



Software Environment

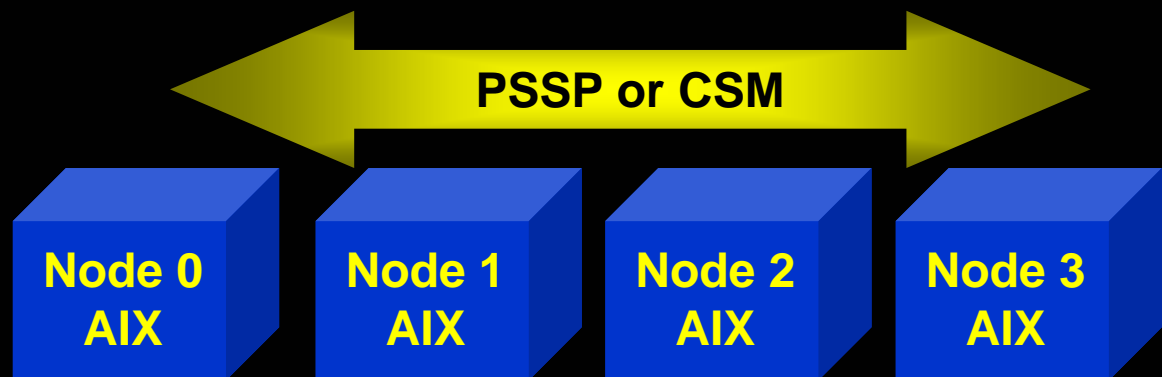
IBM
July, 2006

Agenda

- **AIX**
- **Performance Monitors**
- **Development tools**
- **Debuggers**

Operating System Software

- **Server OS**
 - **AIX (Advanced Interactive eXecutive)**
 - AT&T System V
 - Journaled
- **Cluster System Management (CSM)**
 - Management of distributed or clustered servers
- **Parallel System Support Program (PSSP)**
 - Predecessor to CSM
- **Parallel Environment (PE)**



HPC Software Stack

- **Batch queuing:**
 - LoadLeveler
- **Parallel file system:**
 - General Parallel File System (GPFS)
- **MPI tools and library:**
 - Parallel Environment (PE)
- **Math library:**
 - Engineering and Scientific Software Library (ESSL)

AIX Operating System

- **AIX 5L**
 - **Current (new) generation of IBM's Unix**
 - Linux “affinity”
 - Combines Unix technologies of AIX and Linux
- **Current Versions:**
 - AIX 5L version 5.2
 - AIX 5L version 5.3

Getting to know your system

- **uname -a**
- **oslevel -f -r**
- **df**
 - **lspv ...**
 - **lslv ...**
 - **lsvg ...**
- **ifconfig -a; netstat -i**
- **lsdev -C | grep proc | wc -l**
- **lsattr -E -l proc0 -a type**
- **/usr/bin/pmcycles**
 - **Install bos.pmapi**
- **lsattr -E -l mem0 -a size**
- **prtconf**

Parallel Environment (PE)

- **Develop, debug, analyze, tune, and execute parallel applications**
 - **Parallel Operating Environment (POE)**
 - **MPI**
 - **Optimized for IBM switches and nodes**
 - **pdbx (Parallel Debugger)**
 - **Attach to running process**
 - **Parallel utilities, for easing file manipulation**

PE Example

```
*****
```

```
c* Hello World Fortran Example
```

```
c To compile: mpixlf_r -o hello_world_f hello_world.f f
```

```
c*****
```

```
c
```

```
    program hello
```

```
    implicit none
```

```
    write(6,*) 'Hello, World!'
```

```
    stop
```

```
    end
```

PE Example

Host.list:

Node1

Node1

Node2

Node2

```
$ xlc -o hello_world hello_world.c
```

```
$ export MP_HOSTFILE=$PWD/host.list
```

```
$ poe hello_world -procs 4
```

Hello, World!

Hello, World!

Hello, World!

Hello, World!

AIX Characteristics

- **Journalized file system**
 - JFS and JFS2
 - File coherency
- **Other AIX'isms and terms**
 - LPP - Licensed Program Product (/usr/lpp/...)
 - BOS - Base Operating System (bos.rte, bos.up, etc)
 - Administration:
 - PTF- A specific software patch
 - APAR - A software fault or enhancement description
 - EFIX -An emergency software fix, invalidated
- **Note: New process for delivering fixes**
 - Sets of fully tested combinations of updates

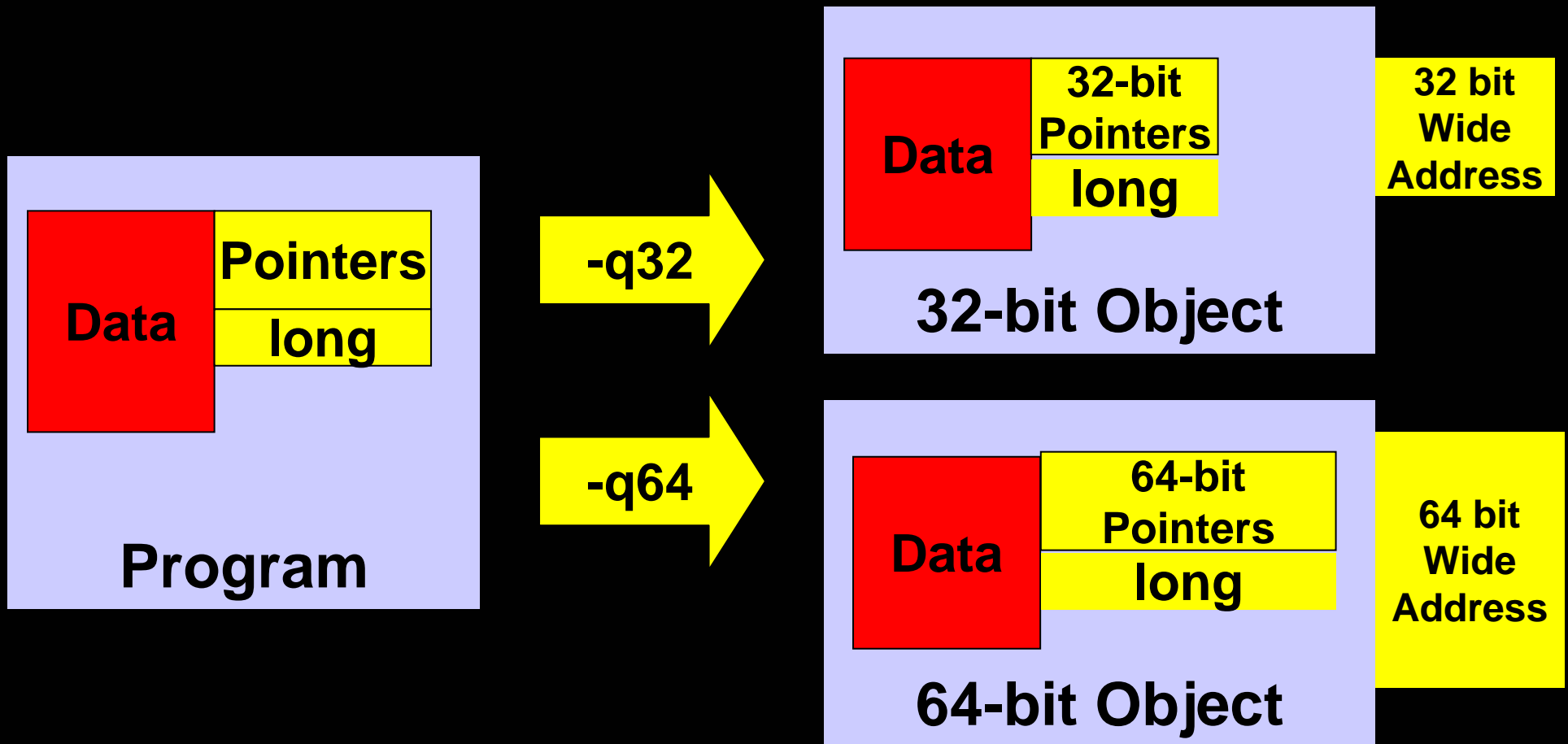
Linux Affinity

- **AIX bias toward “Linux – like” environment**
- **Emerging Linux applications**
- **GNU tools**
- **GNU utilities**
- **Linux look & feel**

64-bit Operating System

- **Operating system address modes:**
 - **32-bit kernel**
 - Limitation: 96 Gbyte memory
 - Built -qarch=com
 - **64-bit kernel**
 - No limitations
 - Built -qarch=ppc
- **Hardware:**
 - 64-bit design
- **Software:**
 - 32-bit model (default)
 - 64-bit model

Application Address Mode



Application address mode is independent of operating system address mode

Even more on 64-bit... (because it is so often confused)

- 64-bit floating point representation is higher precision
 - Fortran: REAL*8, DOUBLE PRECISION
 - C/C++: double
 - You can use 64-bit floating point with `-q32` or `-q64`
- 64-bit addressing is totally different. It refers to how many bits are used to store memory addresses and ultimately how much memory one can access.
 - Compile and link with `-q64`
 - Use `file a.out myobj.o` to query addressing mode
- The AIX kernel can be either a build that uses 32-bit addressing for kernel operations or uses 64-bit addressing, but that does not affect an application's addressibility.
 - `ls -l /unix` to find out which kernel is used
 - Certain system limits depend on kernel chosen

One more thing about 64-bit...

- **If you use `-q64`:**
 - You can use lots of memory
 - `INTEGER*8` or long long operations are faster
- **If you use `-q32`:**
 - You may run a few percent faster
 - Fewer bytes are used storing and moving pointers
 - You will have to learn about `-bmaxdata`
 - AIX link option
 - `-bmaxdata:0x10000000` = 256 Mbyte = default
 - `-bmaxdata:0x80000000` = 2 Gbyte
 - `-bmaxdata:0xC0000000` = not widely publicized trick to use more than 2 Gbyte with `-q32`
 - “C” is the maximum
 - `-q64`
 - `-bmaxdata:0` = default = unlimited
 - Other `-bmaxdata` values will be enforced if set

Essential Commands

- **Software interrogation**
- **Hardware interrogation**
- **System management**

Hardware Configuration

lscfg: Installed resource list

```
/home/myuid$ lscfg  
INSTALLED RESOURCE LIST
```

The following resources are installed on the machine.
+/- = Added or deleted from Resource List.
* = Diagnostic support not available.

Model Architecture: chrp

Model Implementation: Multiple Processor, PCI bus

+ sysplanar0	00-00	System Planar
+ mem0	00-00	Memory
+ proc0	00-00	Processor
+ L2cache0	00-00	L2 Cache

Software Configuration

lslpp: Installed Software

```
/home/myuid$ lslpp -L
```

Fileset	Level	State	Description

Adobe.acrobat	3.0.1.0	C	Adobe acrobat reader
DB2V5CAE.Bnd	5.2.0.0	C	DB2 Client Application
DB2V5SERV.Bnd	5.2.0.0	C	DB2 Server(s) Software
IBMVJava.dab.adt	2.0.0.0	C	VisualAge for Java
IBMVJava.dab.rte	2.0.0.0	C	VisualAge for Java

Configuration Report

- **prtconf**
 - **Print Configuration**
 - **Standard Unix command**
- **Information**
 - **Processors**
 - **Memory**
 - **Operating system**

prtconf

```
$ prtconf -ckLms  
CPU Type: 64-bit  
Kernel Type: 32-bit  
LPAR Info: 1 NULL  
Memory Size: 131072 MB  
Processor Clock Speed: 1900 MHz
```

System Management Installation Tool (SMIT or SMITTY)

Move cursor to desired item and press Enter.

Software Installation and Maintenance

Software License Management

Devices

System Storage Management (Physical & Logical
Storage)

...

Problem Determination

Performance & Resource Scheduling

..

Applications

F1=Help

F2=Refresh

F3=Cancel

F8=Image

F9=Shell

F10=Exit

Enter=Do

Performance Monitors

- **System (node) performance**
 - **topas**
 - Similar to Linux “top”
 - Root user has to invoke first time to create a file in /etc
 - **nmon**
 - Freeware from IBM UK
 - **vmstat**
 - virtual memory statistics

Topas

```
magnet                averages:  7.90,  7.90,  7.38
Cpu states:           ...system,  0.0% wait,  50.0% idle
Real memory:          ...procs  512.0M files 2544.0M total
Virtual memory:       ... used   150.2M total
```

PID	USER	...STAT	TIME	CPU%	COMMAND
30972	myusrid	... run	0:22	50.0%	lu.W
516	root	... run	591:16	49.5%	Kernel (wait)
23950	myusrid	... run	0:00	0.4%	monitor4.1
774	root	... run	568:46	0.0%	Kernel

nmon

- **Performance tuning utility**
- **Freeware, AIX and Linux**
- **Performance data:**
 - **CPU utilization**
 - **Memory use**
 - **Kernel statistics and run queue information**
 - **Disks I/O rates, transfers, and read/write ratios**
 - **Free space on file systems**
 - **Disk adapters**
 - **Network I/O rates, transfers, and read/write ratios**
 - **Paging space and paging rates**
 - **Etc.**

Virtual Memory STATistics: vmstat

- System (node) resources
 - Memory
 - Page faults
 - CPU

```

$ vmstat 1
kthr      memory          faults        cpu
-----  -
 r   b   avm    fre    ...    us  sy  id  wa
  0   0 70893 364069  ...    3   2 92   2
24   1 70894 364068  ...   35  65  0   0
25   2 70895 364067  ...   37  63  0   0

```

Application Performance Monitors

- **User (application) performance**
 - xprofiler
 - tprof
 - Very much improved in AIX 5.3
- **Performance Monitor Tools**
 - hpmcount
 - hpmstat
 - pmcycles
 - pmlist

Xprofiler

- **Compile, link, execute:**
 - `$ xlf -g -pg ... # "-g" enables histogramming`
 - `$ a.out # →gmon.out file is produced`
- **Analyze:**
 - `$ xprofiler a.out gmon.out # GUI interface`
 - `$ gprof > gprof.report ! text report # Text`
- **Time sample interval: 0.01 sec.**
- **MPI codes generate "N" gmon.out files**

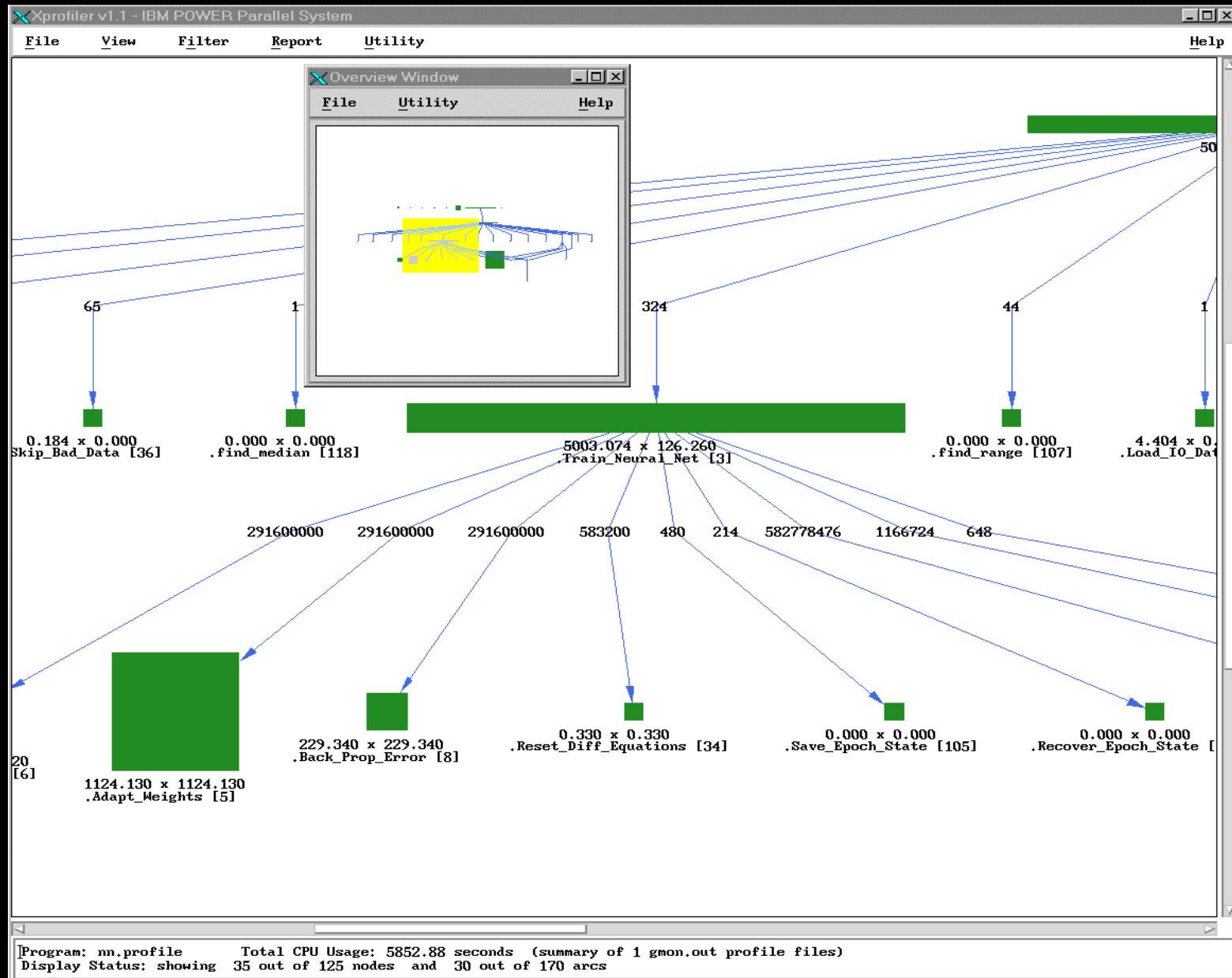
Xprofiler Tips

- **Text mode:**
 - use gprof
 - Libraries must match across systems.
- **Flat profile:**
 - Select "Report"...
- **Drill:**
 - Right click on routine
 - Select "Code Display" for a source-code view
 - Use "File ; Set File Search Paths" to set source directory
- **Graphical Display:**
 - Width of a bar: time including called routines
 - Height of a bar: time excluding called routines

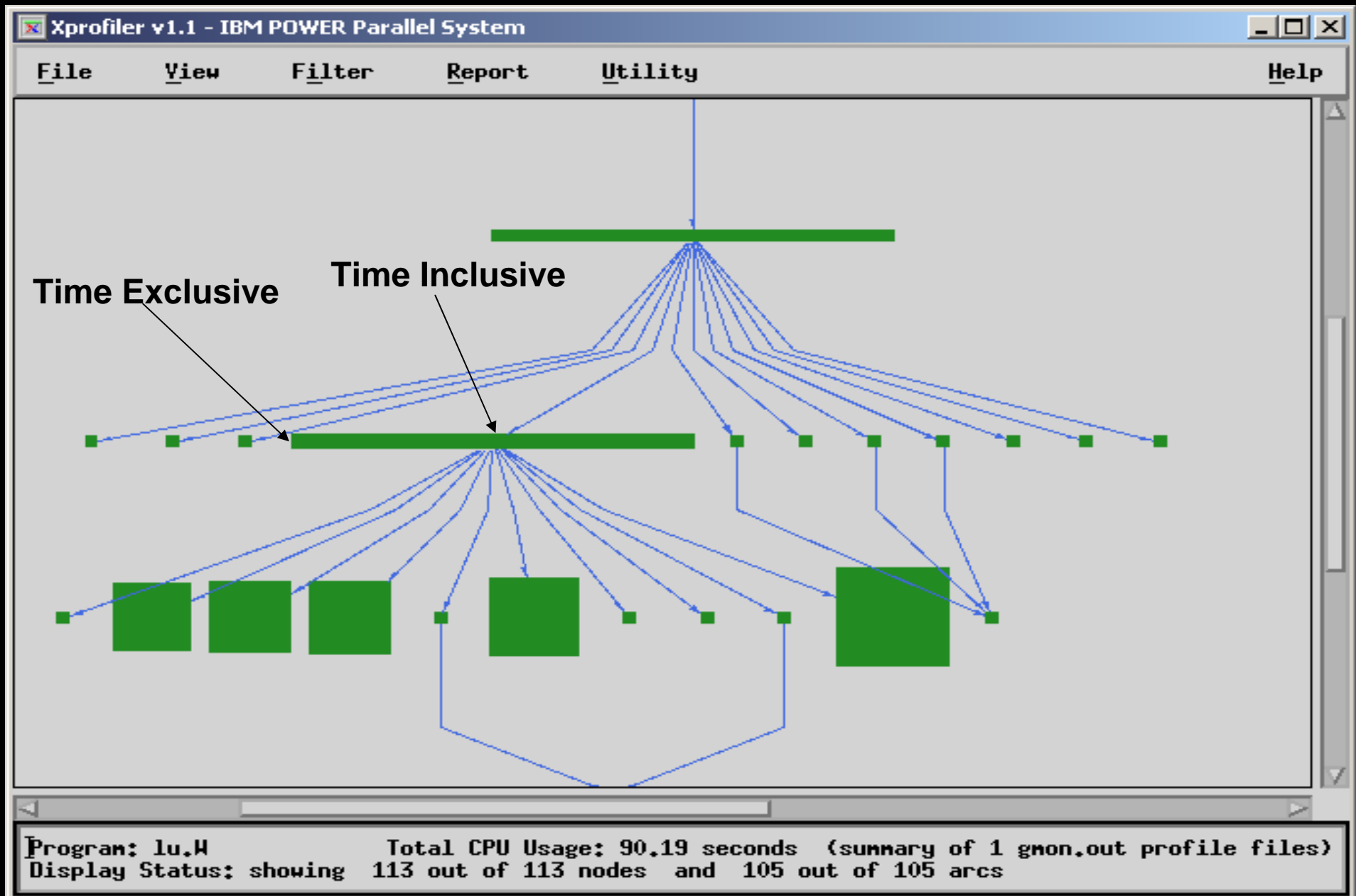
Xprofiler Tips (continued)

- **Select:**
 - Filter ; Uncluster Functions
 - Filter ; Hide All Library Calls
- **Select:**
 - Filter ; Filter by CPU Time
 - Filter ; by Call Count
- **If xprofiler fails with "bad font" error message:**
 - 1.) `edit /usr/lib/X11/app-defaults/Xprofiler`
 - replace `*narc*font: -ibm-block-...`
 - with `*narc*font: fixed`
 - 2.) `xrdb -load /usr/lib/X11/app-defaults/Xprofiler`

Xprofiler: Initial Screen



Xprofiler: Example



Xprofiler: Flat Profile

File	Code Display	Utility					Help
%time	cumulative seconds	self seconds	calls	self ms/call	total ms/call	name	
62.9	15.64	15.64	1	15640.00	15650.00	.main [1]	mmult.c
33.9	24.08	8.44	10	844.00	844.00	.thsub [3]	mtdsqmm.c
2.7	24.75	0.67				.datb [5]	datb.f
0.2	24.81	0.06				.dtrca [6]	dtrca.f
0.2	24.85	0.04				.durand [7]	durand.f
0.0	24.86	0.01	28	0.36	0.36	.fwrite_unlocked [9]	../../../../..
0.0	24.87	0.01				.dgetmo [12]	dgetmo.f
0.0	24.87	0.00	55	0.00	0.00	.leftmost [13]	../../../../..
0.0	24.87	0.00	43	0.00	0.00	.splay [14]	../../../../..
0.0	24.87	0.00	35	0.00	0.00	.malloc [15]	../../../../..
0.0	24.87	0.00	35	0.00	0.00	.malloc_y [16]	../../../../..
0.0	24.87	0.00	32	0.00	0.00	.free [17]	../../../../..
0.0	24.87	0.00	32	0.00	0.00	.free_y [18]	../../../../..
0.0	24.87	0.00	28	0.00	0.36	.fwrite [8]	../../../../..
0.0	24.87	0.00	28	0.00	0.00	.memchr [19]	../../../../..
0.0	24.87	0.00	16	0.00	0.00	.rightmost [20]	../../../../..
0.0	24.87	0.00	10	0.00	0.00	.mtdsqmm [21]	mtdsqmm.c
0.0	24.87	0.00	10	0.00	0.00	.splint [22]	../../../../..
0.0	24.87	0.00	10	0.00	0.00	.syncthread [23]	mtdsqmm.c
0.0	24.87	0.00	9	0.00	1.11	._doprnt [10]	../../../../..
0.0	24.87	0.00	9	0.00	0.00	._xflsbuf [24]	../../../../..
0.0	24.87	0.00	9	0.00	0.00	._xwrite [25]	../../../../..
0.0	24.87	0.00	9	0.00	1.11	.printf [11]	../../../../..
0.0	24.87	0.00	9	0.00	0.00	.time_base_to_time [26]	../../../../..

Search Engine: (regular expressions supported)

Xprofiler: Source Code View

Source Code for mtdsqmm.c

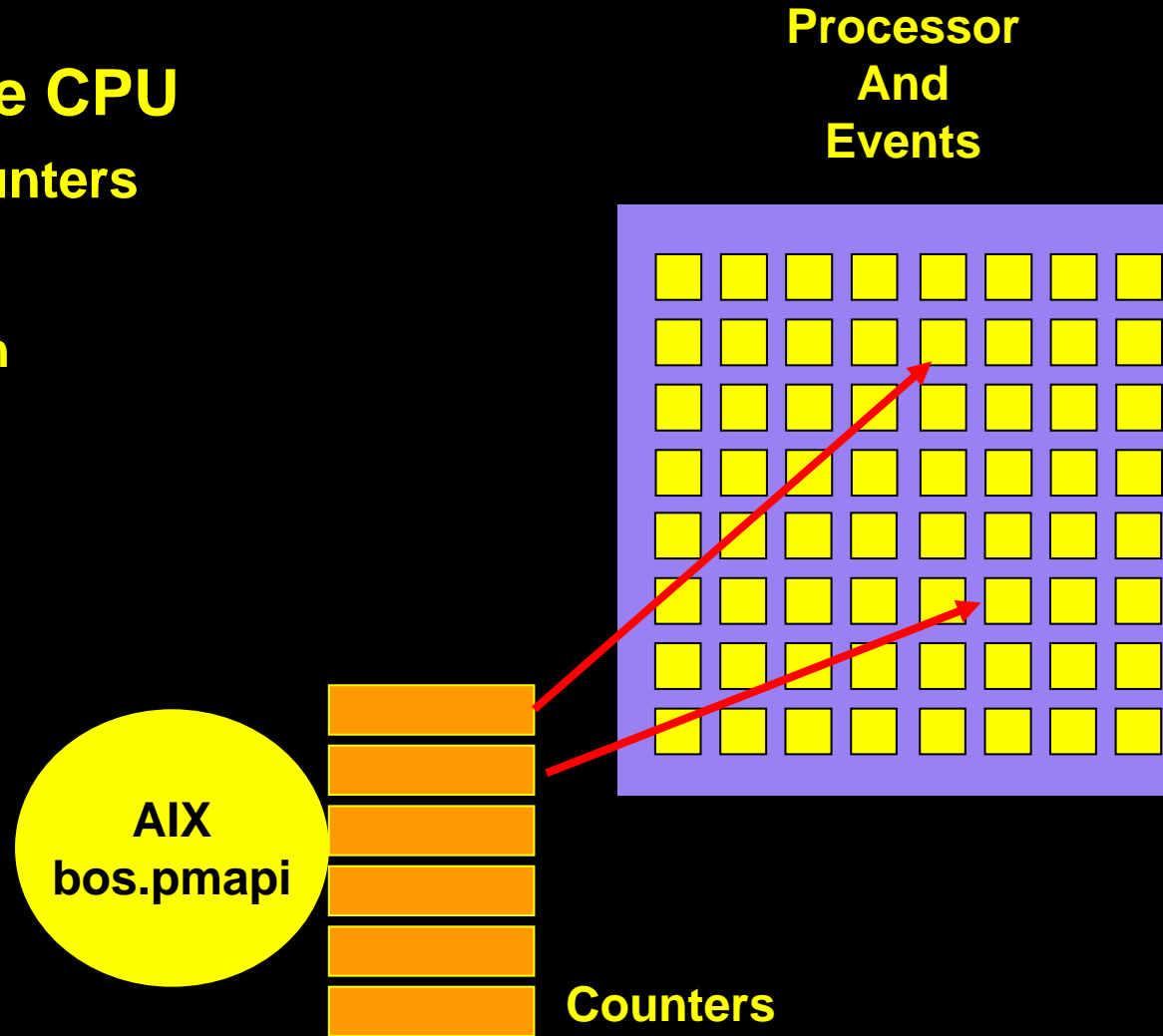
line	no. ticks per line	source code
202		/*-----*/
203		/* use 2x-unrolling of the outer two loops */
204		/*-----*/
205	4	for (i=i0; i<i0+is-1; i+=2)
206		{
207	8	for (j=j0; j<j0+js-1; j+=2)
208		{
209	1	t11 = c[i*n+j];
210	5	t12 = c[i*n+j+1];
211	5	t21 = c[(i+1)*n+j];
212	19	t22 = c[(i+1)*n+(j+1)];
213		for (k=k0; k<k0+ks; k++)
214		{
215	260	t11 = t11 + a[i*n+k]*bt[j*n+k];
216	116	t12 = t12 + a[i*n+k]*bt[(j+1)*n+k];
217	229	t21 = t21 + a[(i+1)*n+k]*bt[j*n+k];
218	144	t22 = t22 + a[(i+1)*n+k]*bt[(j+1)*n+k];
219		}
220	7	c[i*n+j] = t11;
221		c[i*n+j+1] = t12;
222	3	c[(i+1)*n+j] = t21;
223	5	c[(i+1)*n+(j+1)] = t22;
224		}
225		for (j=j; j<j0+js; j++)
226		{
227		t11 = c[i*n+j];
228		t21 = c[(i+1)*n+j];
229		for (k=k0; k<k0+ks; k++)
230		{
231		t11 = t11 + a[i*n+k]*bt[j*n+k];
232		t21 = t21 + a[(i+1)*n+k]*bt[j*n+k];
233		}
234		c[i*n+j] = t11;
235		c[(i+1)*n+j] = t21;
236		}
237		}

Search Engine: (regular expressions supported)

thsub

Hardware Performance Monitor

- **Counters are built into the CPU**
 - More than 250 (relevant) counters
 - Accessible through library
 - Require AIX kernel extension
 - Default in AIX 5
 - Accurate counts, but...
- **Instrumentation**
 - Whole program
 - Instrumentation point
- **Text output**
- **GUI output**



Hardware Performance Monitor and Tools

- **Contained in bos.pmapi file set**
- **System-level APIs**
 - **Allow counting of events**
 - **Whole machine**
 - **Set of processes with a common ancestor.**
- **First party kernel-thread-level APIs**
 - **Allow threads to count their own activity**
- **Third party kernel-thread-level APIs**
 - **Allow a debug program to count the activity of target threads**

Hardware Performance Monitor Tools

- **pmcycles**
- **pmlist**
- **hpmcount**
- **hpmstat**

pmcycles

- Returns the processor clock and decrementer speeds

```
$ pmcycles -d -m
```

```
Cpu 0 runs at 1656 MHz
```

```
Cpu 1 runs at 1656 MHz
```

```
Cpu 2 runs at 1656 MHz
```

```
Cpu 3 runs at 1656 MHz
```

```
Cpu 4 runs at 1656 MHz
```

```
Cpu 5 runs at 1656 MHz
```

```
Cpu 6 runs at 1656 MHz
```

```
Cpu 7 runs at 1656 MHz
```

```
The decrementer runs at 207.1 MHz (4.83 ns per tick)
```

pmlist

- **Information:**
 - **Processors**
 - **Events**
 - **Event groups**
 - **Derived metrics**

```
$ pmlist
```

```
usage: pmlist -h
```

```
    pmlist -l [ -o t | c ]
```

```
    pmlist -s | -e <short|select> | -c counter[,event] | -g group | -S set | -D
```

```
DerivedMetric
```

```
    [-p procname] [-s] [-d] [-o t|c] [-f filter]
```

```
where:
```

```
-h          this help screen
```

```
-l          lists all supported processor types
```

```
-s          displays processor information summary
```

```
-e short|select lists all events with this short name or select event value
```

```
-c -1      lists all events for all counters
```

```
...
```

hpmstat

- **System-wide event collection**
 - Raw and derived metrics
- **Default root user only**

hpmcount

- Executes applications
- Provides
 - Wall clock time
 - Raw and derived metrics
 - Operating system resource-utilization statistics

```
$ hpmcount a.out
```

```
...
```

```
##### End of Resource Statistics #####
```

```
...
```

Utilization rate	:	95.810 %
MIPS	:	1294.629
Instructions per cycle	:	0.816
HW Float point instructions per Cycle	:	0.526
HW floating point / user time	:	870.450 M HWflop/sec
HW floating point rate (HW Flops / WCT)	:	833.976 M HWflops/sec

hpmcount -s 1

```
$ hpmcount -s 1 a.out
```

```
Utilization rate           :      98.474 %  
MIPS                       :      2743.325  
Instructions per cycle     :         1.465  
HW Float point instructions per Cycle :         0.676  
HW floating point / user time : 1285.383 M HWflop/sec  
HW floating point rate (HW Flops / WCT) : 1265.764 M HWflops/sec
```

hpmcount -s 2

Utilization rate	:	98.183 %
Total load and store operations	:	21967.949 M
Instructions per load/store	:	1.576
MIPS	:	2730.613
Instructions per cycle	:	1.462
% Instructions dispatched that completed	:	98.552 %

hpmcount -s 3

```
Utilization rate           :      98.293 %
Total L2 data cache accesses :    6276.074 M
% accesses from L2 per cycle :      26.594 %
L2 traffic                 :    766122.306 MBytes
L2 bandwidth per processor :    60695.238 MBytes/sec
MIPS                       :      2742.726
Instructions per cycle     :           1.467
```

hpmcount -s 4

number of loads per TLB miss	:	23797159.282
Total load and store operations	:	21980.992 M
Instructions per load/store	:	1.575
MIPS	:	2748.390

hpmcount -s 5

Number of loads per TLB miss	:	23797159.282
Total load and store operations	:	21980.992 M
Instructions per load/store	:	1.575
MIPS	:	2748.390

Debuggers

- **Default (standard AIX):**
 - dbx
 - pdbx
- **Commercial:**
 - TotalView
 - ddt

Debuggers

- **Examine core file**
 - “where”
- **Start and run programs**
- **Attach to running programs**
 - **THIS IS COOL!**
 - **MPI program hangs? Use pdbx**
- **Program stop on specified conditions**
- **Examine stopped state**
- **Change things in program**
 - **Experiment with correcting the defects**

AIX Default Debugger: dbx

- **Symbolic debugger for languages:**
 - C
 - C++
 - Fortran
- **Examine object and core files**
- **Trace back**
- **Controlled breakpoints**
- **Syntax:**
 - **dbx [-a Process ID] [-r] ... [Object file [corefile]]**

Run Program Under dbx

```
$ dbx a.out
```

```
(dbx) run
```

```
Segmentation fault in test_sub at line 18 in file "testprog.f"
```

```
18          array(1,1) = 1.
```

```
(dbx) where
```

```
test_sub(array = (...), nvar = 2, nrec = 3), line 18 in "testprog.f"  
main(), line 10 in "testprog.f"
```

```
(dbx) print array(1,1)
```

```
reference through nil pointer
```

Attach Program Under dbx

```
$ ps
  UID  PID  TTY  TIME CMD
 206 454854 pts/8 0:09 a.out
$ dbx -a 454854
Waiting to attach to process 454854 ...
Successfully attached to a.out.
reading symbolic information ...
stopped in __divi64 at 0xd03a9d64
0xd03a9d64 (__divi64+0xec) 41800010      blt  0xd03a9d74 (__divi64+0xfc)

(dbx) print icount
421440793
(dbx) print l
0
(dbx) assign l=4
(dbx) print l
4
```

Summary

- **Operating system:**
 - AIX (Unix)
- **Components:**
 - PSSP, PE, Batch Queue, XL compilers
- **Interrogation tools:**
 - Iscfg, Islpp
- **Performance analysis:**
 - xprofiler
 - Hardware Performance Monitor
- **Debugger:**
 - dbx
 - TotalView