

Introducing the IBM Software Development Kit for PowerLinux

Wainer S. Moschetta

IBM, PowerLinux SDK Team Leader

wainersm@br.ibm.com

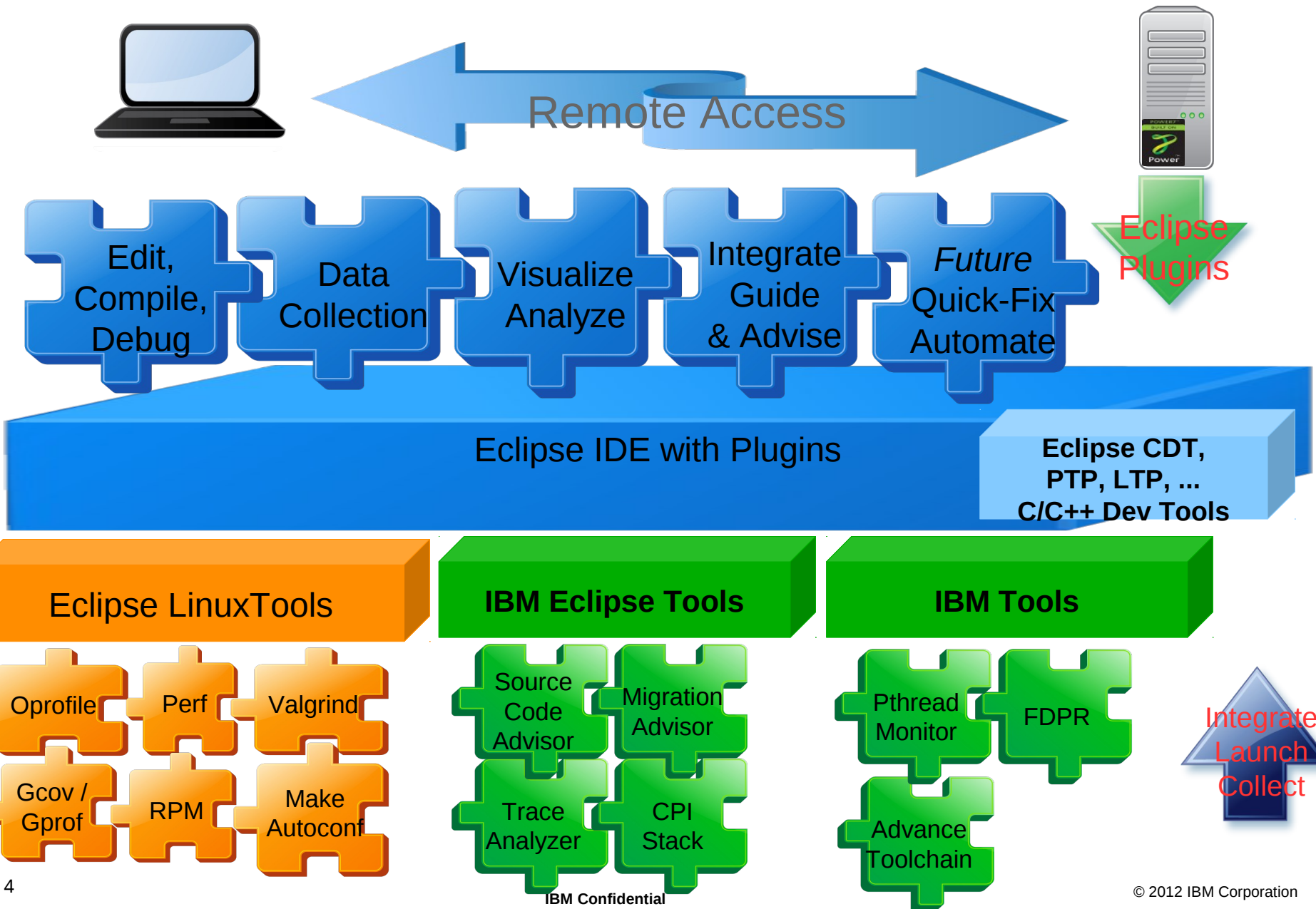


Acknowledgments

The information in this presentation was created with the assistance of Steve Munroe (sjmunroe@us.ibm.com), he is the PowerLinux SDK and Advance Toolchain architect.

- Support single source, cross platform, Linux applications
 - Port existing Linux Intel applications to PowerLinux
 - Develop new PowerLinux applications
 - Tune Linux applications for POWER
- Leverage Eclipse to Integrate existing Linux build and development tools
 - Import and use existing autoconf and Makefile projects
 - Integrates existing Linux Tools with Eclipse IDE
 - GCC, GDB, Oprofile, Perf, Valgrind, ...
- Adds powerful PowerLinux specific tools
 - AT / GCC and associated tools/components fully enable and tuned for POWER
 - Improve/extend the function/usability of Linux tools
 - Tools to identify and convert Intel specific source to:
 - Platform independent GCC builtins
 - Correct struct/union for Big & Little Endian
 - Provide equivalent PowerISA builtins
 - Tools to identify poorly performing code and propose appropriate changes
 - In source context using meaningful terms
 - Identify lock contention and associate it specific locks and source files/lines
 - High level analysis of program behavior via CPI-Stack model
- It is a process, not an event
 - Continually adding features, improving function, lean/agile development
 - Driven by first hand experience and customer feedback

Software Development Kit *concept*



The quintessential development environment

- Standard Eclipse Integrated development Environment (IDE)
 - Extensible via plugins
 - Common look & feel across tools
 - Integrated help, accessibility, usability features
- Additional Eclipse.org plugins
 - C/C++ development tools (CDT) (Edit compile debug)
 - Linux Tools Project (Linux tool; automation, visualization, jump to source line)
 - Import standard Makefile and autoconf projects
 - Parallel Tools Project (remote PowerLinux server access)
- Enhanced with PowerLinux tools
 - Analyzer and Advisor Plugins
 - Migration Advisor (cross platform code porting with Quick-Fix)
 - Source Code Advisor (guided application tuning for POWER)
 - Trace Analyzer (analyze bottlenecks in threaded applications)
 - POWER7 CPI Stack model (with Drill Down to source/file)
 - PowerLinux community message board tool
 - Supporting tools (integrates with plugins above)
 - IBM Advance toolchain (latest GCC, tuned libraries, perf tools, multi-core libraries)
 - Feedback Directed Program Restructuring (FDPR)
 - Pthread Monitor trace tool

Introducing the new PowerLinux™ SDK

What's new in 1.3.0

- IBM Eclipse SDK 4.2.0
 - Updated CDT, PTP, Linux Tools
- Enhanced Migration & Source Code Advisors, added quick-fixes
- FDPDR 5.6.1-9
- CPI analysis tool with drill-down
- Advance Toolchain 6.0
- New Integrated bug report

Available as:

- ISO image
- RPM packages
- YUM packages

IBM Java VM 1.6 included!!!



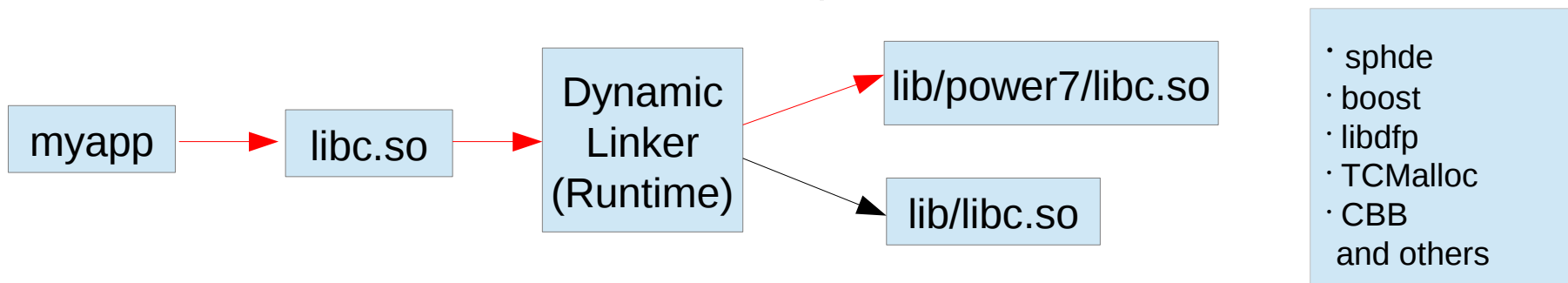
All in one place: the best tooling for Linux on POWER development

Give it a try and let us know how it goes:

<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/sdklop.html>

■ IBM Advance Toolchain for PowerLinux

- Provides latest stable versions
 - Base toolchain (GCC, GDB, Binutils..etc)
 - Runtime library (GLIBC)
 - Performance tools (Valgrind and Oprofile)
- Provides CPU-tuned libraries for performance

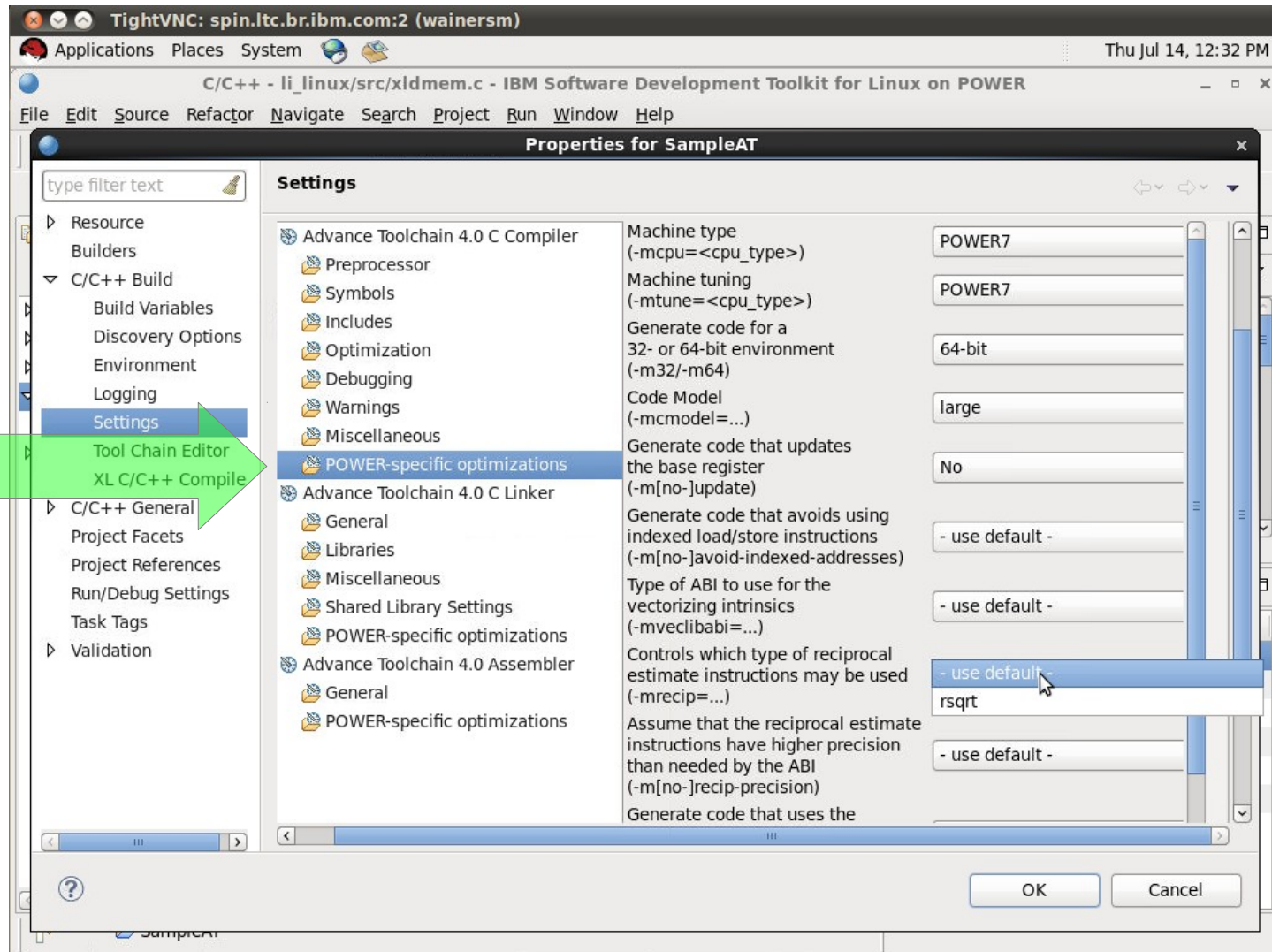


– Don't interfere with system's toolchain

```
[root@cheerios4p02 ~]# ls /opt/at6.0/  
bin bin64 etc include lib lib64 libexec libexec64 man powerpc64-linux sbin sbin64 scripts share ssl  
[root@cheerios4p02 ~]# ls /opt/at7.0/  
bin bin64 etc include lib lib64 libexec libexec64 man powerpc64-linux sbin sbin64 scripts share ssl
```

Latest Advance Toolchain 7.0-0 available!
Download at <ftp://ftp.unicamp.br/pub/linuxpatch/toolchain/at>

Provided more specific tunings as needed



- Code Migration Advisor plugin
 - Integrated with Eclipse context sensitive source tooling
 - Leverage CDT Codan (Code Analysis)
 - Integrated with source edit tools
 - Leverage CDT Quick Fix Processor to provide code fixes
 - Scan/Analyze application source for common migration issues
 - Data Endian dependent unions and structures
 - Cast with potential endian issues
 - Non-portable data types
 - Non-portable inline assembler code
 - Non-portable or arch dependent compiler builtins
 - Proprietary/Arch specific APIs
 - Performance degradation
 - Apply quick fixes
 - Common Intel specific inline assembler sequences
 - Non-portable or arch dependent vector builtins

See common migration problems in the code

The screenshot shows the IBM Software Development Kit for PowerLinux interface. The main editor displays the code for `mesh.h`, featuring a structure definition for `Mesh_Triangle_Struct` with fields like `Smooth`, `Dominant_Axis`, `vAxis`, `ThreeTex`, `Normal_Ind`, and `P1, P2, P3`. A green arrow points from the `libquantum-1.0.0` folder in the Project Explorer to the `mesh.h` file. The Migration Advisor View window at the bottom shows a table of migration warnings:

Description	Resource	Path	Location	Migration Adv
Union with endianness issues (21 items)				
Cast with endianness issues (100 of 1160 items)				
Performance Degradation (2 items)				
Struct with BitFields (8 items)				
Check for possible problems related to this struct with	mesh.h	/povray	line 78	Struct with Bit
Check for possible problems related to this struct with	tif_vms.c	/povray	line 338	Struct with Bit
Check for possible problems related to this struct with	tif_vms.c	/povray	line 343	Struct with Bit
Check for possible problems related to this struct with	tif_vms.c	/povray	line 355	Struct with Bit

A green callout bubble labeled "Problems report" points to the Migration Advisor View window. A status bar at the bottom indicates "Check for possible problems related to this struct with bit fields".

Apply quick fixes for migration problems in the code

The screenshot shows an IDE window with a C/C++ project named 'lock'. The main editor displays the source code for 'lock.c'. A green arrow points to the `asm("xchgl %0,%1"` line. A blue popup window titled 'Asm quick fix' is open over this line, suggesting a replacement with `_sync_lock_test_and_set`. The build console at the bottom shows compilation errors: 'Error: Unrecognized opcode: `xchgl`' and 'Error: Unrecognized opcode: `pause`'.

Change preview

May be replaced by a built in.
Suggestion: `_sync_lock_test_and_set`
For more information, consult:
<http://gcc.gnu.org/onlinedocs/gcc-4.1.1/gcc/Atomic-Builtins.html>

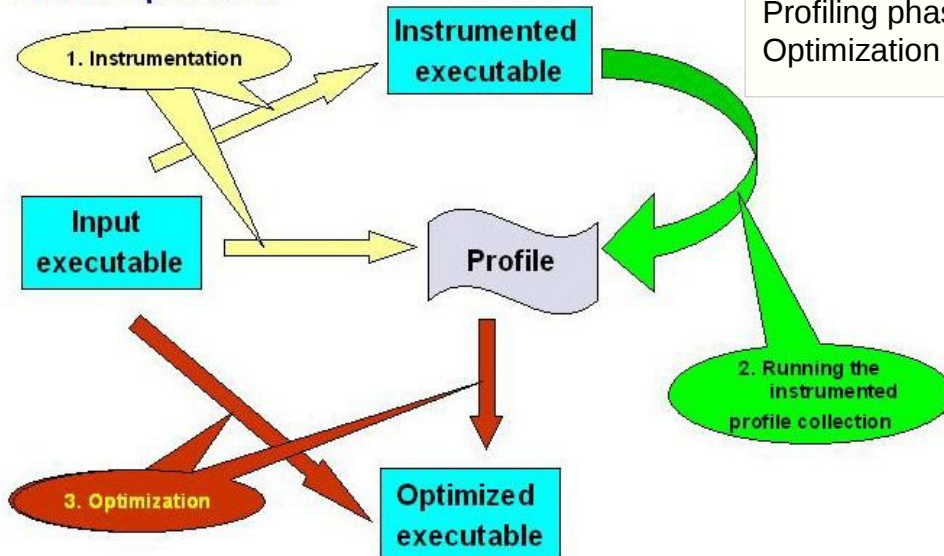
Replacement

```
int xchg(unsigned x) {  
    /*asm("xchgl %0,%1"  
        : "=r" (x), "=m" (lck)  
        : "m" (lck), "0" (x)  
        : "memory");*/  
    //TODO: Fill the parameters  
    _sync_lock_test_and_set(/*type *ptr, type value, ...*/);  
    return x;  
}
```

▪ FDPR (Feedback Directed Program Restructuring)

- FDPR is a feedback-based, directed, and post-link optimization tool
- Usable from command line or from SDK (eclipse plug-in)
- Works on both executable programs and shared libraries
- Provides post-link global code optimization step
- Tunes program to a representative workload

FDPR operation



```
$ fdpr --instrument --train ./train_bzip2 --reset --optimize -O3 -m power6 ./bzip2
Instrumentation phase ...
Profiling phase ...
Optimization phase ...
```

- Source Code Advisor (Eclipse plugin)
 - Leverages FDPR Inter-Procedural-Analysis capabilities
 - Provides interactive feedback to the developer
 - In plain language and in source code context
 - Identifies hot spots in source code that need rework. Some examples:
 - High call overhead of a hot small function (inline function)
 - High branch penalty in a small loop (unroll loop)
 - Data cache pressure is caused by TOC-load instructions (Direct TOC access)
 - Heavy register-save prolog with dominant early exit (Reduce early exit)
 - (...)
 - Propose specific suggestions for:
 - Source code structure improvements
 - Compiler/linker options to use

Improve code efficiency with Source Code Advisor



The screenshot displays the IBM Software Development Kit for PowerLinux interface. The main window shows the source code of a file named `exclude.c`. The code includes a function `excluded_file_name` that checks if a file is excluded based on certain conditions. The Source Code Advisor (SCA) panel at the bottom provides a detailed analysis of the code's performance, highlighting a "High call overhead of a hot small function" problem. The SCA panel shows a tree view of the analysis results, with the following data:

- FIX LOAD-HIT-STORE
- INLINE FUNCTION
 - 26.44% in `process_file()` [/home/wainersm/demo_projects/coreuti]
 - 6.63% on line 428
 - callee: line 420 in `excluded_file_name()` [exclude.c], xcount:
 - 6.63% on line 487
 - 6.63% on line 487
 - 6.54% on line 435
 - 20.19% in `fts_build()` [/home/wainersm/demo_projects/coreutis-8
 - 13.66% in `duinfo_add()` [/home/wainersm/demo_projects/coreuti]

The SCA panel also provides a solution: "Compiler: inline callee into caller - replace call to callee with its body". A green arrow points to the SCA panel, highlighting the problem and solution.

- Pthread_mon (command line tool)
 - High performance pthread (create, lock/unlock, condvar, etc) tracing
 - Multiple threads and processes
 - Selectable trace by API, levels of trace-back, ...
- Trace Analyzer (eclipse plugin)
 - Use to Identify lock contention and associate it specific locks and source files/lines
 - Correlates and displays traces
 - Pthread Monitor, SystemTap syscall, or both
 - Spot bottlenecks, IO/sleep/yield while holding mutex, ...
 - Multiple views
 - Thread Overview, Locks by thread, Hot Locks, Hot condvar, and more

Analyze thread usage using the Trace Analyzer

The screenshot displays the Trace Analyzer interface for a pthreads trace. The main window shows a timeline of thread execution with various colored bars representing different thread states. A callout labeled "Threads" points to these bars. On the left, a "Navigator" pane shows a tree view of the trace structure, with a callout labeled "View tabs" pointing to the "COND" folder. Below the Navigator is a "Trace Table" showing a list of records with columns for Index and Record Type. A callout labeled "Events color map" points to a table in the bottom right corner that maps event names to colors.

Name	Color
Thread	Grey
Start	Purple
Exit	Brown
Join	Green
Mutex_lock	Pink
Condvar_wait	Light Green

The "Trace Table" shows the following records:

Index	Record Type
62	Mutex
63	Mutex_Status
64	Element_Hold
65	Join

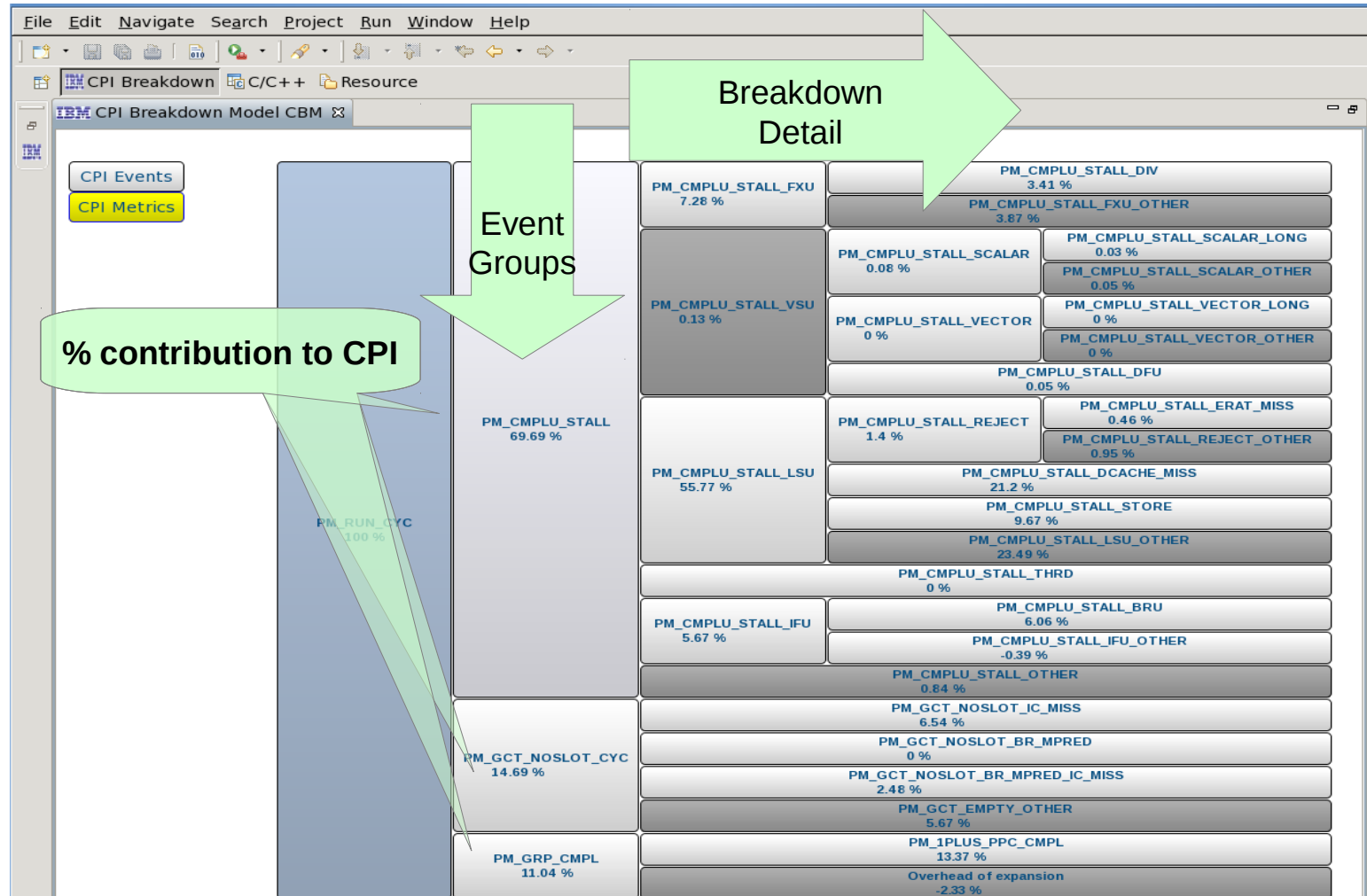
The "Record Details" pane shows the following log output:

```
<terminated> COND (1) [Trace Analyzer for pthreads] /opt/ibm/pthread-mon/  
SASAllocateSemNameProj(/tmp/sasdata20120411150917, 4c, 199);  
SASAllocateSemID(6c0006c3, 199); allocated ID = 950282  
SAS Lock Semaphores already initialized  
context already exists  
find lfLog = 8796361457664 in context  
after SPHContextFindByName  
before create iterator  
tid = 16054, pthread_tid = 17590124940448  
tid = 16062, pthread_tid = 17590114447808  
tid = 16061, pthread_tid = 17590124933568
```


■ CPI Tool

- Diagnosis tool that relates functional processor stages (pipeline) with performance counters to show which CPU functional unit is hitting stall conditions
 - Leverage PMU(Performance Monitoring Unit) for hardware events analysis in a systematic way
 - Implements CPI (cycles per instructions) breakdown model for POWER7 Systems
 - [Commonly Used Metrics for Performance Analysis](#) documentation available from Power.org
- Provides a top-level view of the applications performance
 - Useful for comparing programs or versions of the same program
 - Clues to which hardware PMU events to look at next
 - Drill-down to specific source/line for specific HW events

Cycles Per Instruction Breakdown Analysis

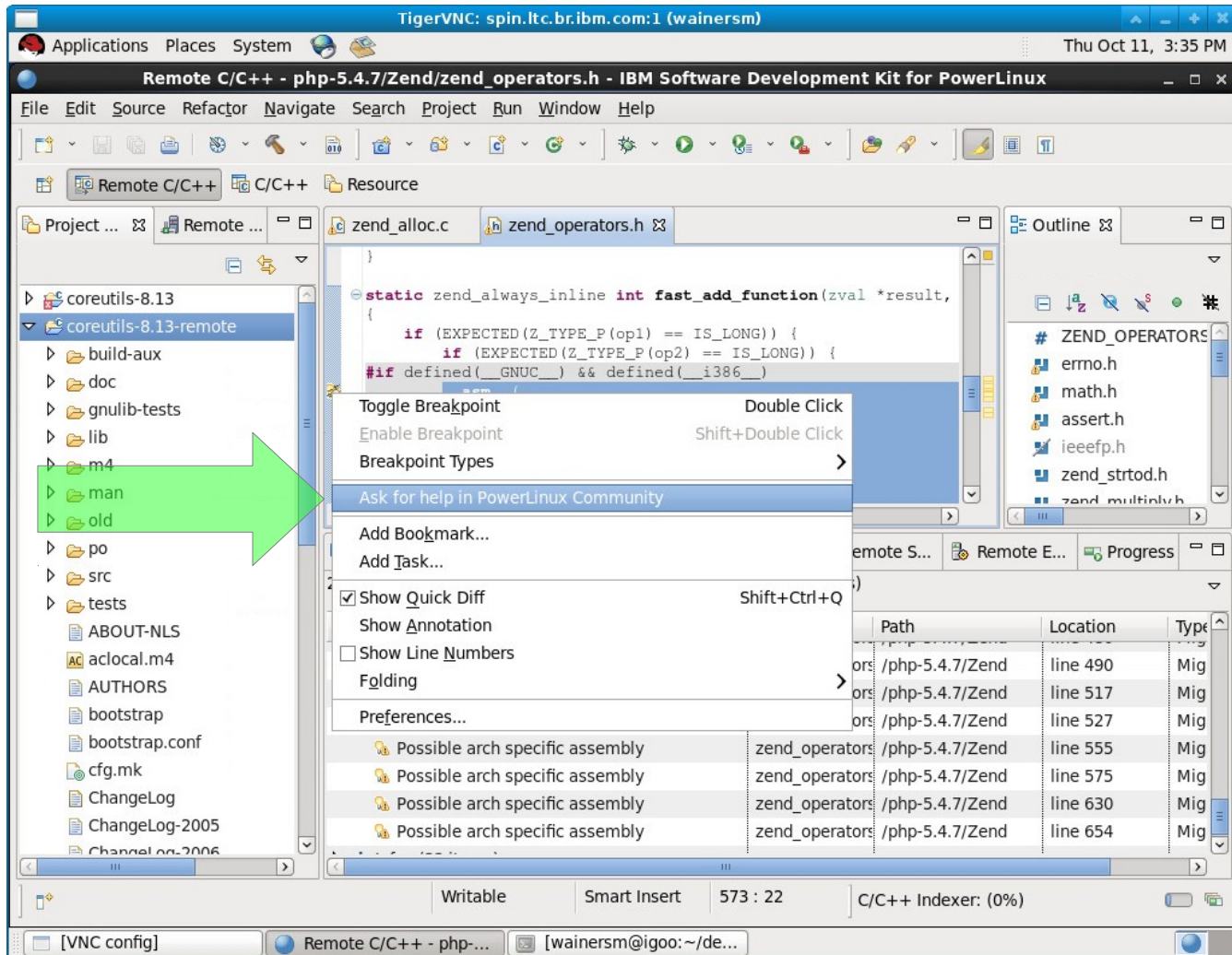


- Linux Tools Oprofile and Perf plugins
 - Launch and analysis integrated with code development
 - Configurable for HW specific event profiling
 - POWER6/7 PMU events

- Linux Tools Valgrind plugin
 - Launch and analysis integrated with code development
 - Open framework for dynamic analysis
 - Memcheck, detects memory leaks and malloc/free errors
 - Cachegrind, cache and branch miss analysis
 - Helgrind, thread and data race analysis
 - Massif, heap and stack usage analysis
 - PowerISA features for POWER6/7

- Ask for PowerLinux community help from within Eclipse (**NEW**)
 - Create a report that contains source code, error markers, and logs to be posted in the IBM developerWorks PowerLinux Community message board.
 - You can include specifics about your question or problem.
 - Leverage our experts

PowerLinux Community is willing to help you!



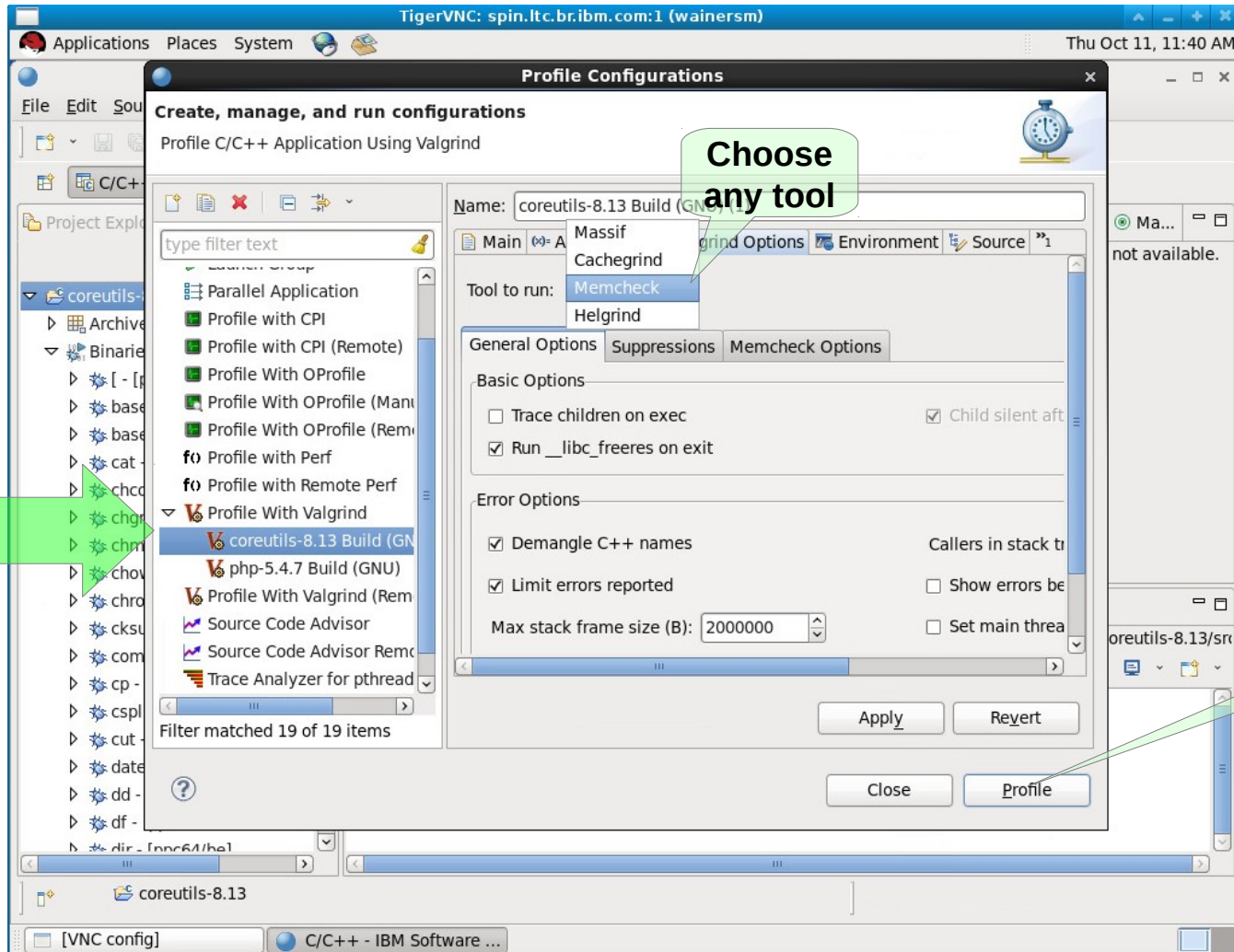
Profile application performance with ease

The screenshot shows the IBM Software Development Toolkit for Linux on POWER interface. The main editor displays the source code for the `add_pair_to_block` function in `bzlib.c`. The OProfile performance profiler is active, showing a call stack in the console. A green arrow points from the OProfile console to the function code in the editor.

```
void add_pair_to_block ( EState* s )
{
    Int32 i;
    UChar ch = (UChar) (s->state_in_ch);
    for (i = 0; i < s->state_in_len; i++) {
        BZ_UPDATE_CRC( s->blockCRC, ch );
    }
    s->inUse[s->state_in_ch] = True;
    switch (s->state_in_len) {
        case 1:
            s->block[s->nblock] = (UChar)ch; s->nblock++;
            break;
        case 2:
            s->block[s->nblock] = (UChar)ch; s->nblock++;
    }
}
```

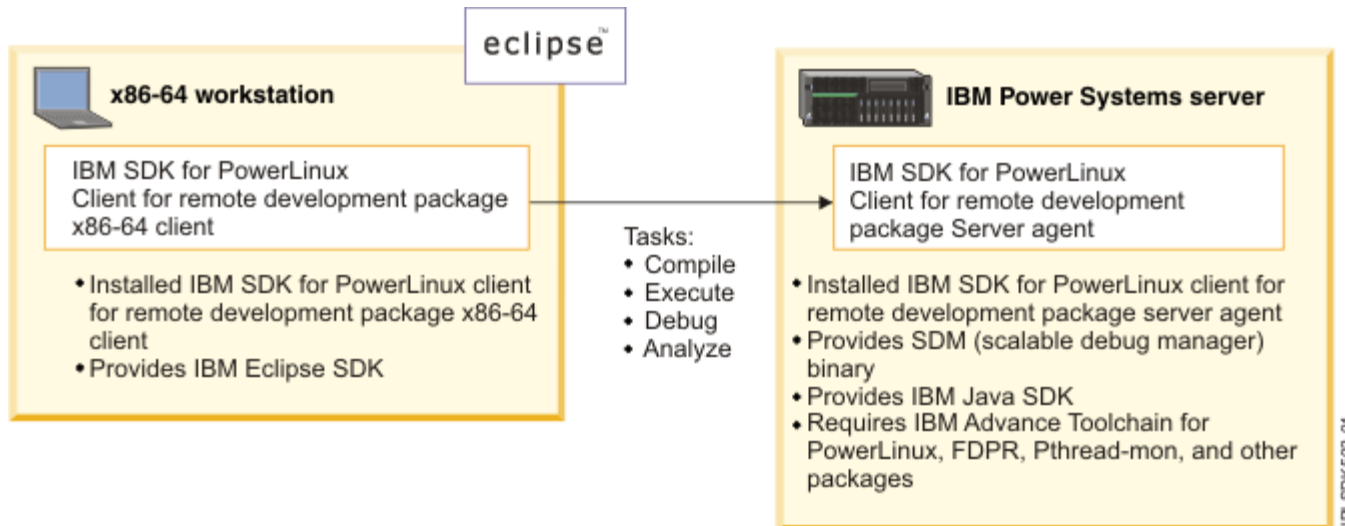
OProfile Console:

- CYCLES
- current
- 100.00% in /home/wainersm/sandbox/bzip2-1.0.6/bzip2
 - fo 54.22% in .mainSort [blocksort.c]
 - fo 34.26% in .BZ2_compressBlock [compress.c]
 - fo 8.94% in .handle_compress.clone.2 [bzlib.c]
 - fo 1.33% in .add_pair_to_block [bzlib.c]
 - 0.25% on line 221
 - 0.21% on line 220
 - 0.13% on line 235
 - 0.12% on line 241



Develop from x86_64 computer

- Use the IBM SDK for PowerLinux directly on the Power Systems™ server
- Also allow you to use the IBM SDK for PowerLinux from your personal x86_64 computer for development (remote development to the Power Systems™ server!)
- Leverage PTP remote tools and RDT



Our participation in Eclipse community

- Actively engaged with Linux Tools
 - Contributed Helgrind plug-in
 - Contributed Perf plug-in
 - Implemented remote for most of Linux Tools plug-ins
 - Helped with the implementation of remote proxy
 - Many bug reports and fixes
 - Currently three committers
- Few bug reports to PTP and one fix
- Few bug reports to CDT and a fix to Codan

- IBM Advance Toolchain for PowerLinux and IBM SDK for PowerLinux boost performance in IBM InfoSphere Streams on POWER
- The IBM InfoSphere Streams development team had a positive experience with the SDK.
 - InfoSphere Streams saw direct performance gains from using the Advance Toolchain compiler and optimized libraries.
 - Product code changes made as a result of SDK for PowerLinux application analysis further improved performance.
 - Performance hot spots in dependent Linux libraries were resolved by choosing alternative libraries or performance tuning critical runtime libraries.
 - These improvements were integrated and delivered in Advance Toolchain updates
- While results for other products will certainly vary, Customer related sample workloads built and executed within the InfoSphere Streams V3.0 product saw improved performance of between 26% and 166%

More information

- IBM Power Linux SDK landing page
<http://www14.software.ibm.com/webapp/set2/sas/f/lopdiags/sdklop.html>
- Introduction to IBM SDK LoP demo video
<http://www.ibm.com/developerworks/offers/lp/demos/summary/l-sdklinuxpower.html>
- Using Autotools with the IBM SDK LoP demo video
<http://www.ibm.com/developerworks/offers/lp/demos/summary/l-autotoolslinuxonpower.html>
- PowerLinux Community blog posts
<https://www.ibm.com/developerworks/mydeveloperworks/blogs/fe313521-2e95-46f2-817d-44a4f27eba32/tags/ibm-sdk-lop?lang=en>
- Free support – TPL Message Board
<http://www.ibm.com/developerworks/group/tpl>
- IBM PowerLinux SDK User Guide
<http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaal/iplsdkmain.htm>
- POWER7 Optimization and tuning Guide
<http://www.redbooks.ibm.com/redpieces/abstracts/sg248079.html>



Give it a try on your application!!!

- The new Power Linux Software Development Kit (SDK) provides a traditional GUI for developing, porting, and tuning applications
 - Eclipse-based
 - Complete bundle of tools
- Power-specific features have been added to existing tools
 - C/C++ projects, OProfile, Valgrind, Helgrind and more
- New tools have been added for Power development
 - Post-link optimization (FDPR), Source Code Advisor, Trace Analyzer, Migration Assist, CPI Tool
- Begin using this tool today on Power or x86 Systems and help us improve it as we move forward

Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Revised September 26, 2006

QA