

# Benchmarking a Network of PCs Running Parallel Applications

Jeffrey K. Hollingsworth  
Erol Guven  
Cuneyt Akinlar

University of Maryland - College Park



© Copyright 1998, J.K. Hollingsworth, All Rights Reserved.

# Motivation

- **Commodity Hardware:**
  - is much cheaper than custom parallel hardware.
  - OS's are more refined due to economy of scale.
- **Questions**
  - How well do PCs run parallel programs?
  - What PC operating systems are appropriate?
  - How important are lightweight protocol stacks?
- **Methodology:**
  - Use micro-benchmarks for communication.
  - Run un-modified parallel applications:
    - to evaluate overall performance.
    - to eliminate benefits of “heroic” porting efforts.

# Test Environment

- Hardware

- 120Mhz Pentium PCs with 64 MB of memory
- 100Mbps Ethernet (non-switched)

- Operating Systems

- Windows NT 4.0 with TCP/IP
- Linux 2.0.1 with TCP/IP
- Linux 2.0.1 with U-Net and Active Messages
  - U-NET: user mode, low latency/high bandwidth
  - Active Messages: polling based message handlers

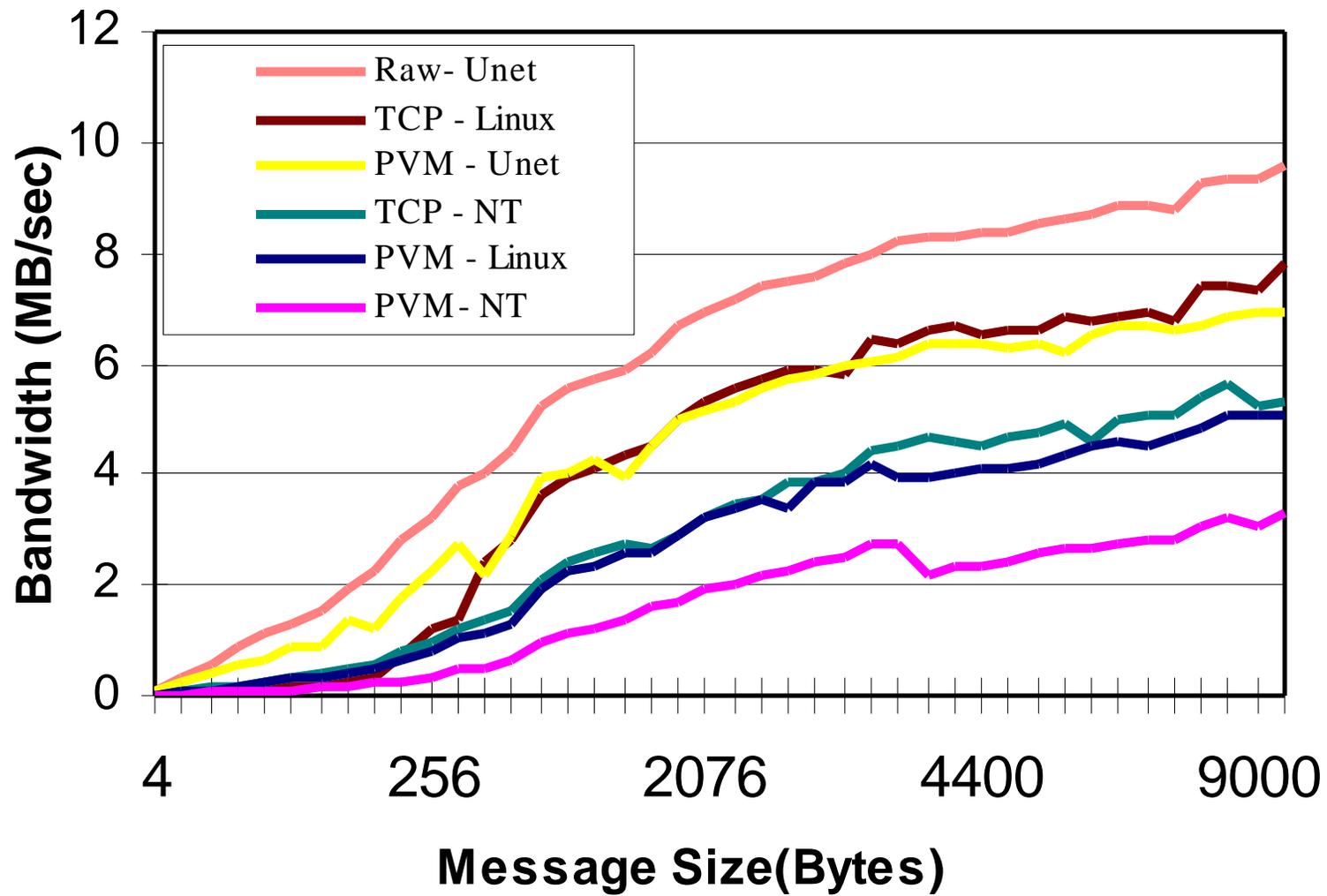
- Parallel Environment

- PVM - Message Passing

# Micro-benchmarks

- Designed to measure maximum performance
- Test Program
  - simple ping-pong program send and echo messages
  - vary size of messages sent
  - measure 1/2 roundtrip time
  - compute average of 3,000 runs to reduce timing glitches.
- Tests
  - use native message passing for three environments
  - measure for `pvm_send` and `pvm_recv`

# Micro-benchmark Results



# Sources of differences in TCP Performance

## Use Pentium hardware monitor

measured: caches, v-pipe, branch prediction

## Windows NT

executes 32% more instructions for the same task  
 averages 1.6 extra cycles per instruction

	Linux				Windows NT				Extra CPI	Percent Extra CPI
	Ops (Million)	Pct. Instr.	Cycles (Million)	Avg. CPI	Ops (Million)	Pct. Instr.	Cycles Millions	Avg. CPI		
<b>Instructions executed</b>	165.0		989.1	6.0	217.5		1,648.6	7.6	1.6	
<b>Data read Misses</b>	6.0	3.7	283.2	1.7	12.3	5.7	502.5	2.3	0.6	37.5
<b>Code cache miss</b>	12.5	7.6	162.1	1.0	20.5	9.5	267.0	1.2	0.2	15.5
<b>Misaligned data refs</b>	8.1	5.0	24.4	0.2	23.4	10.7	70.1	0.3	0.2	11.0
<b>V-Pipe Utilization</b>	38.8	23.6	(233.1)	(1.4)	45.0	20.7	(269.9)	(1.2)	0.2	11.0
<b>Un-predicted Branches</b>	6.0	3.6	29.9	0.2	13.0	6.0	64.9	0.3	0.1	7.4
<b>Code TLB miss</b>	0.4	0.2	15.1	0.1	0.9	0.4	35.8	0.2	0.1	4.6
<b>Data TLB miss</b>	0.0	0.0	1.7	0.0	1.7	0.8	14.8	0.1	0.1	3.6
<b>Other</b>									0.2	9.3

# Application Measurement Study

- Environment

- 120Mhz Pentium PCs with 64 MB of memory
- 100Mbps Ethernet segment
- Operating Systems
  - Windows NT 4.0 (Service Pack 2)
  - Linux 2.0.1

- Applications

- NAS Parallel Benchmarks
- Run with the largest class that fit into memory

- Communication (PVM)

- TCP/IP
- U-Net

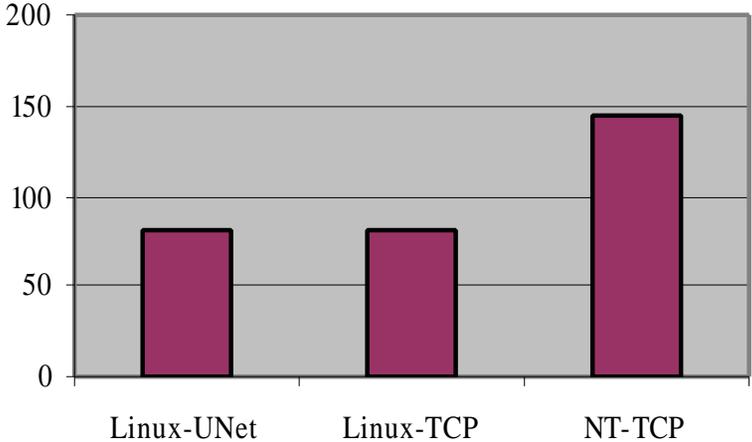
# Message Traffic for NAS Applications

- Times relative to PVM/U-net
- Message Traffic
  - Application Data Only (no protocol overhead)
  - Is total volume for four process

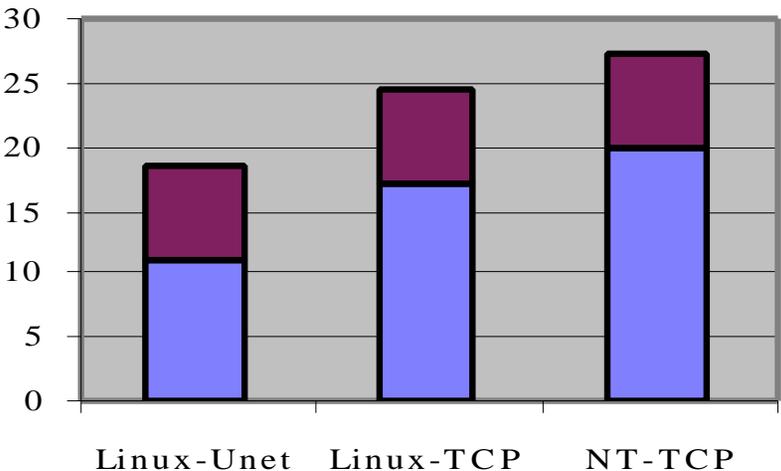
<b>Program</b>	<b>MBytes</b>	<b>Msgs</b>	<b>Msg Size</b>	<b>MBytes/ sec</b>	<b>Msgs/ sec</b>
<b>MG</b>	1.5	181	8,466	0.1	16.6
<b>IS</b>	33.5	41	818,491	2.6	3.1
<b>BT</b>	88.7	52,923	1,676	2.6	1,579.8
<b>SP</b>	30.9	22,703	1,363	2.9	2,102.1
<b>FT</b>	11.0	42	262,152	3.4	13.1
<b>LU</b>	0.2	5,105	48	0.09	191.2

# NAS Results

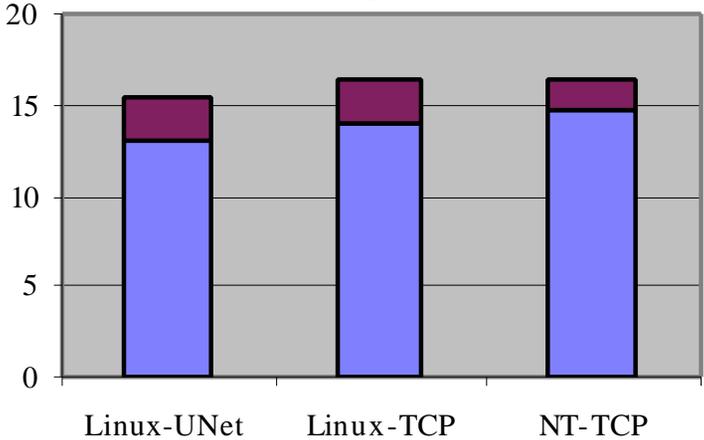
EP



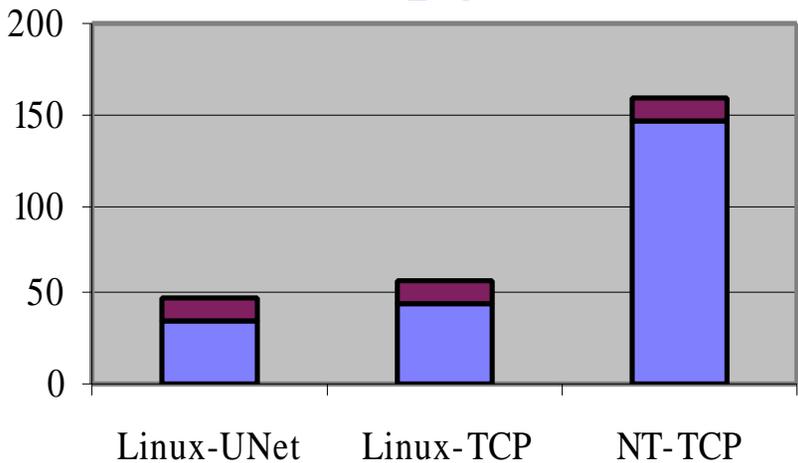
MG



IS



BT

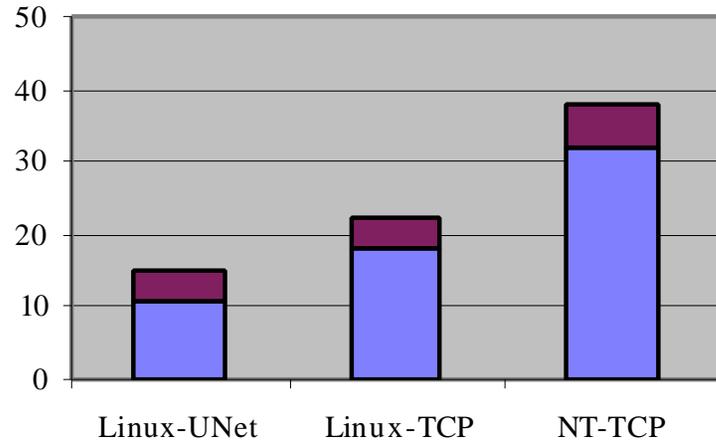


 Communication

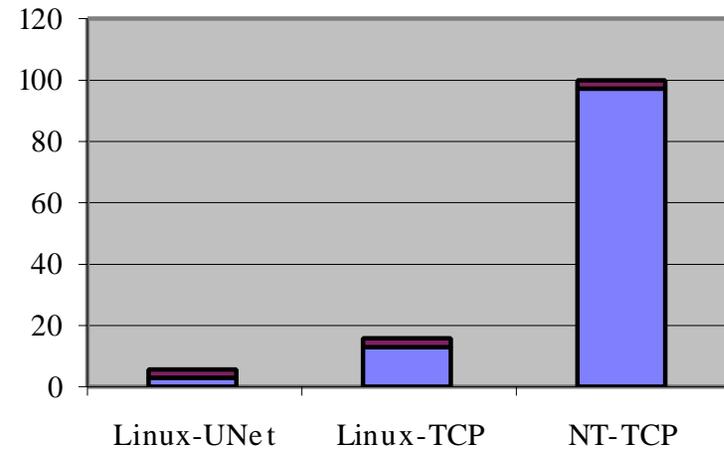
 Computation

# NAS Results (Cont.)

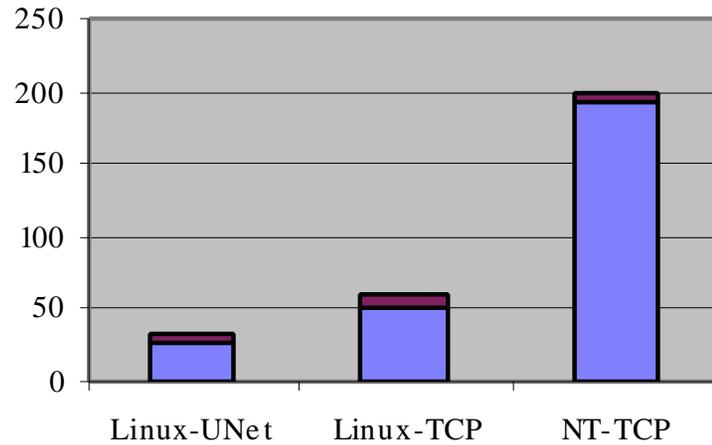
FT



LU



SP

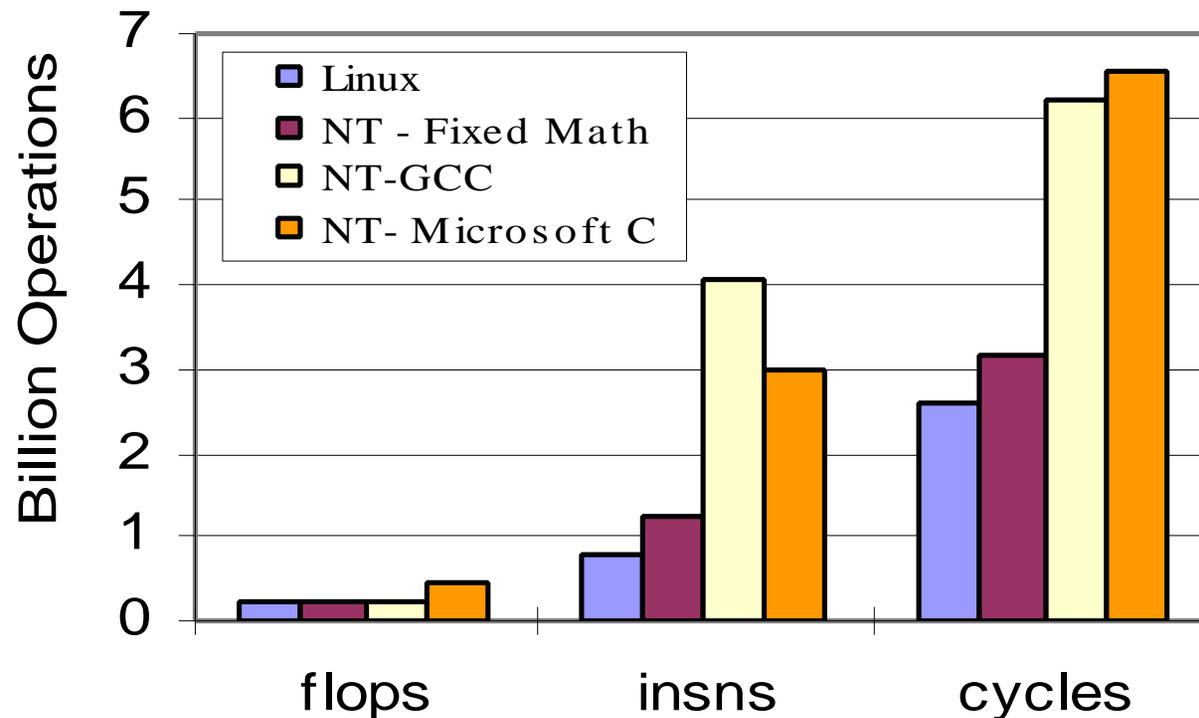


 Communication

 Computation

# Comparison of Math Libraries

- EP ran slower on NT, but its just computation
- Possible Causes: Compiler, runtime library
  - extracted computation from EP to isolate problem
- NT Math library used software versions of
  - floor, ceil, fabs, and log10



# Conclusions

- NT is not ready for high performance apps.
  - TCP performance is at least 60% worse than Linux.
  - Supplied math libraries don't use hardware features.
- TCP and PVM add substantial overhead.
  - TCP up to 200% slower than U-Net for applications.
  - PVM adds at least 40% over raw communication layer.
- Hardware is not the main limit.
  - Communication performance of OS is the problem.
  - True for micro-benchmarks and applications.