-level language for numerical math
• Hugely popular among scientists and engineers worldwide
  - 1,000,000 million licenses in 100 countries
• Primarily used for prototyping!
  - inadequate performance on real-world applications
The Productivity Problem

Information Processing Technology Office

High Productivity Computing Systems (HPCS)

Mission:

- Provide a focused research and development program, creating new generations of high end programming environments, software tools, architectures, and hardware components in order to realize a new vision of high end computing, high productivity computing systems (HPCS). Address the issues of low efficiency, scalability, software tools and environments, and growing physical constraints.

- Fill the high end computing gap between today’s late 80’s based technology High Performance Computing (HPCs) and the promise of quantum computing.

- Provide economically viable high productivity computing systems for the national security and industrial user communities with the following design attributes in the latter part of this decade:
  - **Performance**: Improve the computational efficiency and performance of critical national security applications.
  - **Programmability**: Reduce cost and time of developing HPCS application solutions.
  - **Portability**: Insulate research and operational HPCS application software from system specifics.
  - **Robustness**: Deliver improved reliability to...
It’s the Compiler

We did not regard language design as a difficult problem, merely a simple prelude to the real problem: designing a compiler that could produce efficient programs.

–John Backus, the “Father of Fortran”
Example Program

function demo
  x = 1;
  y = x / 10;
  z = x * 20;
  r = y + z;
What Really Happens

```c
static void Mmcc demo (void) {
    mxArray * r = NULL;
    mxArray * z = NULL;
    mxArray * y = NULL;
    mxArray * x = NULL;
    mlfAssign(&x, mxarray0 ); /* x = 1; */
    mlfAssign(&y, mclMrddivide(mclVv(x, "x"), mxarray1 )); /* y = x / 10; */
    mlfAssign(&z, mclMtimes(mclVv(x, "x"), mxarray2 )); /* z = x * 20; */
    mlfAssign(&r, mclPlus(mclVv(y, "y"), mclVv(z, "z"))); /* r = y + z; */
    mxDestroyArray(x);
    mxDestroyArray(y);
    mxDestroyArray(z);
    mxDestroyArray(r);
    . . .
}
```

function demo
x = 1;
y = x / 10;
z = x * 20;
r = y + z;
Type Inference

- Type ≡ <τ, δ, σ, Ψ>
  - τ = intrinsic type, e.g., int, real, complex, etc.
  - δ = array dimensionality (0 for scalars)
  - σ = δ-tuple of positive integers
  - Ψ = “structure” of an array

- Examples
  - x is scalar, integer
    * type = <int, 0, ?, ?>
  - y is 3-D 10x5x20 dense array of reals,
    * type = <real, 3, <10, 5, 20>, dense>
Role of Libraries

• Libraries are key in optimizing scripting languages
  - $a = x^y \Rightarrow a = mclMTimes(x, y)$

• Libraries practically define high-level scripting languages
  - high-level operations are often “syntactic sugar”
  - a large effort in HPC is toward library development
  - domain-specific libraries make scripting languages useful and popular
Library Hierarchy

MATLAB

user-level libraries

toolboxes

C, Fortran

run-time library
Libraries as Black Boxes

- Library
- Compiler
- Library binaries
- User program
- Compiler
- Object code
Optimizing Libraries

procedure VMP (M, V, ..., s)

context

s = 1

s = ?

for i = 1, s, N
{
...
}

VMP_stridel (M, V, ...)

VMP (M, V, ...)

Honors Seminar, Fall 2005

Computing Power to the People
Architecture of the Library Compiler
The Performance Gap

jakes: Type-specialized FORTRAN vs MATLAB

- Sun SPARC 336MHz
- SGI Origin
- Apple PowerBook G4 667MHz

MATLAB 6.x
FORTRAN
What About Parallelism?

It doesn’t make good business sense for us to undertake fundamental changes in MATLAB’s architecture. There are not enough potential customers with parallel machines.

–Cleve Moler, co-founder of MathWorks, 1995
Parallel Computation

Clock Speed (MHz)

Transistors (000)

Research Issues: Immediate

- Annotation language
  - what to annotate
  - how to annotate
  - how to utilize the annotations
- Compiling for parallel systems
- Diversifying the domain
  - other languages
  - other problem domains, e.g., VLSI design
Research Issues: Longer Term

- Front-end for library development
- Time and space trade-offs in library generation
- Refining speculative optimization techniques
- Self-learning systems
- Compiling for dynamically evolving systems
Concluding Remarks

• Computing needs to be made more widely accessible

• Current problem solving systems force users to choose between performance and accessibility

• Compilers have a critical role to play and novel approaches are needed

• In modern context, the role of compilers becomes even more important to deal with parallel machines