

The History of the Argonne Advanced Computing Research Facility

Paul Messina, ANL ALCF

Jack Dongarra, ORNL, UT

Rusty Lusk, ANL MCS



Outline

- **The Inspiration – Paul Messina**
- **The Machines – Jack Dongarra**
- **The Programming Models and Outreach – Rusty Lusk**



The History of the Argonne Computing Research Facility - *the inspiration*



Paul Messina

Argonne Leadership Computing Facility

Argonne National Laboratory

Symposium on Thirty Years of Parallel Computing at Argonne

May 14, 2013

Parallelism was in the air

- **ILLIAC IV**
 - Some Argonne researchers had access to it in the late 1970s
- **Jack Schwartz and the NYU Ultracomputer project**
- **Geoffrey Fox and the Caltech Concurrent Computation Program**
- **Dave Kuck and the CEDAR project at UIUC**
- **The turning point for me: a DOE Applied Mathematical Sciences PI meeting in Germantown, MD**



MCS Division meeting c. 1983

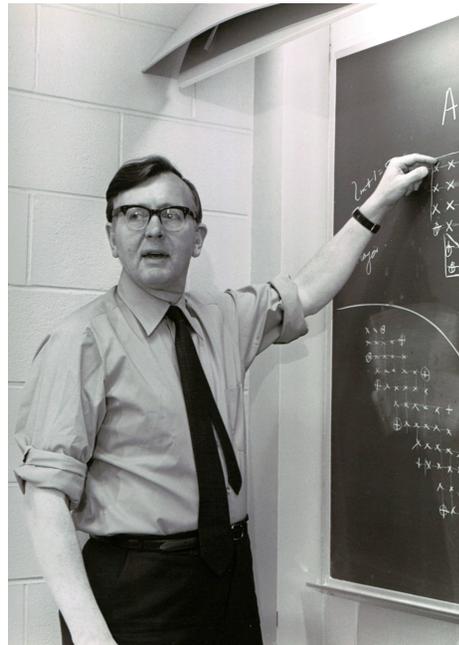
- “If our R&D is going to be relevant ten years from now, we need to shift our attention to parallel computer architectures”
- “Los Alamos has a Denelcor HEP: let’s experiment with it”



A Natural Role for Argonne's Mathematics and Computer Science Division

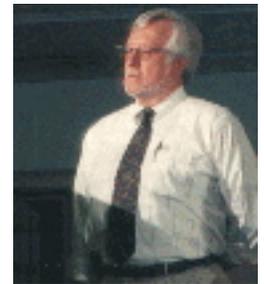
- **Our track record of algorithms implemented in carefully crafted software and made available widely**
 - Argonne numerical subroutine library
 - Automated theorem-proving systems
 - The PACKs
 - Portable software, to the extent possible

- **Jim Wilkinson and Jim Cody – and many others, some of whom are at this symposium – had charted the course**



Let's start a facility with heterogeneous architectures

- Jack Dongarra, Danny Sorensen, Rusty Lusk, Ross Overbeek – and others – suggested we establish the Experimental Computing Facility for MCS and other interested researchers to experiment with parallel computing
- Walter Massey, Argonne Lab Director, and Ken Kliewer, ALD, funded an LDRD proposal to get our own Denelcor HEP
- Soon after, we conceived a national facility that would host computers with a variety of architectures
 - No clear dominant architecture
 - Develop portable approaches
- Don Austin, then Head of DOE's AMS office, was receptive to a proposal to start a research program and establish such a facility and funded it: the ACRF was born
- The MCS staff ran with it, as Jack and Rusty will describe



The History of the Argonne Advanced Computing Research Facility - the machines

Jack Dongarra
Innovative Computer Lab
University of Tennessee



1983

- **Interest in parallel computing, both here and worldwide**
- **Remote use of Denelcor HEP at Los Alamos**
- **Formulated a research plan involving both research effort and equipment acquisition**



Advanced Computing Research Facility: 1984 -1992

- **The Advanced Computing Research Facility (ACRF) was established in recognition of the role that parallel computers would play in the future of scientific computing.**
- **Principal objectives:**
 - To encourage experimentation on computers with innovative designs
 - To assess the suitability of diverse machines for specific applications
 - To assist research in parallel computation
 - To encourage the incorporation of state-of-the-art computational techniques in research, development and industry.
 - To provide leadership in enhancing computing environments
 - To operate as a national user facility in parallel computing



Denelcor HEP - March 1984



- First experimental system installed: a Denelcor Heterogeneous Element Processor (HEP)
- Based on a MIMD architecture (First commercial)
- Shared physical address space
- Fine grain multithreading to tolerate memory latency, sync latency, and function unit latency.
- Memory full/empty bit accessed via “\$variables”, self scheduling
- Allowed multiple instructions to be “in the air” at the same time via 8 stage instruction pipeline
- Up to 50 threads could be maintained
- Suitable for both large-grain and small-grain parallel computations
- Great for multithreading
- Spurred on additional research in other MIMD systems
- Burton’s machine



1985

- **Denelcor was exciting machine, but poor cost-performance**
 - About 200 Kflop/s for Linpack(100)
- **Systems at LANL, ANL, BRL(4 PEMs), DOD (4 PEMs), and Shoko in Japan, and Messerschmitt (MBB) in Germany (3 PEMs)**
 - Argonne's Denelcor HEP decommissioned October 1985



1986

5 state of the art parallel computers replace the Denelcor HEP



- **Encore Multimax (December 85)**
 - SMP, 20 processors, 64 MB memory
 - NS 32332 w/ 32382, 120 KFLOPS / processor
 - Ken Fisher, Gordon Bell & Henry Berkhardt
 - Linpack(100) = .055 Mflop/s / processor



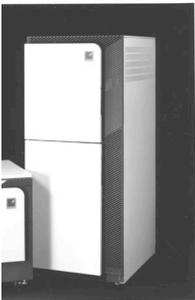
1986

5 state of the art parallel computers replace the Denelcor HEP



- **Encore Multimax (December 85)**

-
- NS 32332 w/ 32382, 120 KFLOPS / processor
- Ken Fisher, Gordon Bell & Henry Berkhardt
- Linpack(100) = .055 Mflop/s / processor



- **Sequent Balance 21000 system (Jan 86 and May 86)**

- SMP, 24 processors, 24 MB memory
- NS 32032 w/ 32082
- Casey Powell & Scott Gibson
- Linpack(100) = .059 Mflop/s / processor

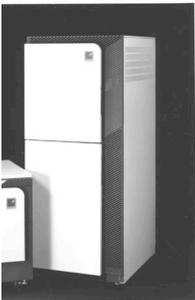


1986

5 state of the art parallel computers replace the Denelcor HEP



- **Encore Multimax (December 85)**
 - SMP, 20 processors, 64 MB memory
 - NS 32332 w/ 32382, 120 KFLOPS / processor
 - Ken Fisher, Gordon Bell & Henry Berkhardt
 - Linpack(100) = .055 Mflop/s / processor



- **Sequent Balance 21000 system (Jan 86 and May 86)**
 - SMP, 24 processors, 24 MB memory
 - NS 32032 w/ 32082
 - Casey Powell & Scott Gibson
 - Linpack(100) = .059 Mflop/s / processor

Intel iPSC/d5 (March 86)

- 32 processors, Hypercube interconnect
- 80286 & 80287 processor
- 0.5 MB memory per node
- Caltech's Cosmic cube project
- Justin's machine



1986

5 state of the art parallel computers replace the Denelcor HEP



▪ **Encore Multimax (December 85)**

- SMP, 20 processors, 64 MB memory
- NS 32332 w/ 32382, 120 KFLOPS / processor
- Ken Fisher, Gordon Bell & Henry Berkhardt
- Linpack(100) = .055 Mflop/s / processor

▪ **Sequent Balance 21000 system (Jan 86 and May 86)**

- SMP, 24 processors, 24 MB memory
- NS 32032 w/ 32082
- Casey Powell & Scott Gibson
- Linpack(100) = .059 Mflop/s / processor



Intel iPSC/d5 (March 86)

- 32 processors, Hypercube interconnect
- 80286 & 80287 processor
- 0.5 MB memory per node
- Caltech's Cosmic cube project
- Justin's machine



▪ **Alliant FX/8 (April 86)**

- SMP, 8 vector processors (MC 68000/ Weitek) -- 6 interactive processors, 32 MB memory
- 1.6 Mflop/s / processor
- Ron Gruner, Craig Mundy & Rich McAndrew
- Connections to CSRD
- Linpack(100) = 10 Mflop/s;
- Linpack(1000) = 69 Mflop/s



1986

5 state of the art parallel computers replace the Denelcor HEP



- **Encore Multimax (December 85)**
 - SMP, 20 processors, 64 MB memory
 - NS 32332 w/ 32382, 120 KFLOPS / processor
 - Ken Fisher, Gordon Bell & Henry Berkhardt
 - Linpack(100) = .055 Mflop/s / processor



- **Sequent Balance 21000 system (Jan 86 and May 86)**
 - SMP, 24 processors, 24 MB memory
 - NS 32032 w/ 32082
 - Casey Powell & Scott Gibson
 - Linpack(100) = .059 Mflop/s / processor

Intel iPSC/d5 (March 86)

- 32 processors, Hypercube interconnect
- 80286 & 80287 processor
- 0.5 MB memory per node
- Caltech's Cosmic cube project
- Justin's machine



Intel iPSC/d4 VX (October 86)

- 16 processors, Hypercube interconnect
- 1.5 MB memory per node
- Vector co-processor (2 pipelined fl pt units; add/subtract and multiply)
- Linpack(1000) = 52 Mflop/s



Alliant FX/8 (April 86)

- SMP, 8 vector processors (MC 68000/ Weitek) -- 6 interactive processors, 32 MB memory
- 1.6 Mflop/s / processor
- Ron Gruner, Craig Mundie & Rich McAndrew
- Connections to CSRD
- Linpack(100) = 10 Mflop/s;
- Linpack(1000) = 69 Mflop/s



ACRF:

Increased staff, visitors, classes, publications, etc.

1986

5 state of the art parallel computers replace the Denelcor HEP



▪ **Encore Multimax (December 85)**

- SMP, 20 processors, 64 MB memory
- NS 32332 w/ 32382, 120 KFLOPS / processor
- Ken Fisher, Gordon Bell & Henry Berkhardt
- Linpack(100) = .055 Mflop/s / processor

▪ **Sequent Balance 21000 system (Jan 86 and May 86)**

- SMP, 24 processors, 24 MB memory
- NS 32032 w/ 32082
- Casey Powell & Scott Gibson
- Linpack(100) = .059 Mflop/s / processor



Intel iPSC/d5 (March 86)

- 32 processors, Hypercube interconnect
- 80286 & 80287 processor
- 0.5 MB memory per node
- Caltech's Cosmic cube project
- Justin's machine



Intel iPSC/d4 VX (October 86)

- 16 processors, Hypercube interconnect
- 1.5 MB memory per node
- Vector co-processor (2 pipelined fl pt units; add/subtract and multiply)
- Linpack(1000) = 52 Mflop/s



▪ **Alliant FX/8 (April 86)**

- SMP, 8 vector processors (MC 68000/ Weitek) -- 6 interactive processors, 32 MB memory
- 1.6 Mflop/s / processor
- Ron Gruner, Craig Mundy & Rich McAndrew
- Connections to CSRD
- Linpack(100) = 10 Mflop/s;
- Linpack(1000) = 69 Mflop/s



ACRF:

Increased staff, visitors, classes, publications, etc.

1986

5 state of the art parallel computers rep

Denelcor HEP

First Production System
Serial #1
Manufactured for ...



- **Encore Multimax (December 85)**
 - SMP, 20 processors, 64 MB memory
 - NS 32332 w/ 32382, 120 KFLOPS / processor
 - Ken Fisher, Gordon Bell & Henry Berkhardt
 - Linpack(100) = .055 Mflop/s / processor



- **Sequent Balance 21000 system (Jan 86 and May 86)**
 - SMP, 24 processors, 24 MB memory
 - NS 32032 w/ 32082
 - Casey Powell & Scott Gibson
 - Linpack(100) = .059 Mflop/s / processor

- **Intel iPSC/d5 (March 86)**
 - 32 processors, Hypercube interconnect
 - 80286 & 80287 processor
 - 0.5 MB memory per node
 - Caltech's Cosmic cube project
 - Justin's machine

- **Alliant FX/8 (November 85)**
 - Weitek) interactive processors, 32 MB memory
 - 1.6 Mflop/s / processor
 - Ron Bruner, Craig Mundy & Fred McAndrew
 - Linpack(100) = 10 Mflop/s;
 - Linpack(1000) = 69 Mflop/s



- **Intel iPSC/d4 VX (October 86)**
 - 16 processors, Hypercube interconnect
 - 1.5 MB memory per node
 - Vector co-processor (2 pipelined fl pt units; add/subtract and multiply)
 - Linpack(1000) = 52 Mflop/s



ACRF:

Increased staff, visitors, classes, publications, etc.

1987

- **Active Memory Technology's 510/20 (Mini-DAP) (Nov 1987)**

- 1024 processors (32 x 32)
- Distributed memory
- Custom CMOS VLSI
- Mesh connected
- Bit-slice parallel processor
- Dennis Parkinson

- **Sequent Balance 21000 system (Sept 87)**

- SMP, 30 processors, 96 MB of memory
- NS 32032 w/ 32082





1988

- **Sequent Symmetry (April 88)**
 - SMP, 16 processes, 32 MB of memory
 - Intel 80386/80387 + Wietek 1167
 - 200 Kflop/s / processor





1988

- **Sequent Symmetry (April 88)**
 - SMP, 16 processes, 32 MB of memory
 - Intel 80386/80387 + Wietek 1167
 - 200 Kflop/s / processor

- **Thinking Machines Corp. CM-2 (May 1988)**
 - 16,384 processors w/ 512 Wietek
 - Hyper-cube connect
 - Bit-slice parallel processor
 - Linpack($1000_{1/2}$) = 5.4 Gflop/s
 - Danny Hillis, Sheryl Handler, & Greg Papadopoulos





1988

- **Sequent Symmetry (April 88)**

- SMP, 16 processes, 32 MB of memory
- Intel 80386/80387 + Wietek 1167
- 200 Kflop/s / processor

- **Thinking Machines Corp. CM-2 (May 1988)**

- 16,384 processors w/ 512 Wietek
- Hyper-cube connect
- Bit-slice parallel processor
- Linpack(1000_{1/2}) = 5.4 Gflop/s

- **Sequent Symmetry (April 88)**
SMP, 16 processes, 32 MB of memory

- **Cydrome Cydra 5 (July 1988)**

- Dataflow VLIW architecture
- 256 wide instruction w/ 7 fields
- Custom Fortran compiler
- Bob Rau
- Linpack(100) = 14 Mflop/s





1988

- SMP, 16 processes, 32 MB of memory
- Intel 80386/80387 + Wietek 1167
- 200 Kflop/s / processor

Thinking Machines Corp. CM-2 (May 1988)

- 16,384 processors w/ 512 Wietek
- Hyper-cube connect
- Bit-slice parallel processor
- Linpack(1000_{1/2}) = 5.4 Gflop/s
- Danny Hillis, Sheryl Handler, & Greg Papadopoulos



- Dataflow VLIW architecture
- 256 wide instruction w/ 7 fields
- Custom Fortran compiler
- Bob Rau
- Linpack(100) = 14 Mflop/s

Ardent (Dana) Graphics Supercomputer (August 1988)

- 4 processors, MIPS R3000 base w/ custom vectors
- Shared memory
- Vector registers
- Interactive graphics
- Allen Michels, Ben Wegbreit, Steve Blank
- Linpack(1000) = 47 Mflop/s





1989

- **BBN TC 2000 (Butterfly II)**

- 32 processors and 128 MB of memory
 - Non uniform memory w/SMP memory semantics
- Connected with a "butterfly" multi-stage switching network
- Motorola 88100 processors
- TotalView debugger initially developed for this system
- Randy Rettberg



ACRF Contributed to CRPC's Computational Resources

NSF S&T Center with Ken Kennedy as head

- **The CRPC has chosen to build a physically distributed, shared computing resource.**
 - Each member institution has research computing facilities with high-performance parallel computers available. The following list includes some of the high-performance computing facilities available to the CRPC:
 - BBN Butterfly GP1000, 96 nodes
 - BBN Butterfly TC 2000, 45 nodes
 - CM-2, 8K processors
 - CM-5, 1024 processors
 - CM-5, 32 nodes
 - CRAY T3D, 256 nodes
 - IBM SP1, 128 nodes
 - IBM SP1, 8 nodes
 - Intel Paragon A4, 60 nodes
 - Intel Paragon L38, 512 nodes
 - Intel Touchstone Delta, 570 nodes
 - Intel iPSC/860, 32 nodes
 - Intel iPSC/860, 64 nodes
 - Intel iPSC/860, 8 nodes
 - Kendall Square Research KSR-1
 - MasPar DEC Mpp 12000 (3)
 - nCube/2, 64 nodes (2)
 - SGI 380VGX, 8 processors
 - Sequent Symmetry, 26 processors



ACRF Machines	Architecture	Number of Processors	Features	Processor	Company in operation
Denelcor HEP	Shared Memory	1	Multithreading (64 threads/ PEM)	Custom	1982 - 1985
Encore Multimax	Shared Memory	20	Bus based	NS 32332/32382	1983 - 2002
Sequent Balance	Shared Memory	24	Bus based	NS 32032/32082	1984 - 1999
Intel iPSC/d5	Distributed Memory	32	Hypercube	Intel 80286/80287	1985 - ?
Alliant FX/8	Shared Memory	8	Vector processing	MC 68000+Weitek	1982 - 1990
Intel iPSC/d4	Distributed Memory	16	Vector processing	Intel 80286/80287+custom	
Active Memory Tech DAP	Distributed Memory	1024	Bit Sliced, Mesh connected	Custom	1986 - 2004
Sequent Balance	Shared Memory	2	Bus based	NS 32032/32082	
Cydrone	Shared Memory	1	VLIW	Custom	1984 - 1988
Thinking Machines CM-2	Distributed Memory	16,384 + 512	Bit Sliced, Fat Tree	Custom+Weitek	1982 - 1994
Ardent	Shared Memory	4	Vector processing, Graphics	MIPS R3000+custom	1985 - 1992
Sequent Symmetry	Shared Memory		Bus based	Intel 80386/80387+Weitek	
BBN TC 2000	Distributed Memory	45	Multi-staged network	MC 88100	



We have come a long way since then...

Current Argonne systems:

Mira – BG/Q

- 49,152 nodes / 786,432 cores
- 786 TB of memory
- Peak flop rate: 10 PF

Vesta - BG/Q

- 4,096 nodes / 65,536 cores
- 64 TB of memory
- Peak flop rate: 832 TF

Cetus - BG/Q

- 2,048 nodes / 32,768 cores
- 32 TB of memory
- Peak flop rate: 416 TF

Tukey – NVIDIA

- 100 nodes / 1600 x86 cores
- 200 M2070 GPUs
- 6 TB x86 memory / 1.1TB GPU memory
- Peak flop rate: 220 TF



Storage - Scratch: 28.8 PB raw capacity, 240 GB/s bw (GPFSS); Home: 1.8 PB raw capacity



The History of the Argonne Advanced Computing Research Facility Programming and Outreach

Rusty Lusk

Mathematics and Computer Science Division

Argonne National Laboratory

There were lots of interesting machines...

But - How to Program Them?

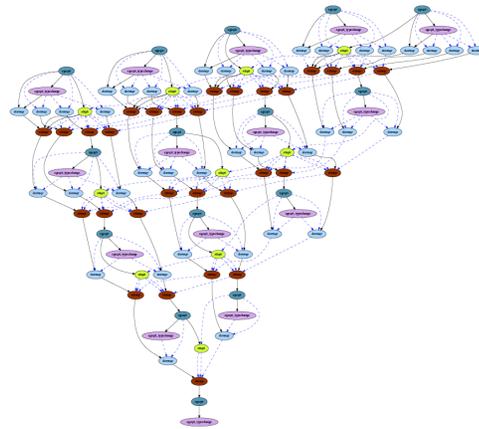
- Little Practical Experience at this point in time
- Some theoretical work
 - Per Brinch Hansen's monitors
 - Tony Hoare's CSP (communicating sequential processes)
 - Gap between theory and practice persists
- HEP Fortran
 - Elegant
 - Fine-grained parallelism!
 - Hardware support
 - BUT: $B^2 \neq B * B$
 - making portable would ruin performance



Some Early Portable Programming Models (Not Dead Yet!)

- The Jack Dongarra – Danny Sorensen approach: SCHEDULE
 - User defines dependency graph. Each unit of computation is a Fortran subroutine.
 - User required to specify each unit of computation, number of tasks below in the graph and the identity of the ones above.
 - Given the dependency information, SCHEDULE takes over and schedules the work.
 - Designed to be portable and easy to bring up on a new system.

- Current Version – QUARK runtime system
- Used in PLASMA



Fork-join
parallelism

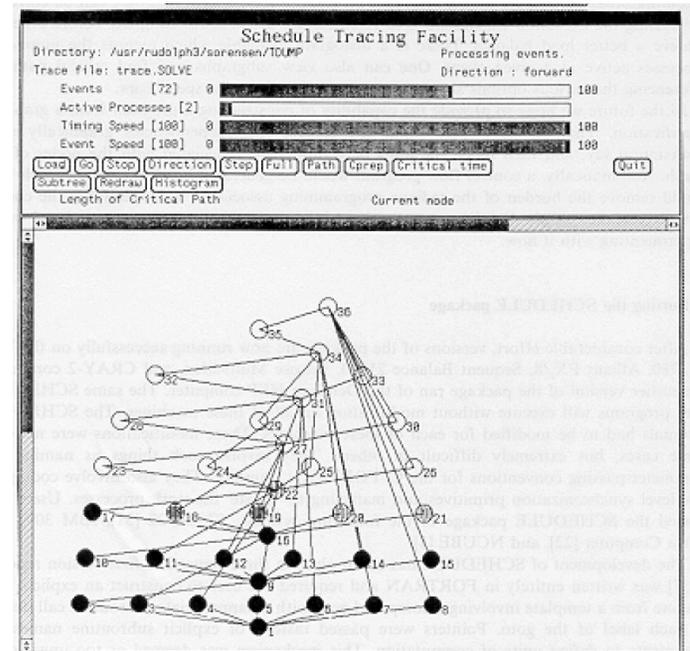


Fig. 4. Output from SCHEDULE for triangular example.

Some Early Programming Models (cont.) (Not Dead Yet!)

- The Ross Overbeek – Rusty Lusk approach: `monmacs`
 - Use vendor-specific concepts to implement locks
 - Use locks to implement monitor-building primitives (enter, exit, delay, continue)
 - Use these primitives to implement a library of useful monitors
 - Implement with (m4) macros for performance
 - Most useful was the askfor monitor: managing a shared work pool without master-slave logic
- Current version: ADLB (Asynchronous Dynamic Load Balancing) Library
 - scales to a million processes
 - used in nuclear physics application
 - put/get for shared work pool of typed tasks
 - can be used to implement multiple algorithms



Some Early Programming Models (cont.)

- The Jim Boyle – Ken Dritz approach – functional programming
- Basic idea: Use pure Lisp as programming model – single assignment variables
- A functional program specifies both parallelism and dependencies
 - $f(g(x,y),h(x,z))$ allows parallel evaluation of g and h ; f is dependent on their results.
- TAMPR transforms pure Lisp programs to C code
- C code uses Lusk/Overbeek monitors for parallelism
- C code can be transformed to use loops in place of recursion where appropriate--yields efficient implementation
- Not dead yet: single assignment variables still a key idea; parallel Prolog also explored during this period by Lusk/Overbeek
- Community still searching for High-level specification of parallelism and dependencies capable of efficient implementation



Outreach



- We got religion and wanted to spread the gospel.
- The two-week summer institutes
- The 3-day courses every six weeks



The ACRF Summer Institutes (Not Dead Yet!)

- Sponsored by NSF
- Held each September, 1987-1989
- Mornings: long lectures by distinguished speakers
- Afternoons: hands-on experience with Argonne software on ACRF machines
- Some speakers: Gordon Bell, Bill Buzbee, Josh Fisher, Dave Kuck, Neil Lincoln, Chuck Seitz, Larry Smarr, Burton Smith, Guy Steele, Don Austin, Mani Chandy, Arvind, Tom DeFanti, David Gelernter, John Gurd, Ken Kennedy, Alex Nicholau, others
- Admission by application and review, about 20 students each summer
- Next Summer Institute: July 29 – August 9, 2013



NSF 1987
Summer Institute of Parallel Computing



Summer Institute in Parallel Processing
September 5-15, 1989
Argonne National Laboratory



The High-Volume Approach: Regular Courses

- Every six weeks, Argonne held a 3-day course in parallel computing free and open to anyone.
- Gave people experience on ACRF machines
- Conveyed Argonne portable programming models and ideas
- Introduced a whole generation to parallel computing



Summary

- The ACRF kicked off Argonne's activities in parallel computing
- It was an exciting time (still is!)
- The key to success was the *combination* of
 - a math/computer science research environment
 - acquisition of machines of many different architectures
 - invention and implementation of software for programming them, focusing on both portability and performance
 - outreach to expand the parallel computing community



The End

