



Software Group

Compilation Technology

IBM pSeries Compiler Roadmap

Roch Archambault
IBM Toronto Laboratory
archie@ca.ibm.com



Agenda

- The pSeries Compiler Products
- Roadmaps
 - Common Features & Compiler Architecture
 - XL Fortran
 - XL C/C++
 - Blue Gene
 - CELL
- Customer Requirements
- Multiple compiler installation
- Online documentation
- Performance Comparison
- Q&A



The pSeries Compiler Products: Latest Versions

- All POWER4, POWER5, POWER5+ and PPC970 enabled
 - XL C/C++ Enterprise Edition V8.0 for AIX
 - XL Fortran Enterprise Edition V10.1 for AIX
 - XL C/C++ Advanced Edition V8.0 for Linux
 - XL Fortran Advanced Edition V10.1 for Linux
- Blue Gene (PPC440) enabled
 - XL C/C++ Advanced Edition V8.0 for BG/L (PRPQ)
 - XL Fortran Advanced Edition V10.1 for BG/L (PRPQ)
- Technology Preview currently available from alphaworks
 - XL C/C++ for Cell Broadband Engine Processor
 - Download: <http://www.alphaworks.ibm.com/tech/cellcompiler>
 - XL UPC language support on AIX and Linux
 - Download: <http://www.alphaworks.ibm.com/tech/upccompiler>



The pSeries Compiler Products: 2006

- SLES 10 support
 - XL C/C++ Advanced Edition V8.0.1 for Linux
 - XL Fortran Advanced Edition V10.1.1 for Linux
- CELL cross compiler
 - XL C/C++ on Linux X86 V8.1 for Cell

All information subject to change without notice



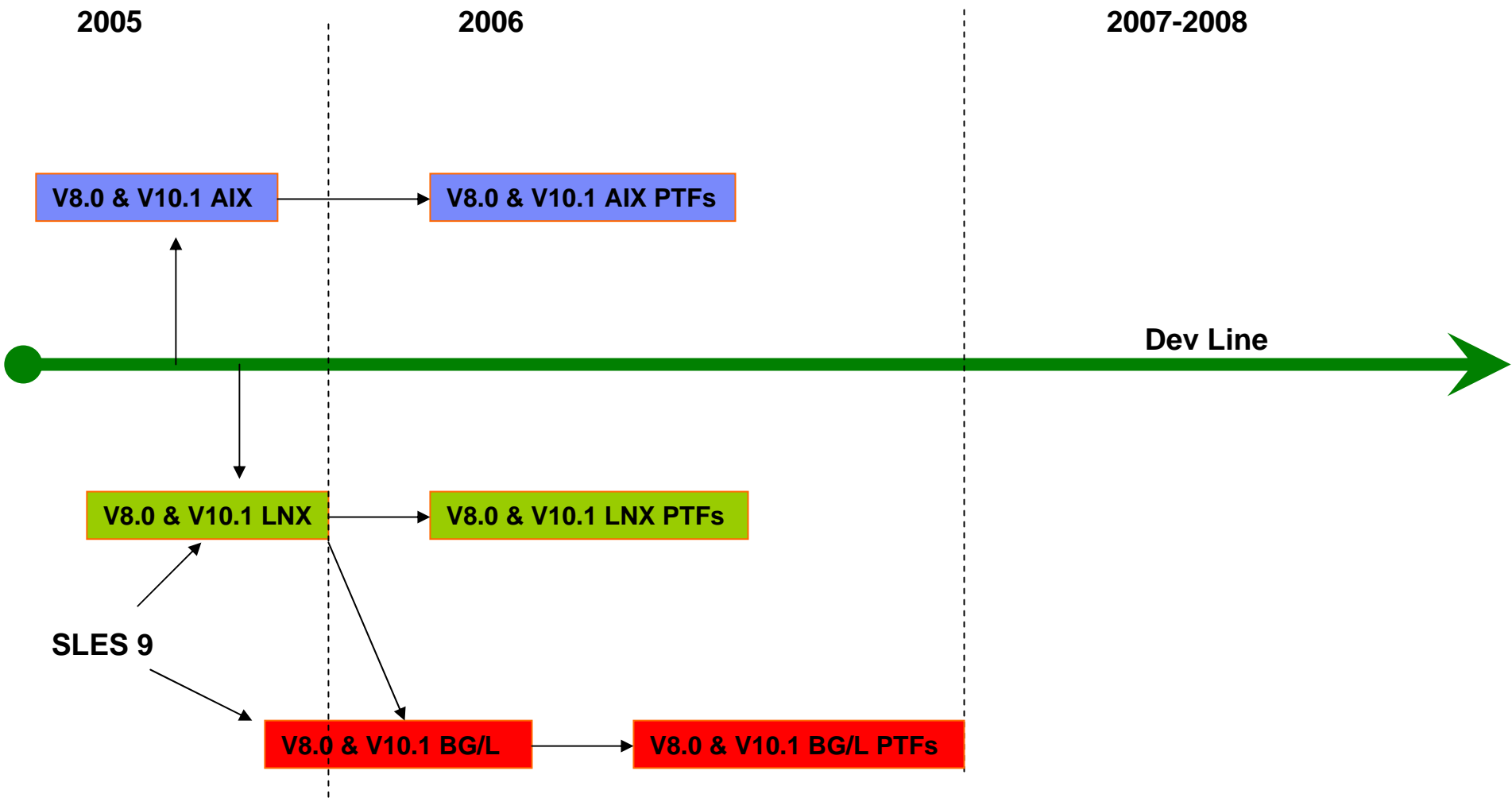
The pSeries Compiler Products: 2007 and 2008

- POWER6 enabled
 - XL C/C++ Enterprise Edition V9.0 for AIX
 - XL Fortran Enterprise Edition V11.1 for AIX
 - XL C/C++ Advanced Edition V9.0 for Linux (SLES 10 and RHEL 5)
 - XL Fortran Advanced Edition V11.1 for Linux (SLES 10 and RHEL 5)
- Blue Gene (PPC450) enabled
 - XL C/C++ Advanced Edition V9.0 for BG/P
 - XL Fortran Advanced Edition V11.1 for BG/P
- CELL cross compiler from Windows, Linux x86 and Linux PPC
 - XL C/C++ on Windows V9.0 for Cell
 - XL C/C++ on Linux X86 V9.0 for Cell
 - XL FORTRAN on Linux X86 V11.1 for Cell
 - XL C/C++ on Linux PPC V9.0 for Cell
 - XL FORTRAN on Linux PPC V11.1 for Cell

All information subject to change without notice



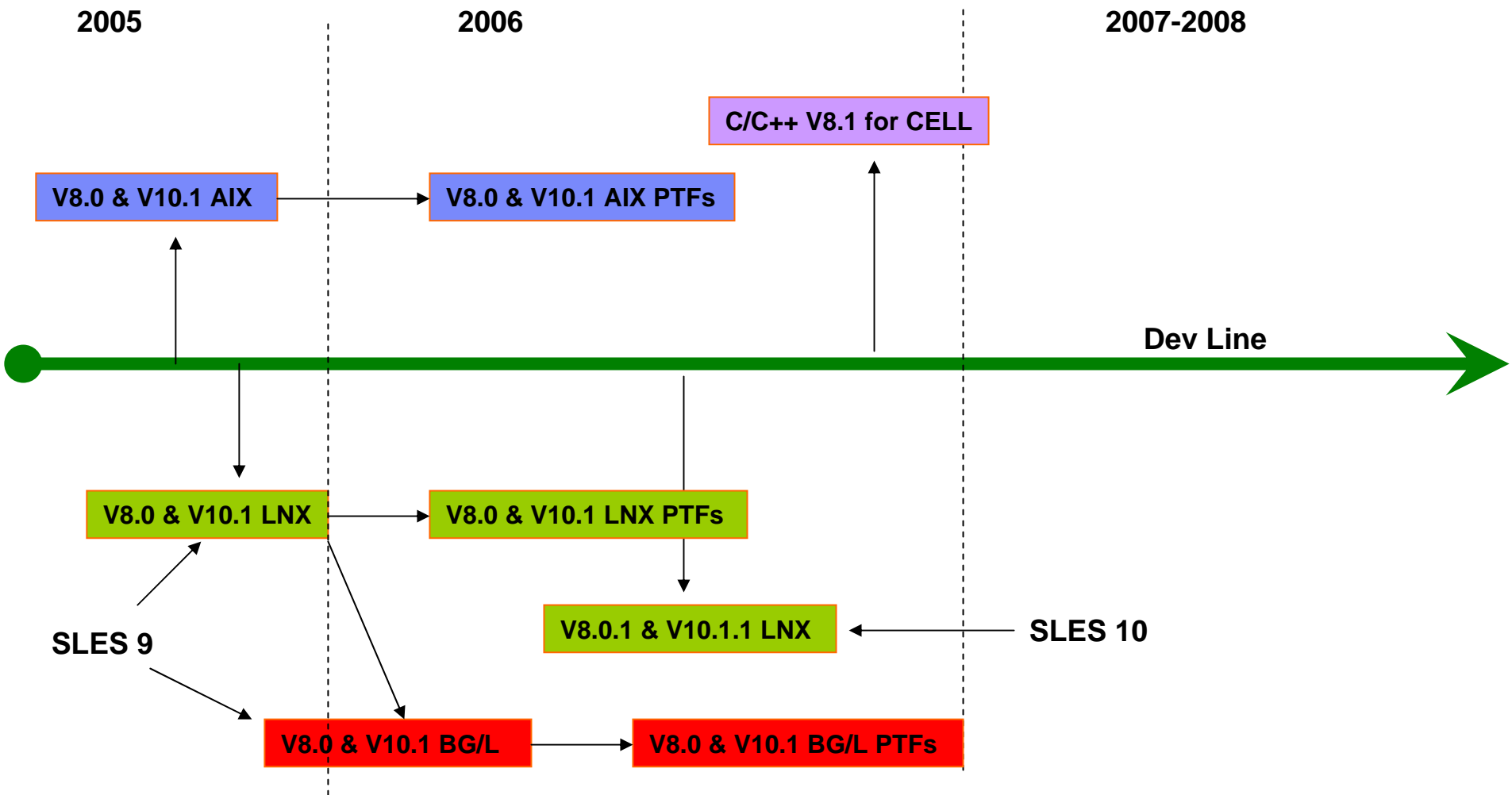
Roadmap of XL Compiler Releases



All information subject to change without notice



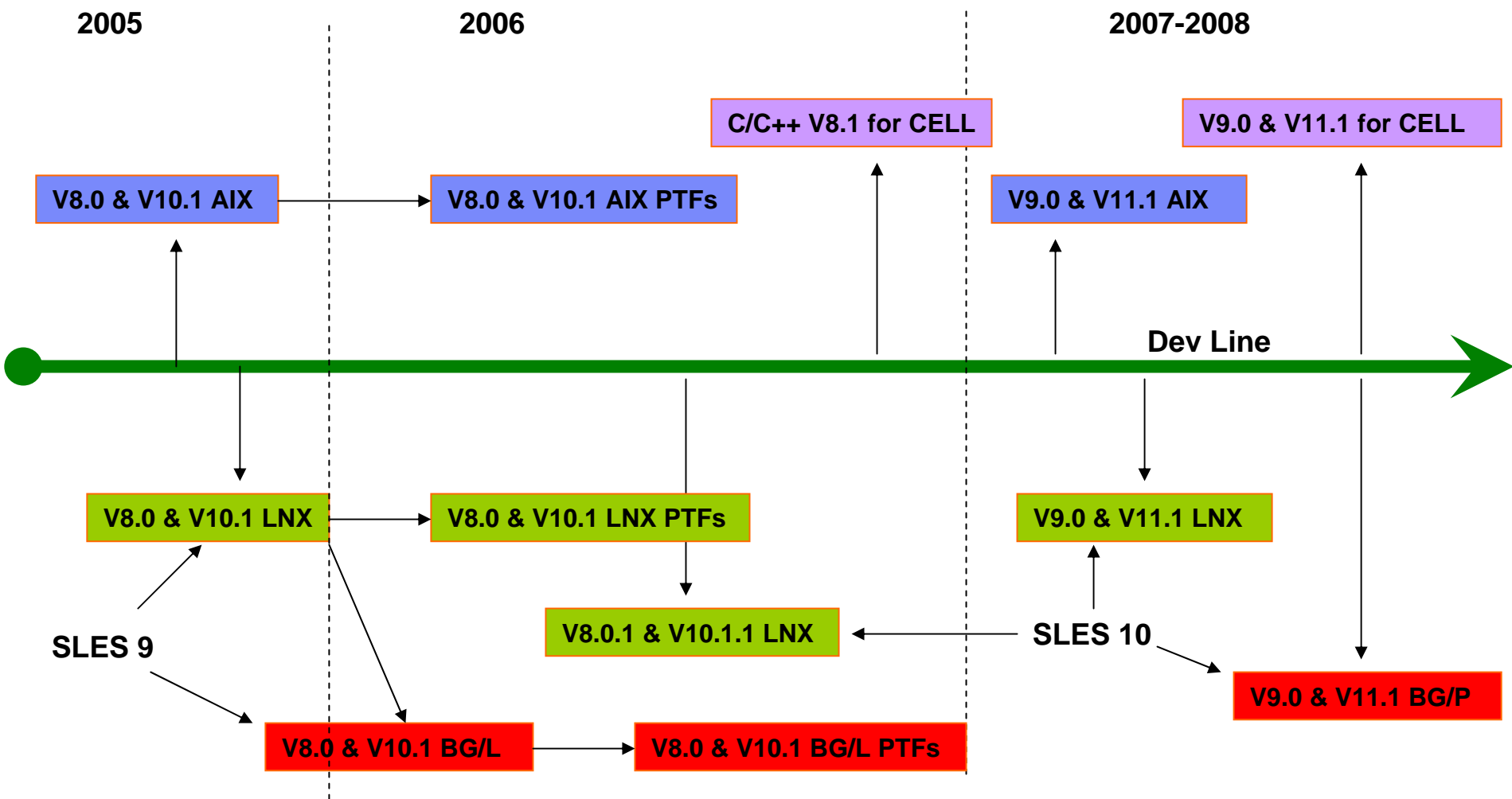
Roadmap of XL Compiler Releases



All information subject to change without notice



Roadmap of XL Compiler Releases



All information subject to change without notice

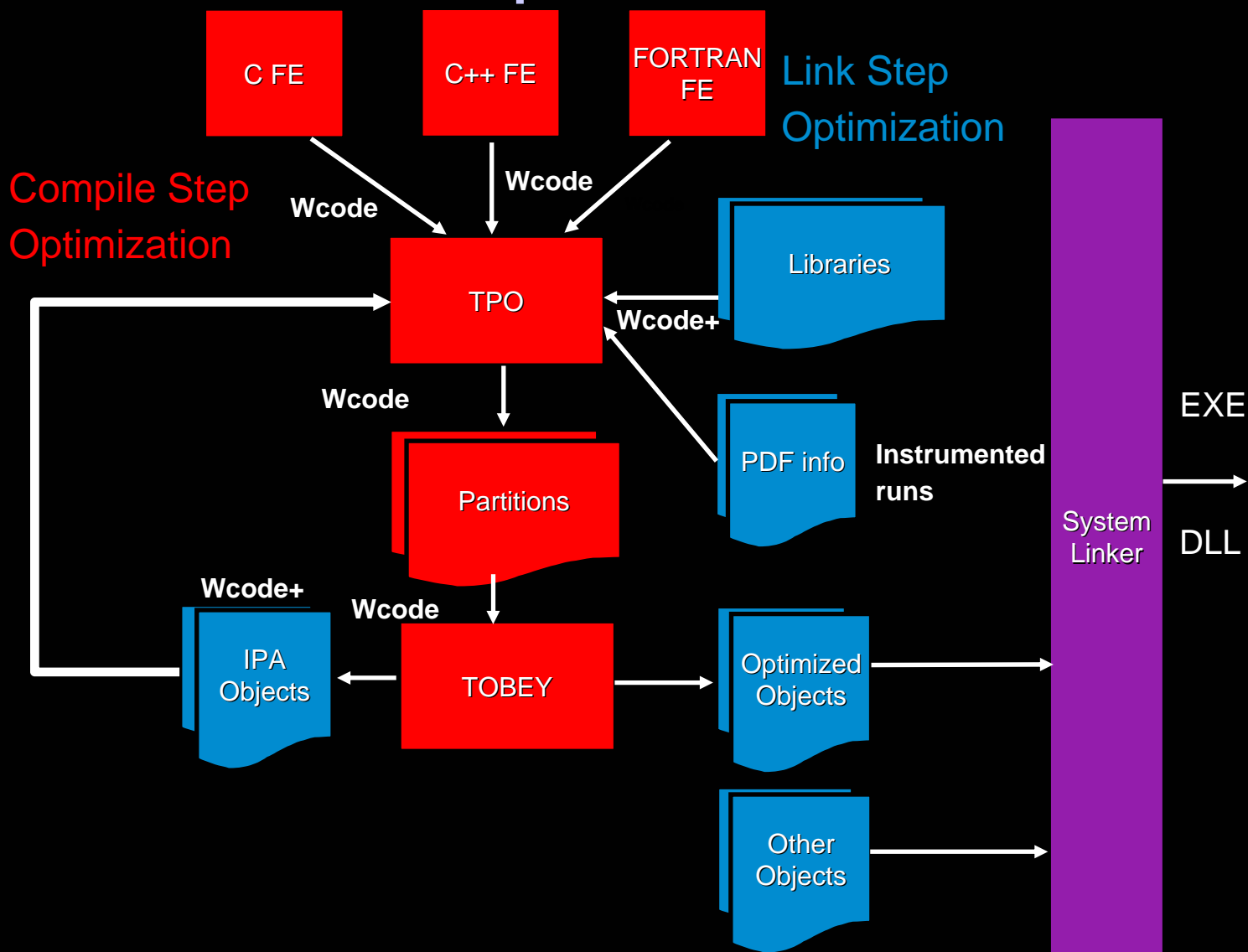


Common Fortran, C and C++ Features

- Linux (SLES and RHEL) and AIX, 32 and 64 bit
- Debug support
 - TotalView (Etnus), DDT (Allinea) and DBX on AIX
 - gdb on Linux
 - Full support for debugging of OpenMP programs
 - Snapshot directive for debugging optimized code
- Portfolio of optimizing transformations
 - Instruction path length reduction
 - Whole program analysis
 - Loop optimization for parallelism, locality and instruction scheduling
 - Use profile directed feedback (PDF) in most optimizations
- Tuned performance on POWER3, POWER4, POWER5, PPC970, PPC440 and CELL systems
- Optimized OpenMP



IBM XL Compiler Architecture





XL Fortran Roadmap: Strategic Priorities

- **Premium Customer Service**
 - Continue to work closely with key ISVs and customers in scientific and technical computing industries
- **Compliance to Language Standards and Industry Specifications**
 - OpenMP API V2.5
 - Fortran 77, 90 and 95 standards
 - Fortran 2003 Standard
- **Exploitation of Hardware**
 - Committed to maximum performance on POWER4, PPC970, POWER5, PPC440 and successors
 - Continue to work very closely with processor design teams



XL Fortran Version 10.1 for AIX/Linux – Fall/Winter 2005

- AIX Announcement Letter:
http://www.ibm.com/ishome/cgi-bin/goto?it=can_announ&on=A05-1365
- Continued rollout of Fortran 2003
- Compliant to OpenMP V2.5
- Generate multi-path code for different architecture (cloning for architecture)
- Perform subset of loop transformations at `-O3` optimization level
- Improved performance of quad precision floating point
- Support for BLAS routines (DGEMM and DGEMV) tuned for POWER4 and POWER5 are included in compiler runtime (libxlopt)
- Runtime check for availability of ESSL
- Intrinsic and data types for direct VMX programming



FORTRAN 2003 Support in XLF V10.1

- Data manipulation enhancements
 - ALLOCATABLE components (except resizing on assignment)
 - INTENT specifications of pointer arguments
 - PROTECTED attribute and statement
 - VALUE attribute and statement
 - procedure declaration statement (PROCEDURE statement)
 - relaxed specification expression**
- Support for IEC 60559 (IEEE 754) exceptions and arithmetic
 - IEEE_EXCEPTIONS, IEEE_ARITHMETIC and IEEE_FEATURES intrinsic modules
- Input/output enhancements
 - stream access (allows access to a file without reference to any record structure)
 - the FLUSH statement
 - the NEW_LINE intrinsic
 - access to input/output error messages (IOMSG= specifier on data-transfer operations, file-positioning, FLUSH and file inquiry statements)
 - BLANK= and PAD= specifiers on READ statement**
 - DELIM= specifier on WRITE statement**
- **Enumerations and enumerators**
- **Procedure pointers (except PASS attribute, declaring intrinsic procedure)**
- Derived-type enhancements
 - mixed component accessibility (allow PRIVATE and PUBLIC attribute on derived type components)
- Interoperability with C programming language
 - ISO_C_BINDING intrinsic module (except C_F_PROCPOINTER)
 - BIND attribute and statement
- The ASSOCIATE construct
- Scoping enhancement
 - the ability to control host association into interface bodies (IMPORT statement)
- Enhancement integration with the host operating system
 - access to command line arguments (COMMAND_ARGUMENT_COUNT, GET_COMMAND_ARGUMENT, and GET_ENVIRONMENT_VARIABLE intrinsics)
 - access to the processor's error messages (IOMSG= specifier)
 - ISO_FORTRAN_ENV intrinsic module



C/C++ Roadmap: Strategic Priorities

- **Premium Customer Service**
- **Compliance to Language Standards and Industry Specifications**
 - ANSI / ISO C and C++ Standards
 - OpenMP API V2.5
- **Exploitation of Hardware**
 - Committed to maximum performance on POWER4, PPC970, POWER5 and successors
 - Continue to work very closely with processor design teams
- **Exploitation of OS and Middleware**
 - Synergies with operating system and middleware ISVs (performance, specialized function)
 - Committed to AIX Linux affinity strategy and to Linux on pSeries
- **Reduced Emphasis on Proprietary Tooling**
 - Affinity with GNU toolchain



XL C/C++ Version 8.0 for AIX/Linux – Fall/Winter 2005

- AIX Announcement Letter:
http://www.ibm.com/ishource/cgi-bin/goto?it=can_announ&on=A05-1367
- Compliant to OpenMP V2.5
- Generate multi-path code for different architecture (cloning for architecture)
- Perform subset of loop transformations at –O3 optimization level
- Improved performance of quad precision floating point
- Support for BLAS routines (DGEMM and DGEMV) tuned for POWER4 and POWER5 are included in compiler runtime (libxlopt)
- Runtime check for availability of ESSL
- Support for auto-simdization and VMX intrinsics on AIX



GNU C/C++ Compatibility Enhancements

Full list of GNU C/C++ compatibility enhancements in XL C/C++ V8.0 can be found here:

http://publib.boulder.ibm.com/infocenter/comphelp/v8v101/index.jsp?topic=/com.ibm.xlcpp8a.doc/language/ref/gcc_cext.htm

- **Labels as values / computed goto**
- **Nested functions (C only)**
- **Naming types**
- **Conditionals with omitted operands**
- **Zero length arrays**
- **Labeled elements (C only)**
- **Case ranges (C only)**
- **Cast to union (C only)**
- **Function Attributes**
 - Support
 - Noinline, always_inline, format, format_arg, section
 - Accept and ignore
 - used



Tentative GNU C/C++ Compatibility Enhancements

- **Variable Attributes**
 - Support**
 - Nocommon, transparent_union**
- **Type Attributes**
 - Support**
 - Aligned, packed**
 - Accept and ignore**
 - Transparent_union**
 - **__extension__**
 - **Incomplete enums**
 - **Function names as strings**
 - **Partial Asm support**



Blue Gene Compilers

XL C/C++ Advanced Edition V8.0 for BG/L and XL Fortran Advanced Edition V10.1 for BG/L

- Performance tuning of SPEC2000FP, DDCMD Kernels, NAS 3.2 Serial and sPPM.
- Performance tuning of MASS library
- Exploit 440D instructions for complex arithmetic
- BG/L compiler white paper (Exploiting the Dual FPU in BG/L):
<http://www-1.ibm.com/support/docview.wss?uid=swg27007511>
- PTF1 compiler refresh:
 - Support Blue Gene software release 3
 - Overall SPEC2000FP faster for 440D than 440
 - Updated white paper to reflect PTF1 performance improvements
- Will continue to improve performance in future compiler refresh

XL C/C++ Advanced Edition V9.0 for BG/P and XL Fortran Advanced Edition V11.1 for BG/P

- Support for OpenMP

All information subject to change without notice



CELL Compilers

- **XL C/C++ on Linux X86 V8.1 for Cell**

 - Hosted on Linux x86 and Linux PPC

 - Support SDK 2.0 interfaces

 - Targets CELL Blade 1 hardware

- **Cross Compilers from Windows, Linux x86 and Linux PPC:**

 - Support SDK 3.0 interfaces

 - Targets CELL Blade 2 hardware

 - User directed single source compiler

 - Includes the following compilers:

 - XL C/C++ on Windows V9.0 for Cell

 - XL C/C++ on Linux X86 V9.0 for Cell

 - XL FORTRAN on Linux X86 V11.1 for Cell

 - XL C/C++ on Linux PPC V9.0 for Cell

 - XL FORTRAN on Linux PPC V11.1 for Cell

All information subject to change without notice



Customer Requirements Planned for 2006-2008

- Provide Filename and Line Number in ALLOC/DEALLOC Failure (Fortran)
- Provide Filename and Line Number in NAMELIST Failure (Fortran)
- Little-Endian Data I/O Support (Fortran)
- Thread Number in Standard Error output (Fortran)

All information subject to change without notice



Customer Requirements Planned for 2006-2008

- Improve performance of critical codes on BG/L
- Detect a thread's stack going beyond its limit (Fortran and C/C++)
- XLF 11.1 will deliver most (but not all) of the remaining F2003 standard
- Exploit “restrict” keyword in C 1999

All information subject to change without notice



Feature Request

- Request for a feature to be supported by our compilers
- C/C++ feature request page:
<http://www-1.ibm.com/support/docview.wss?uid=swg27005811>
- Fortran feature request page:
<http://www-1.ibm.com/support/docview.wss?uid=swg27005812>
- Or send e-mail to xl_feature@ca.ibm.com



Installation of Multiple Compiler Versions

- Installation of multiple compiler versions is supported
- The vacppndi and xlfndi scripts shipped with VisualAge C++ 6.0 and XL Fortran 8.1 and all subsequent releases allow the installation of a given compiler release or update into a non-default directory
- The configuration file can be used to direct compilation to a specific version of the compiler

Example: `xlf_v8r1 -c foo.f`

May direct compilation to use components in a non-default directory

- Care must be taken when multiple runtimes are installed on the same machine (details on next slide)



Coexistence of Multiple Compiler Runtimes

- **Backward compatibility**

C, C++ and Fortran runtimes support backward compatibility.

Executables generated by an earlier release of a compiler will work with a later version of the run-time environment.

- **Concurrent installation**

Multiple versions of a compiler and runtime environment can be installed on the same machine

Full support in xlfndi and vacppndi scripts is now available

- **Limited support for coexistence**

LIBPATH must be used to ensure that a compatible runtime version is used with a given executable

Only one runtime version can be used in a given process.

Renaming a compiler library is not allowed.

Take care in statically linking compiler libraries or in the use of *dlopen* or *load* .

Details in the compiler FAQ

<http://www.ibm.com/software/awdtools/fortran/xlfortran/support/>

<http://www.ibm.com/software/awdtools/xlcpp/support/>



Documentation

- An information center containing the documentation for the **XL Fortran V9.1** and **XL C/C++ V7.0** versions of the AIX compilers is available at:
<http://publib.boulder.ibm.com/infocenter/comphelp/index.jsp>
- An information center containing the documentation for the **XL Fortran V10.1** and **XL C/C++ V8.0** versions of the AIX compilers is available at:
<http://publib.boulder.ibm.com/infocenter/comphelp/v8v101/index.jsp>
- New **Optimization and Tuning Guide** for XLF V10.1 is now available online
- This information center contains all the html documentation shipped with the compilers. It is completely searchable.
- Please send any comments or suggestions on this information center or about the existing C, C++ or Fortran documentation shipped with the products to compinfo@ca.ibm.com.



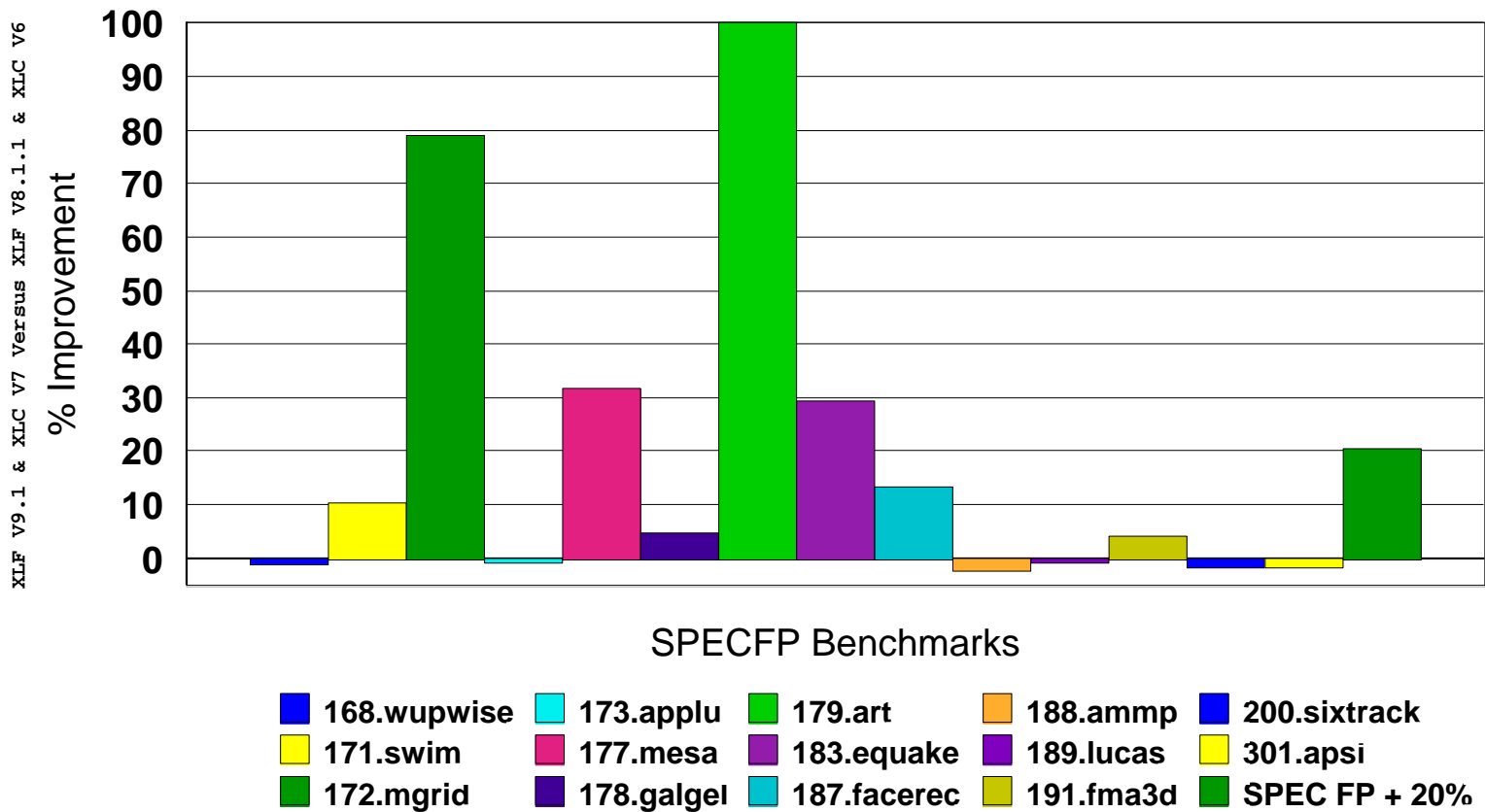
History Of Compiler Improvement On Power4

Compilers	2001 V5/V7.1.1	2002 V6/V8.1	2003 V6/V8.1.1	2004 V7/V9.1	2005 V8/V10.1	Compound Over 4 Years	AGC Rate
SpecINT	baseline	21%	0%	3%	7%	34%	7.6%
SpecFLOAT	baseline	12%	5%	18%	5%	46%	9.9%

Note: SPEC2000 base options improvements from www.spec.org

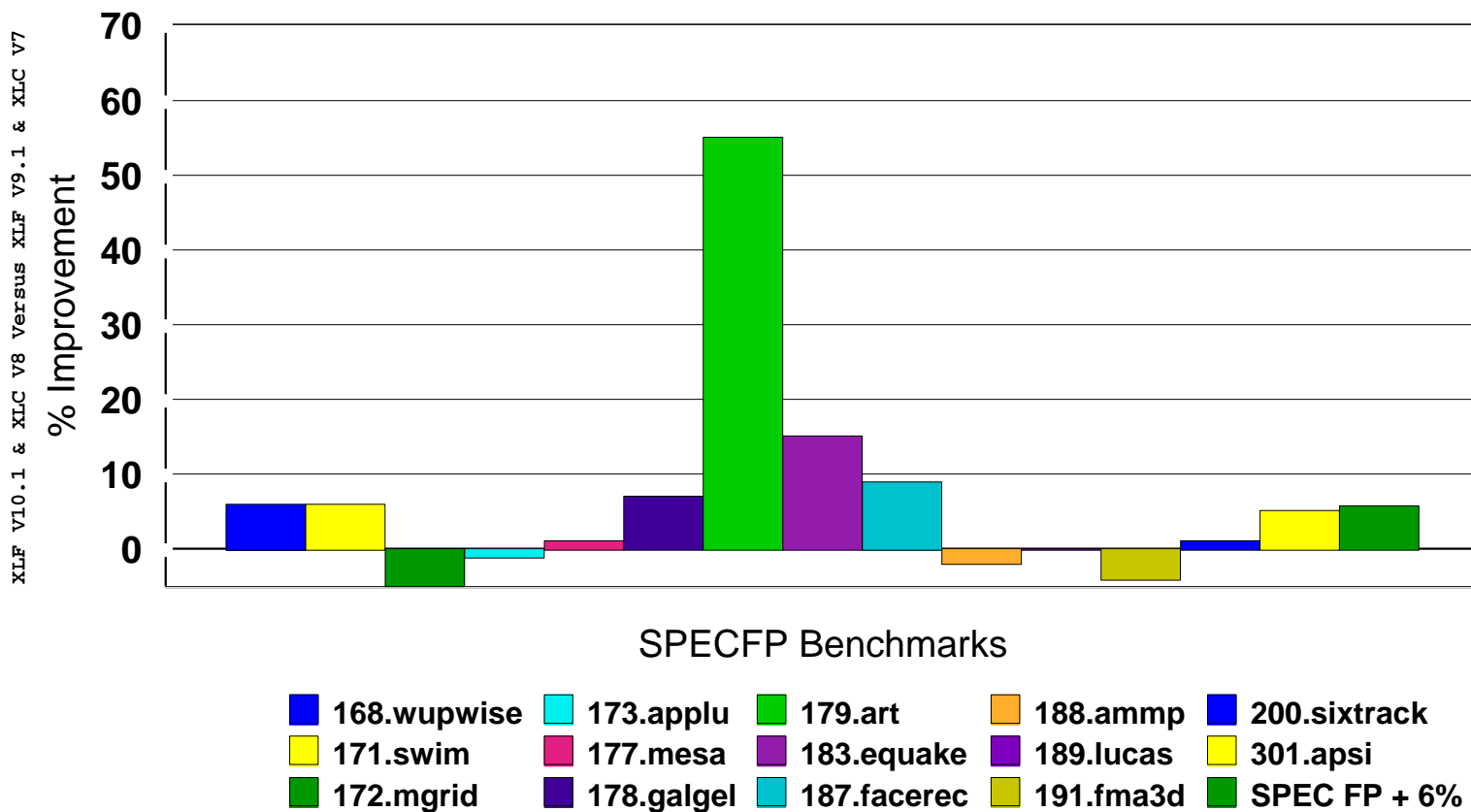


SPEC FP Base Improvements From Compiler On POWER5 XLF V9.1 and XLC V7.0



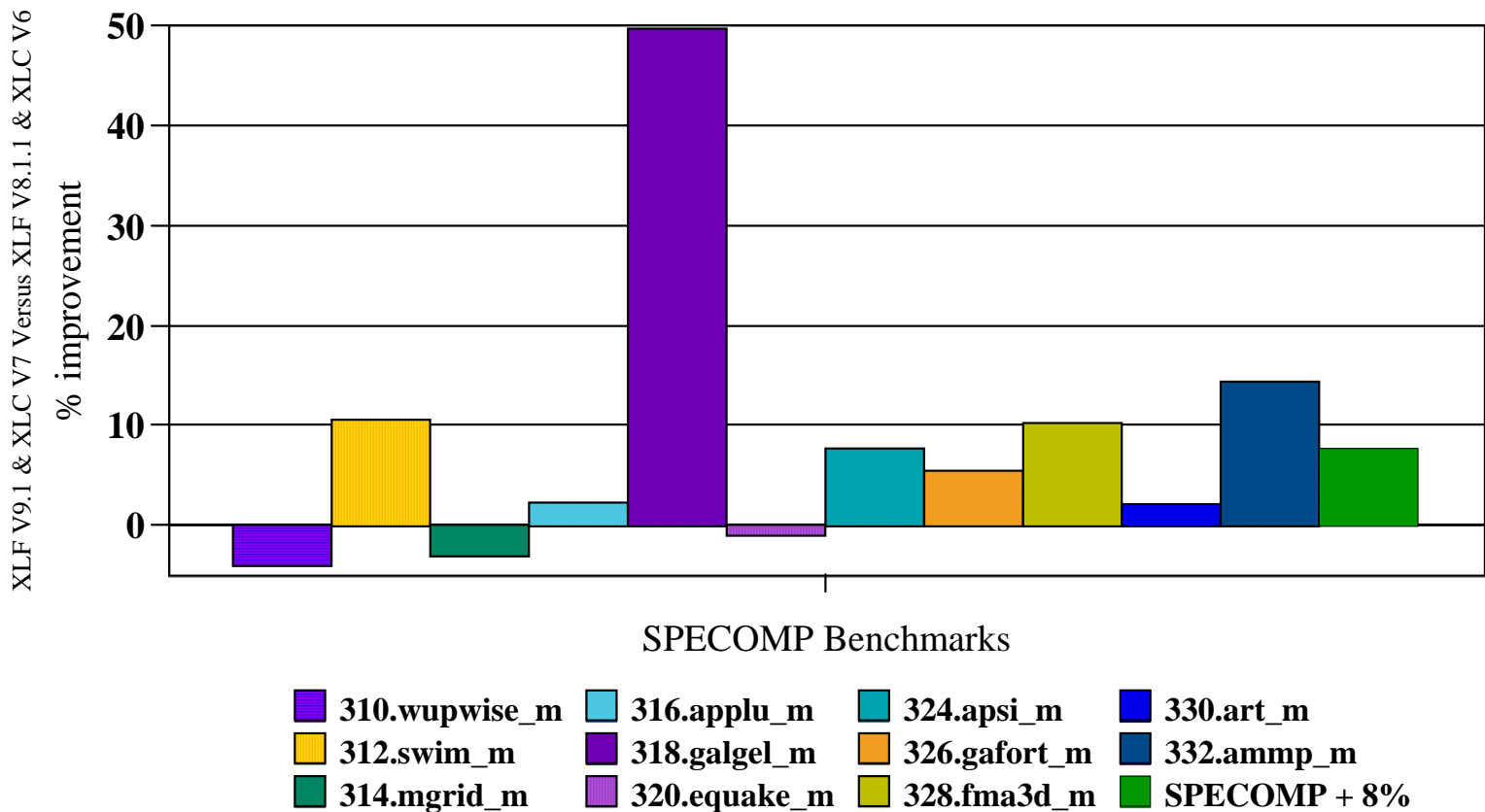


SPEC FP Base Improvements From Compiler On POWER5 XLF V10.1 and XLC V8.0



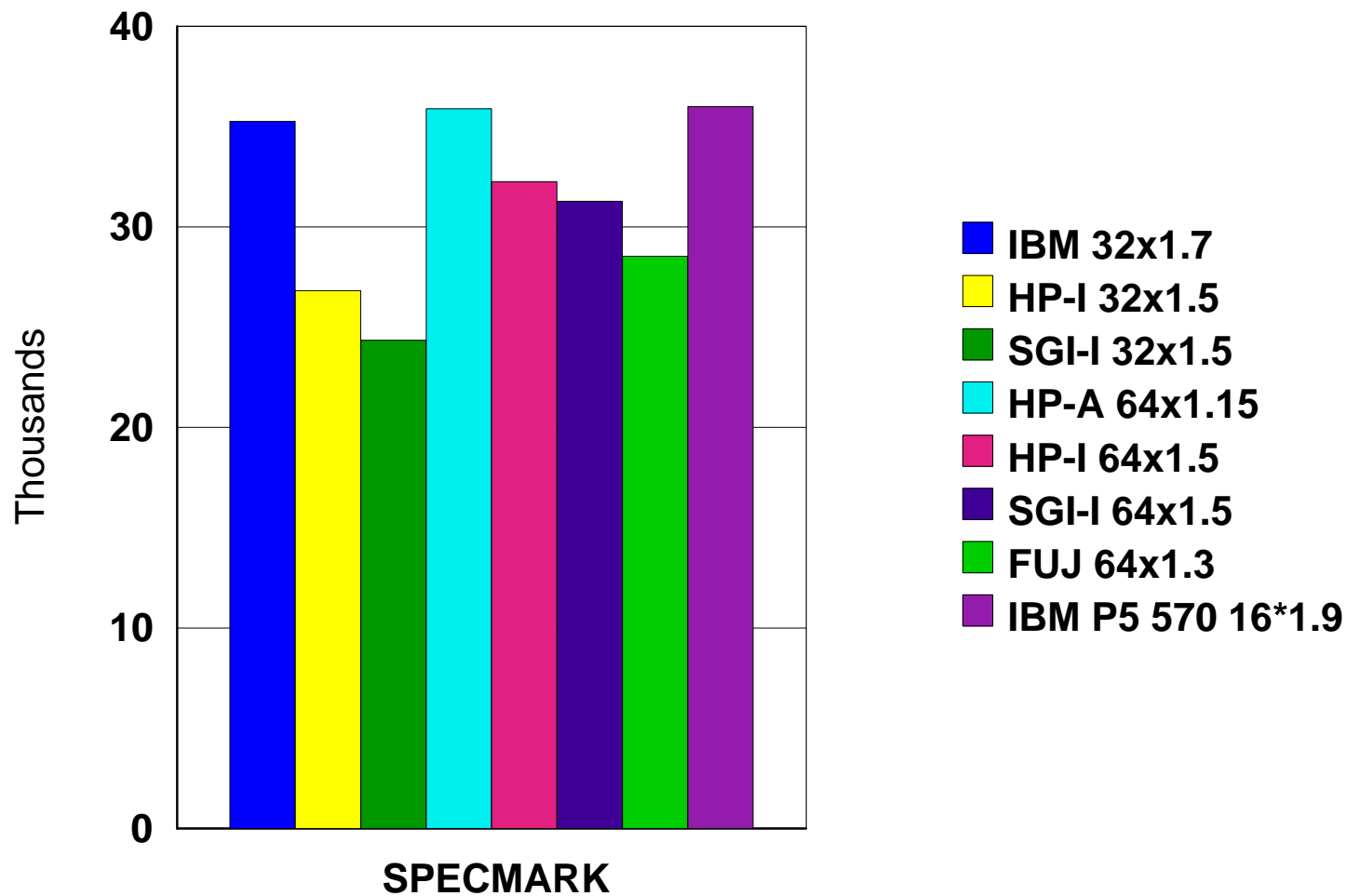


SPECOMP Base Improvements From Compiler On POWER4 (32-way) XLF V9.1 and XLC V7.0



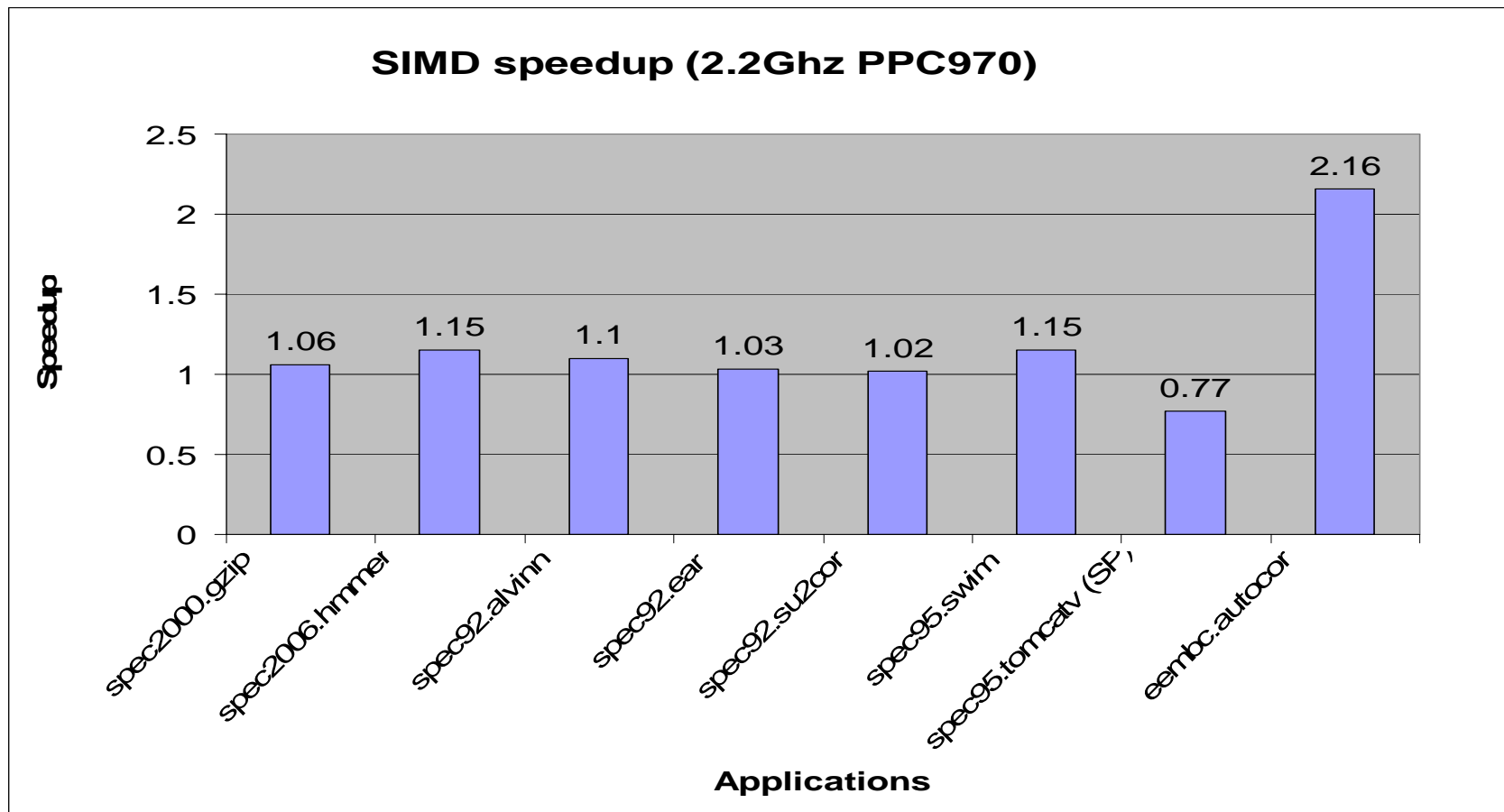


SPEC OMPM2001 Base Versus Competition



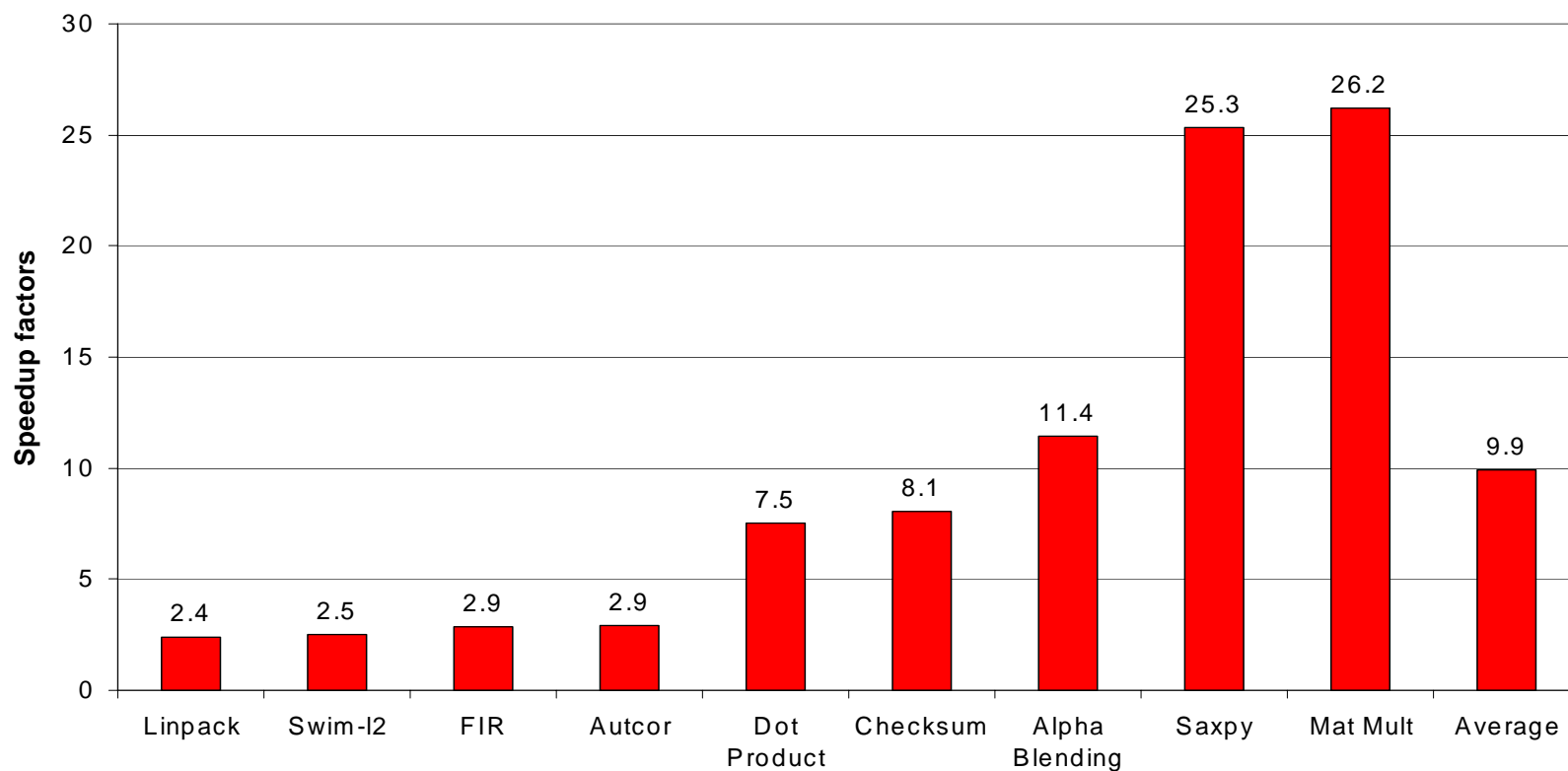


VMX PPC970 Results (V8.0/10.1)





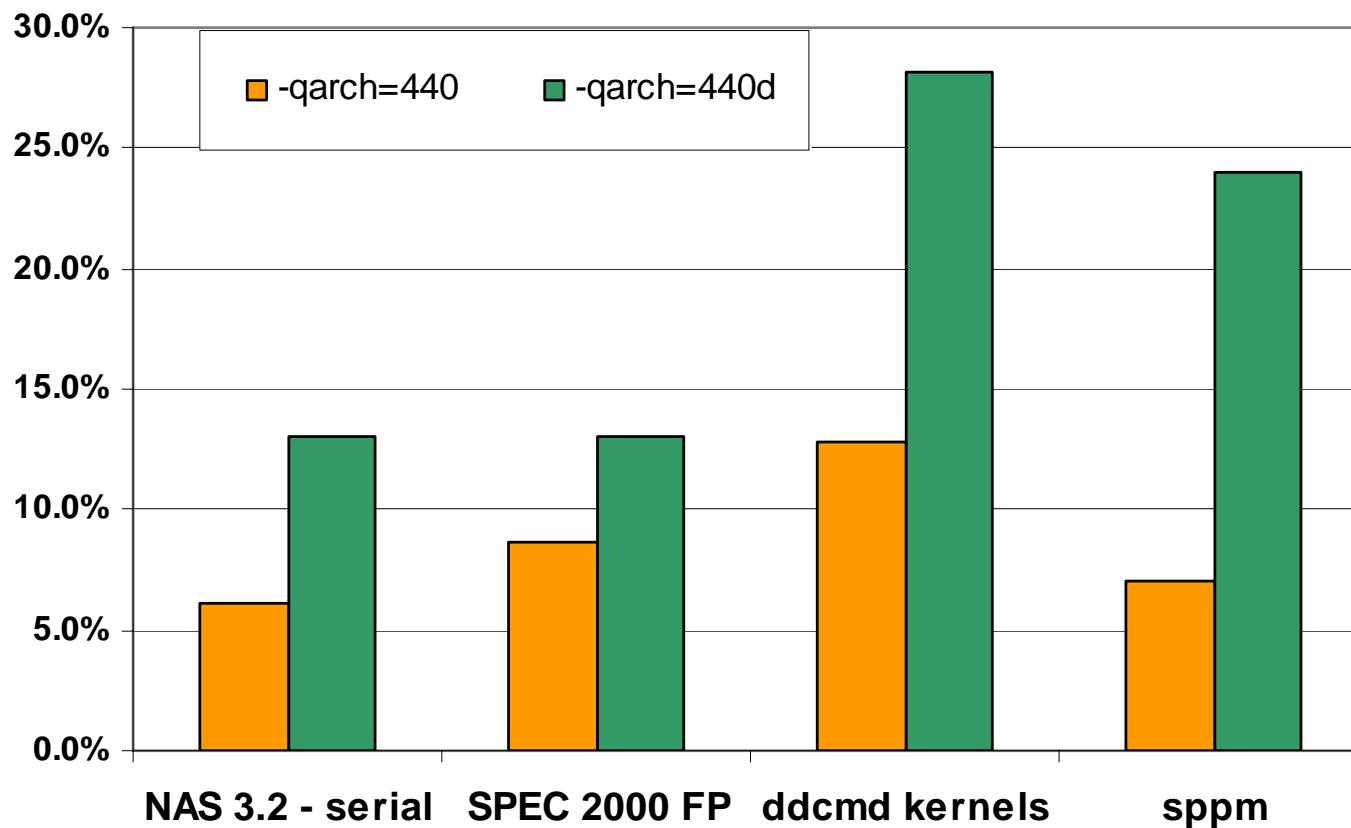
CELL Performance results



NOTE: speedup factor is SIMD versus SCALAR performance on SPE



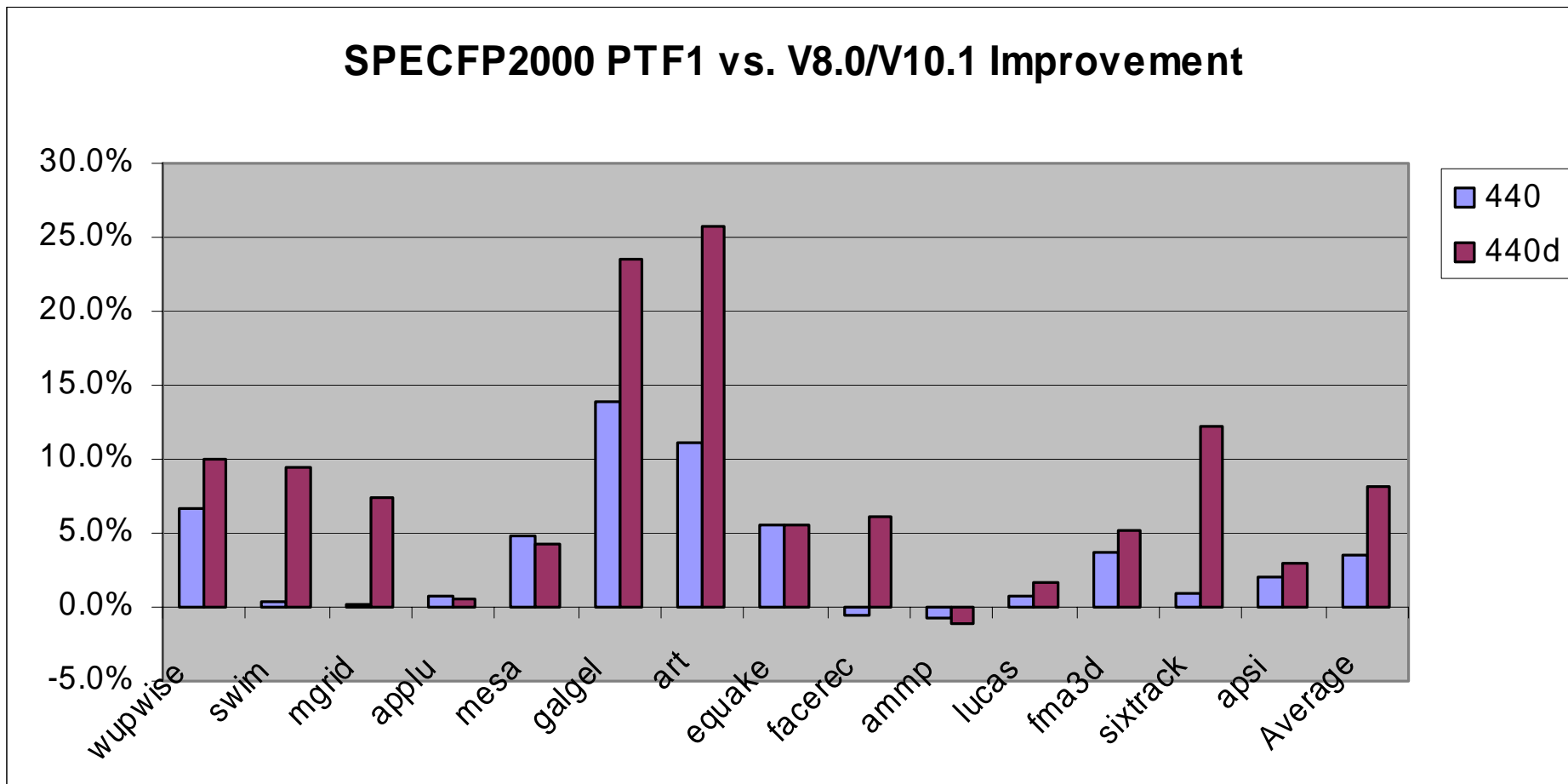
BG/L Improvements: -O5 V8.0/10.1 vs. V7.0/9.1





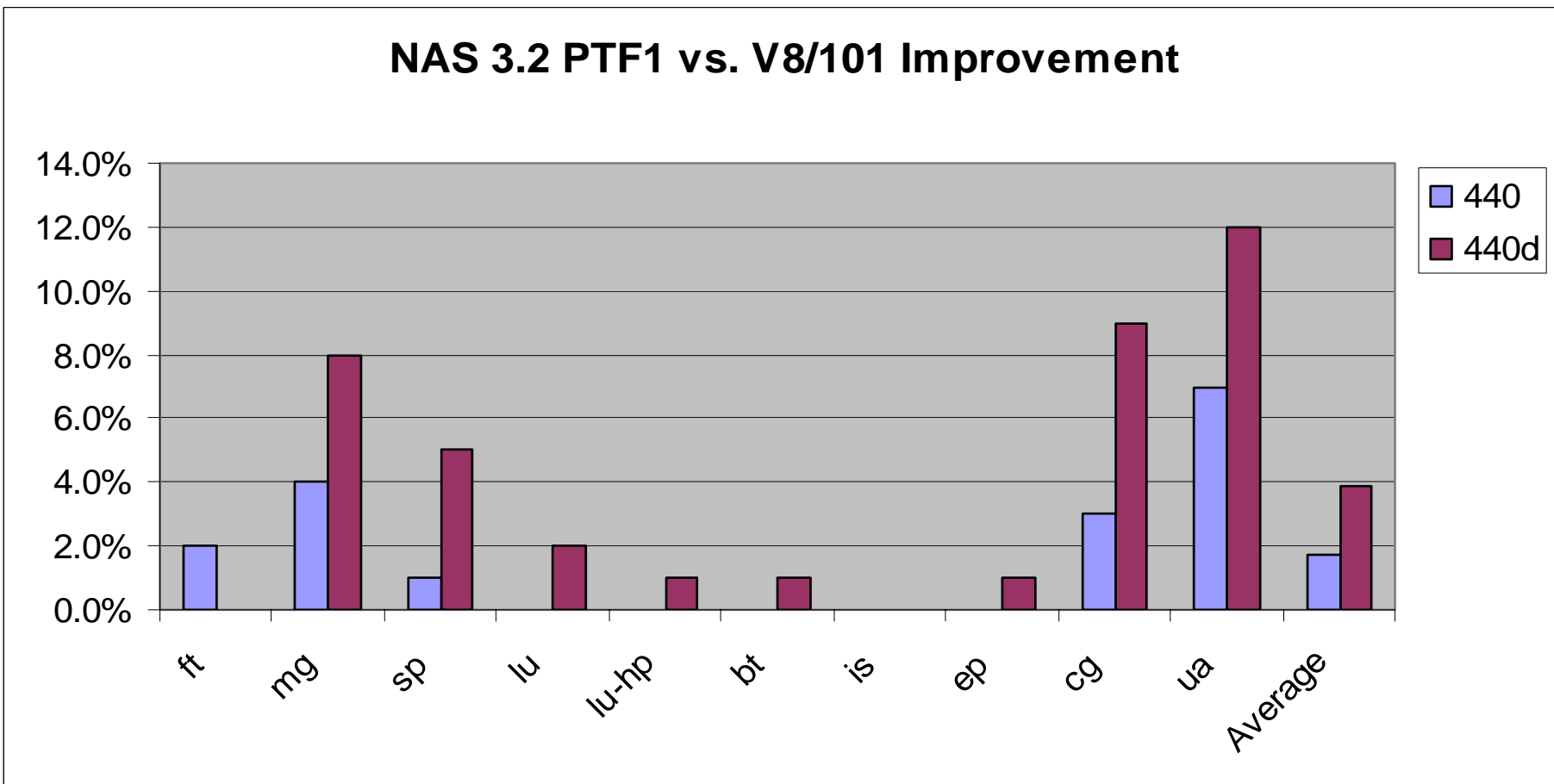
BG/L Improvements: SPEC FP2000 (V8.0/V10.1 PTF1)

SPEC FP2000 PTF1 vs. V8.0/V10.1 Improvement





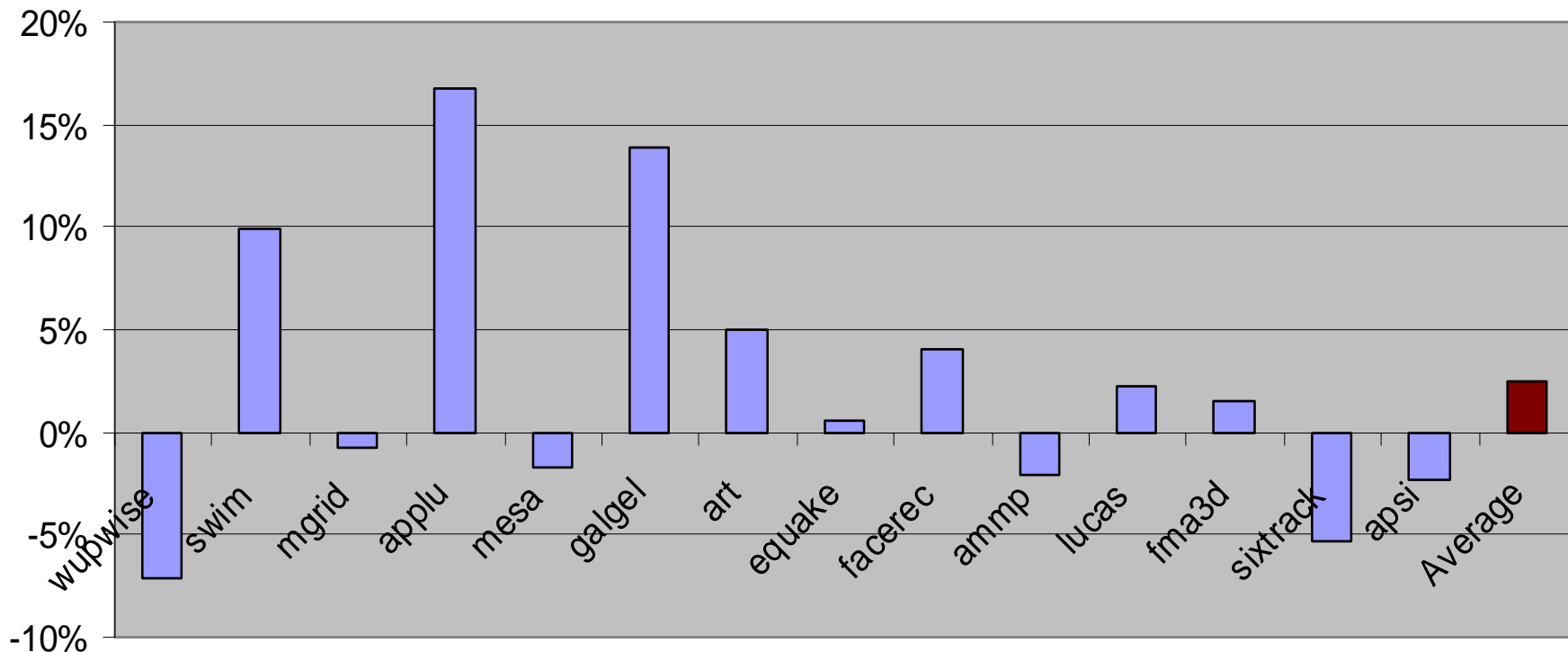
BG/L Improvements: NAS Serial 3.2 (V8.0/V10.1 PTF1)





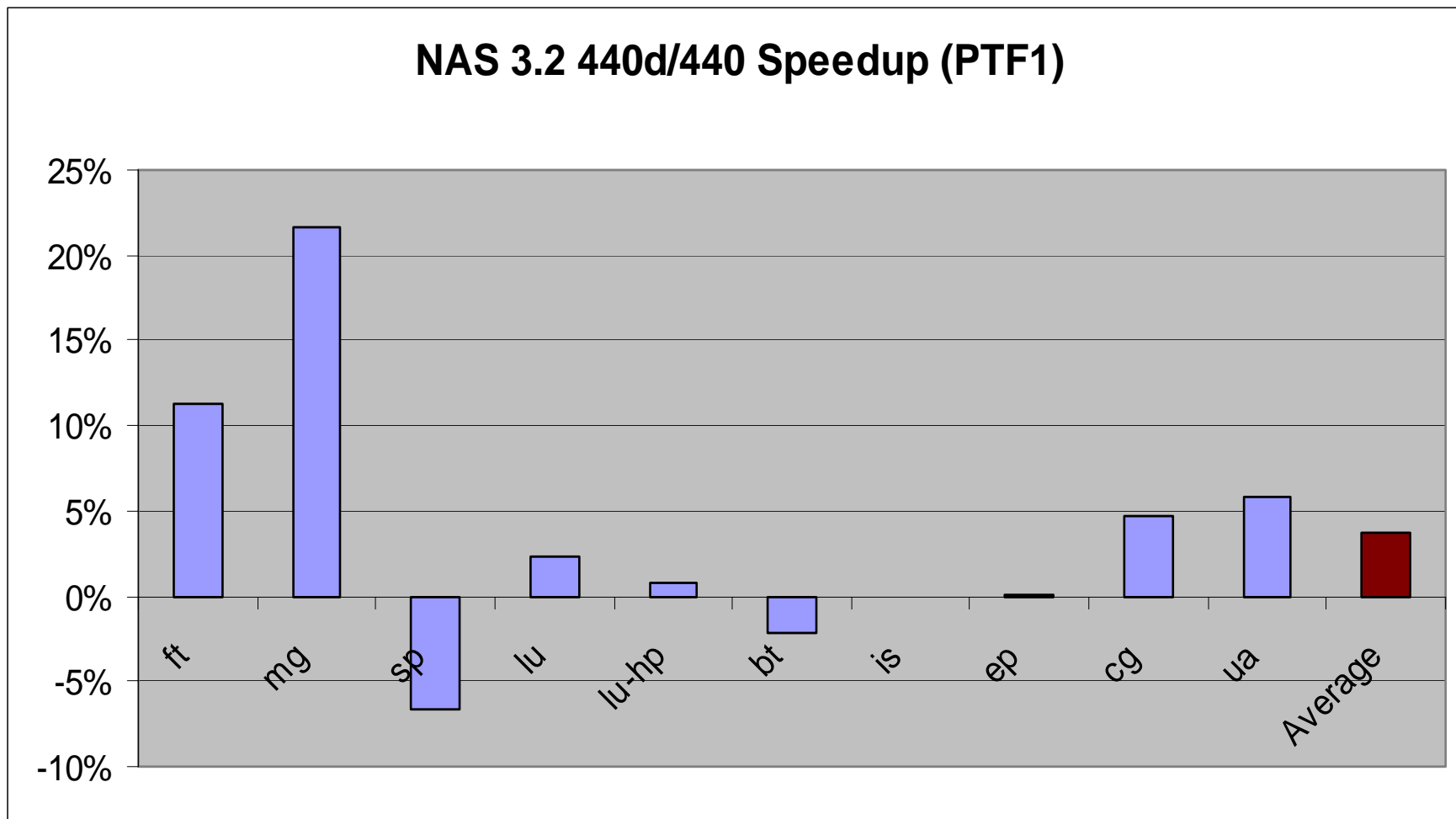
BG/L Improvements: SPEC FP2000 (V8.0/10.1 PTF1)

SPEC FP2000 440d/440 Speedup (PTF1)





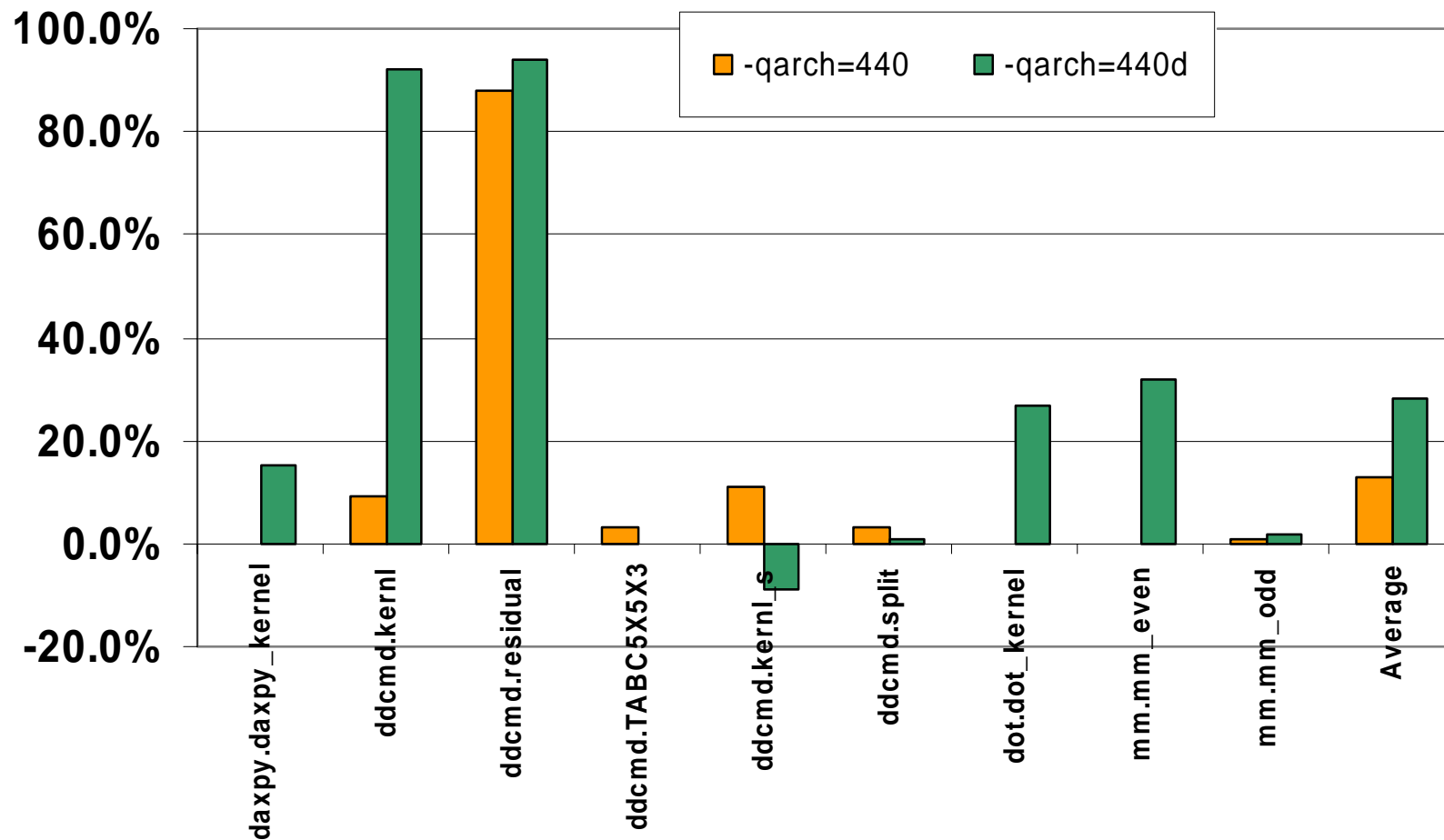
BG/L Improvements: NAS Serial 3.2 (V8.0/10.1 PTF1)





BG/L Improvements: -O5 V8.0/10.1 vs. V7.0/9.1

ddcmd uKernels Improvement V8/10.1 vs. V7/9.1 (-O5)





BACKUP SLIDES



The pSeries Compiler Products: Previous Versions

- All POWER4 enabled

- VisualAge C++ Version 6.0 for AIX

- VisualAge C++ Version 6.0 for Linux on pSeries

- C for AIX V6.0

- XL Fortran Version 8.1.1 for AIX

- XL Fortran Version 8.1.1 for Linux on pSeries

- XL C/C++ Enterprise Edition V7.0 for AIX (POWER5, PPC970)

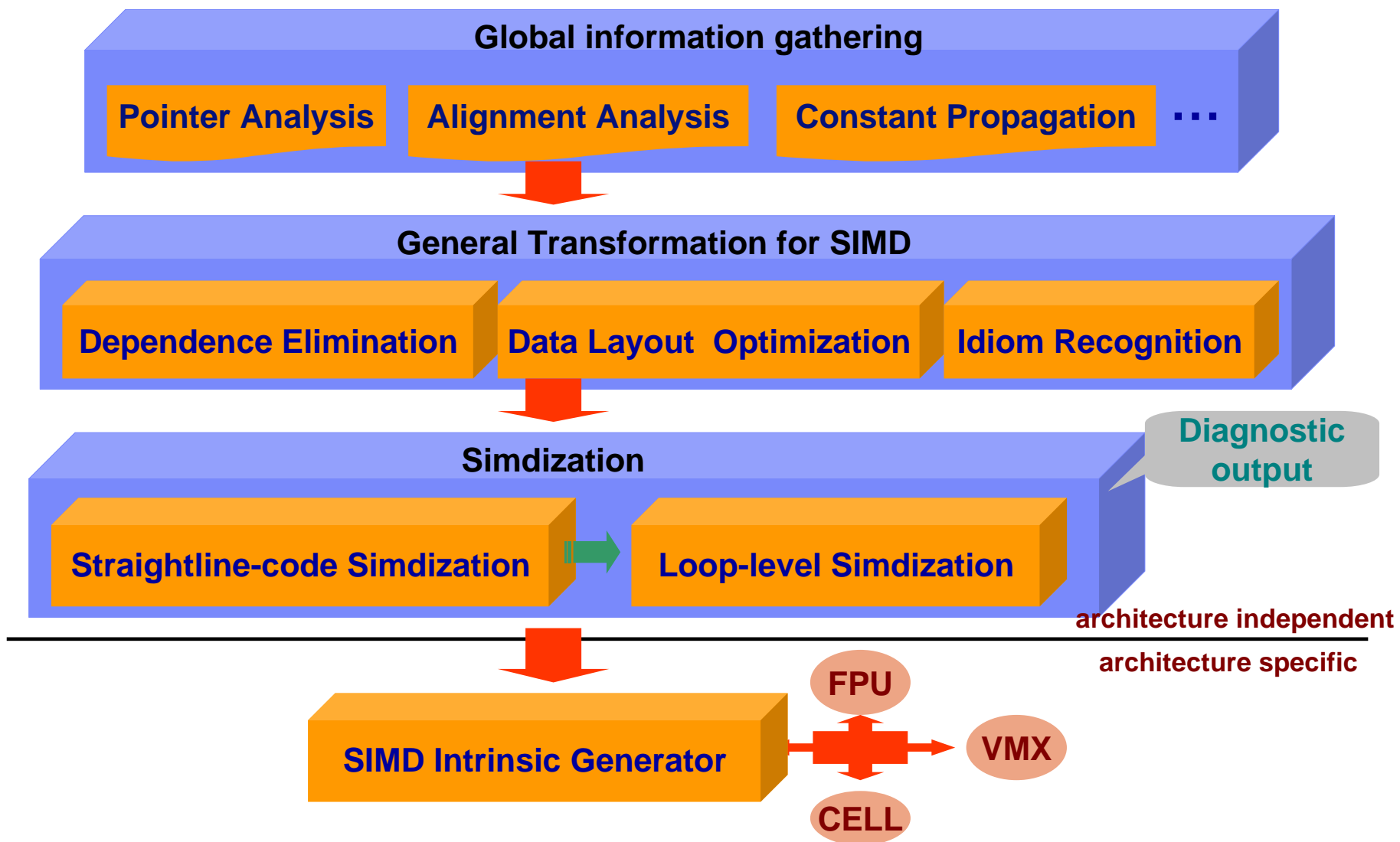
- XL Fortran Enterprise Edition V9.1 for AIX (POWER5, PPC970)

- XL C/C++ Advanced Edition V7.0 for Linux (POWER5, PPC970, PPC440)

- XL Fortran Advanced Edition V9.1 for Linux (POWER5, PPC970, PPC440)



A Unified Simdization Framework





Performance Improvements Delivered In 2004

- **Included in XLF V9.1 and XL C/C++ V7 on all platforms (AIX, Linux)**
- **POWER5**
 - Modified scheduling machine model
 - Usage of improved prefetch facilities
 - Usage of new instructions
- **PPC970**
 - Automatic generation of VMX code on Linux (SIMD vectorization)
 - Interprocedural pointer alignment propagation
- **OpenMP**
 - Tuned support for 64-way SMP
 - Continued improvements in overhead reduction
- **Intrinsic functions (Fortran Only)**
 - MATMUL, TRANSFER, INDEX, TRANSPOSE



Performance Improvements Delivered In 2004

- **Included in XLF V9.1 and XL C/C++ V7 on all platforms (AIX, Linux)**
- **Tuning assists :**
 - BLOCK_LOOP and LOOPID directives to specify which set of loops to tile, interchange or strip-mine
 - NOVECTOR and NOSIMD directive to tell compiler not to vectorize or simdize loop
 - Builtin functions for generating software divides (full double precision on POWER5)
 - Thread binding (set via XLSMPOPTS startproc and stride env variable)
 - Environment variable (set via XLSMPOPTS intrinths env variable) to control number of threads used by MATMUL and RANDOM_NUMBER (added in XLF V8.1.1 PTF)
 - View and manipulate information gathered by profile directed feedback (-qpdf1/-qpdf2) via showpdf and mergepdf tools
 - Prefetch directives for new stream prefetch control on POWER5



Performance Improvements Delivered In 2004

- **Included in XLF V9.1 and XL C/C++ V7 on all platforms (AIX, Linux)**
- **Loop Optimizations :**
 - Modulo scheduling of loops which contain branches
 - Further improvements to loop fusion for data reuse (e.g. loop alignment)
 - Perform vectorization on all platforms (including Linux)
 - Enhancement of vectorization (additional functions, loop versioning, vector merging)
 - Tiling for BLAS-like and streaming loop nests
 - Predictive Commoning (common subexpression elimination across loop iterations)
 - Improved data dependence analysis
 - Automatic generation of software divides on POWER5
 - Automatic generation of new stream prefetch instructions on POWER5



Performance Improvements Delivered In 2004

- **Included in XLF V9.1 and XL C/C++ V7 on all platforms (AIX, Linux)**
- **Other Optimizations :**
 - Interprocedural Strength reduction
 - Interprocedural Register Allocation
 - Split array of structures into multiple arrays for better exploitation of hardware streams and smaller d-cache footprint
 - Use Profile Directed Feedback (PDF) information to:
 - Specialize calls to malloc/calloc to use pools of small objects
 - Specialize memcpy and memset with small lengths
 - Specialize integer divide and modulo
 - Superblock formation for better instruction scheduling



Performance Improvements Delivered In 2005

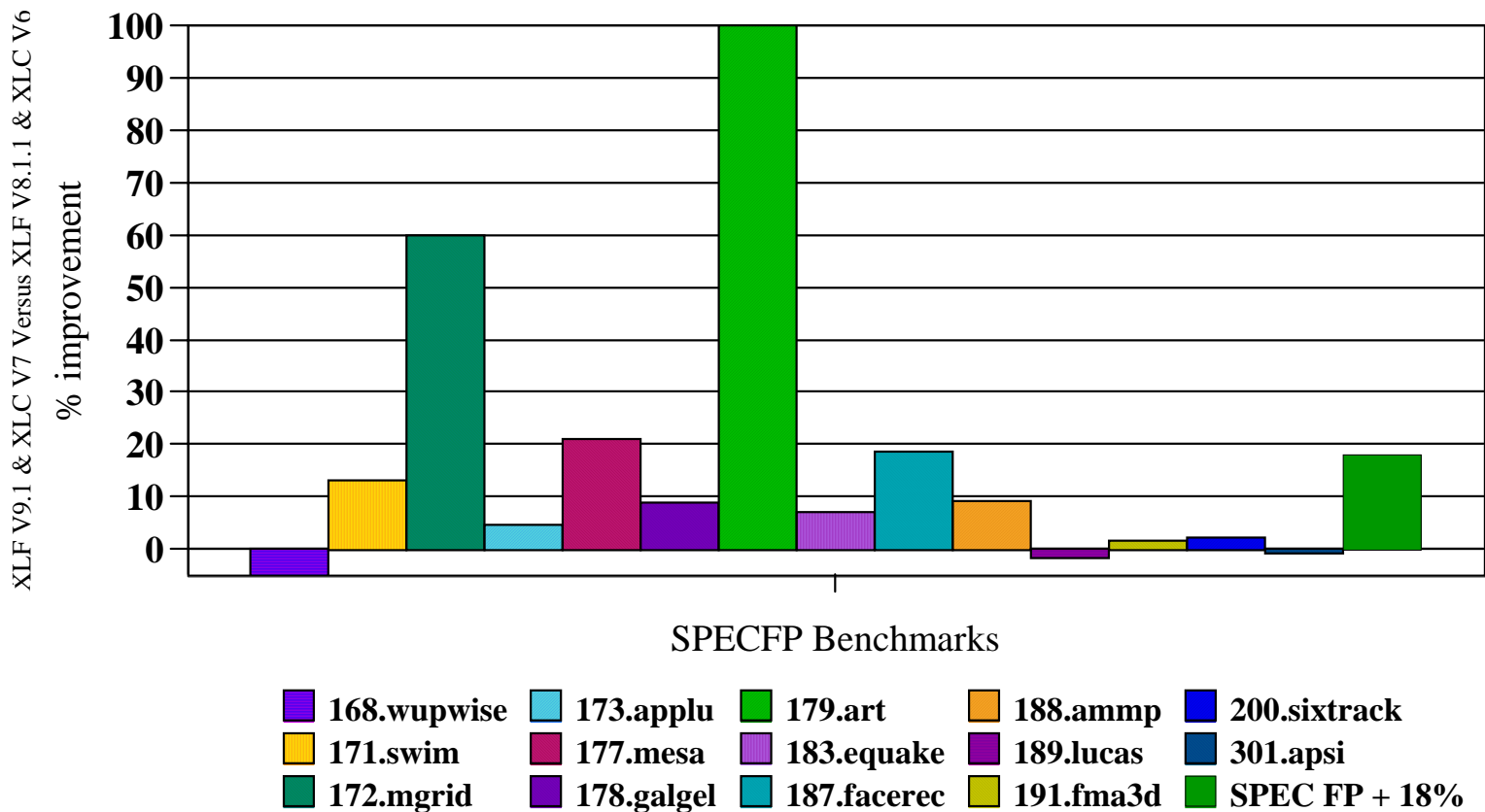
- **Included in XLF V10.1 and XL C/C++ V8.0 on all platforms (AIX, Linux)**

- **Loop Optimizations:**
 - Sparse Vectorization
 - Runtime dependence testing (loop versioning)
 - Insert data cache touch instruction for strided memory access
 - Array data flow analysis for array privatization
 - Improved automatic parallelization (lower barrier overhead, control number of threads per region, multi-dimensional reductions)

- **Other Optimizations:**
 - Outline “cold” fields of data structures for smaller d-cache footprint (using pdf)

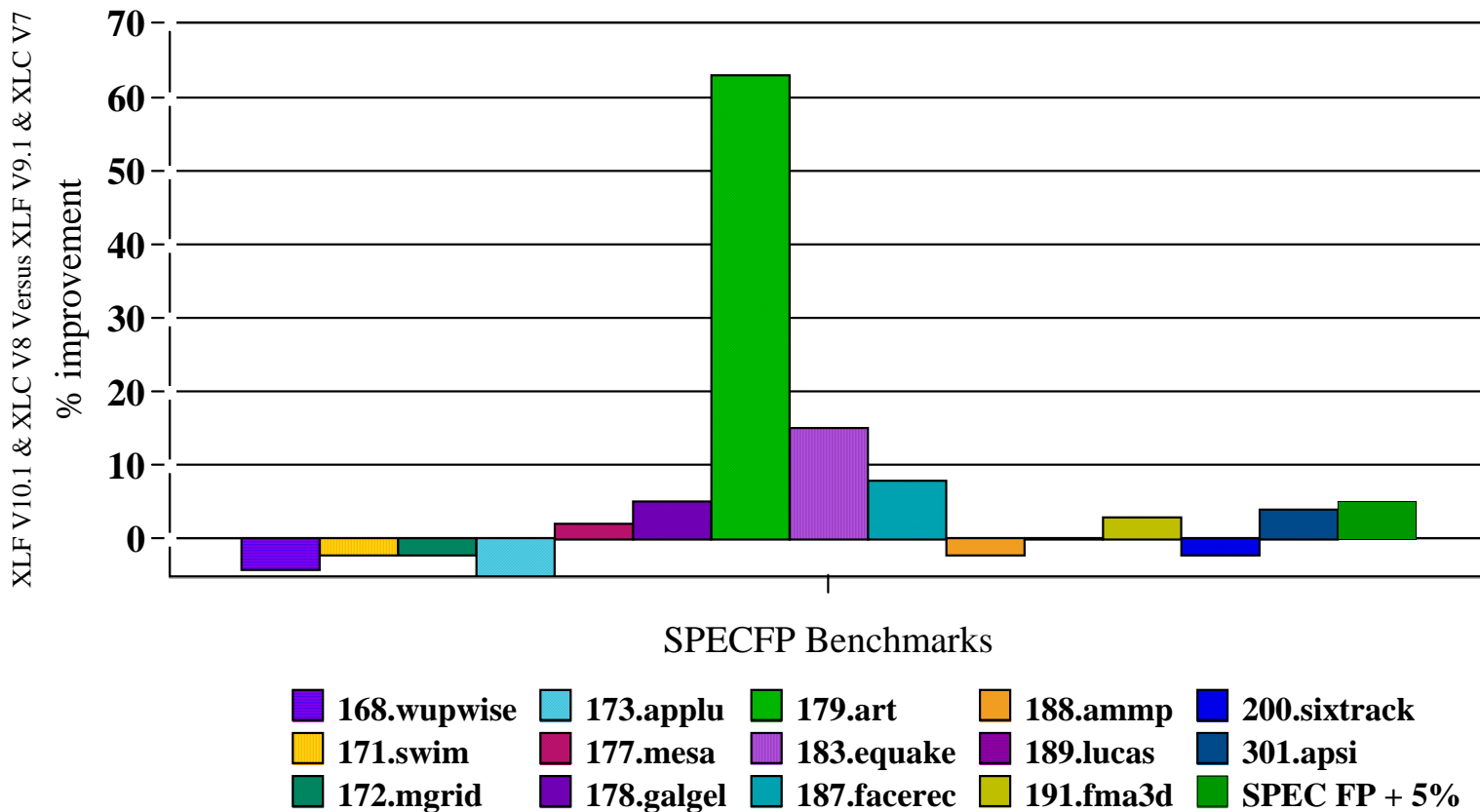


SPEC FP Base Improvements From Compiler On POWER4 XLF V9.1 and XLC V7.0



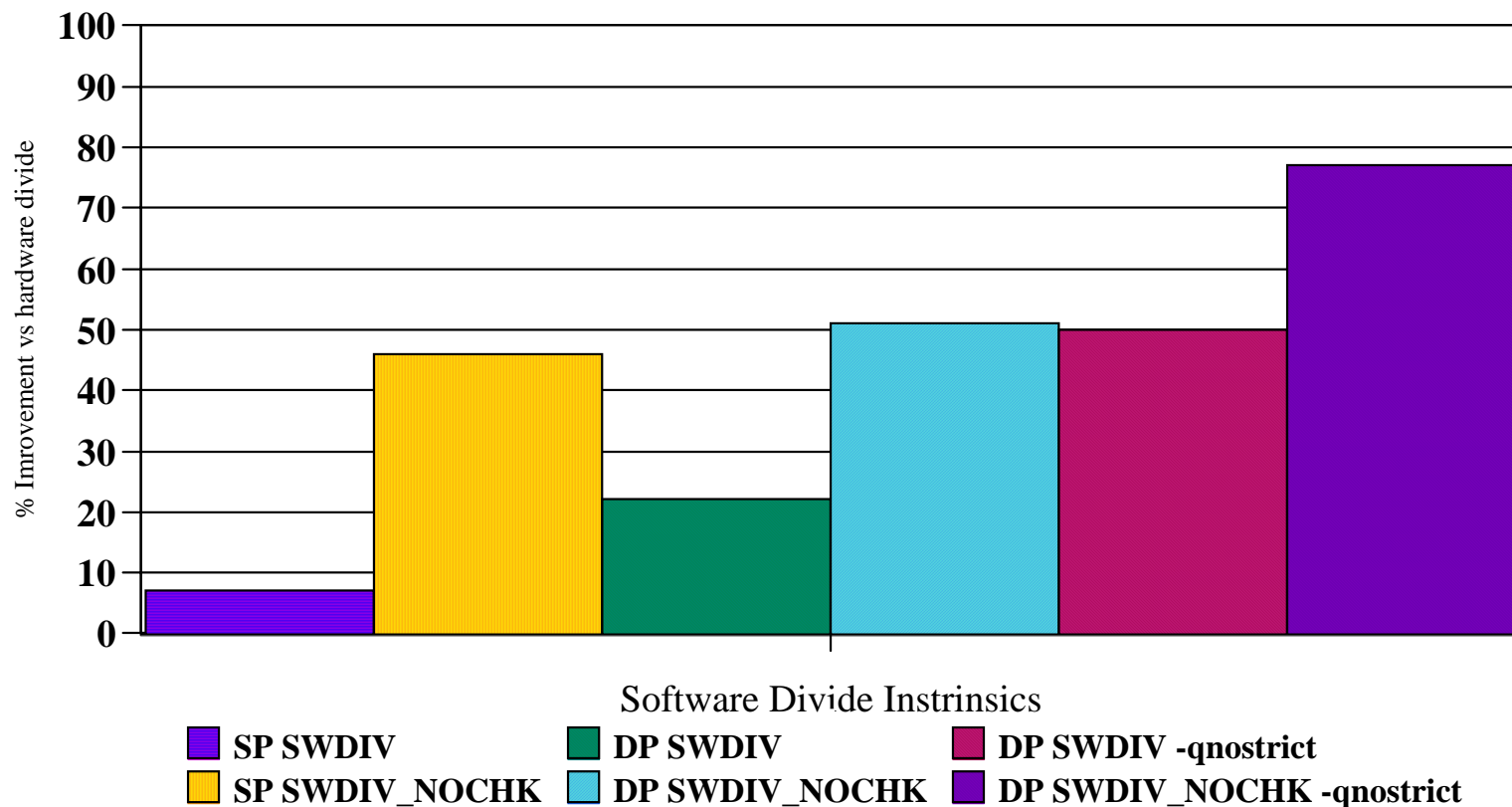


SPEC FP Base Improvements From Compiler On POWER4 XLF V10.1 and XLC V8.0





Software Divide Improvements On POWER5





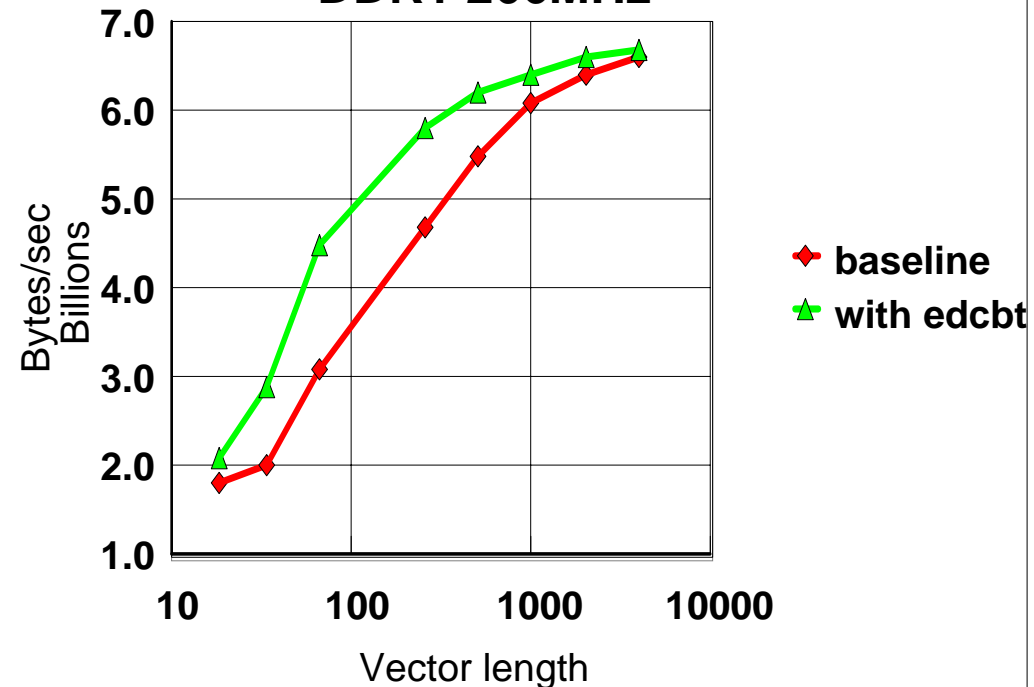
Performance With New Prefetch Directives On POWER5

```

do k=1,m
  lcount = nopt2
  do j=ndim2,1,-1
!!IBM PROTECTED_STREAM_SET_FORWARD(x(1,j),0)
!!IBM PROTECTED_STREAM_COUNT(lcount,0)
!!IBM PROTECTED_STREAM_SET_FORWARD(a(1,j),1)
!!IBM PROTECTED_STREAM_COUNT(lcount,1)
!!IBM PROTECTED_STREAM_SET_FORWARD(b(1,j),2)
!!IBM PROTECTED_STREAM_COUNT(lcount,2)
!!IBM PROTECTED_STREAM_SET_FORWARD(c(1,j),3)
!!IBM PROTECTED_STREAM_COUNT(lcount,3)
!!IBM EIEIO
!!IBM PROTECTED_STREAM_GO
  do i=1,n
    x(i,j)= x(i,j)+a(i,j)*b(i,j) + c(i,j)
  enddo
enddo
call dummy(x,n)
enddo

```

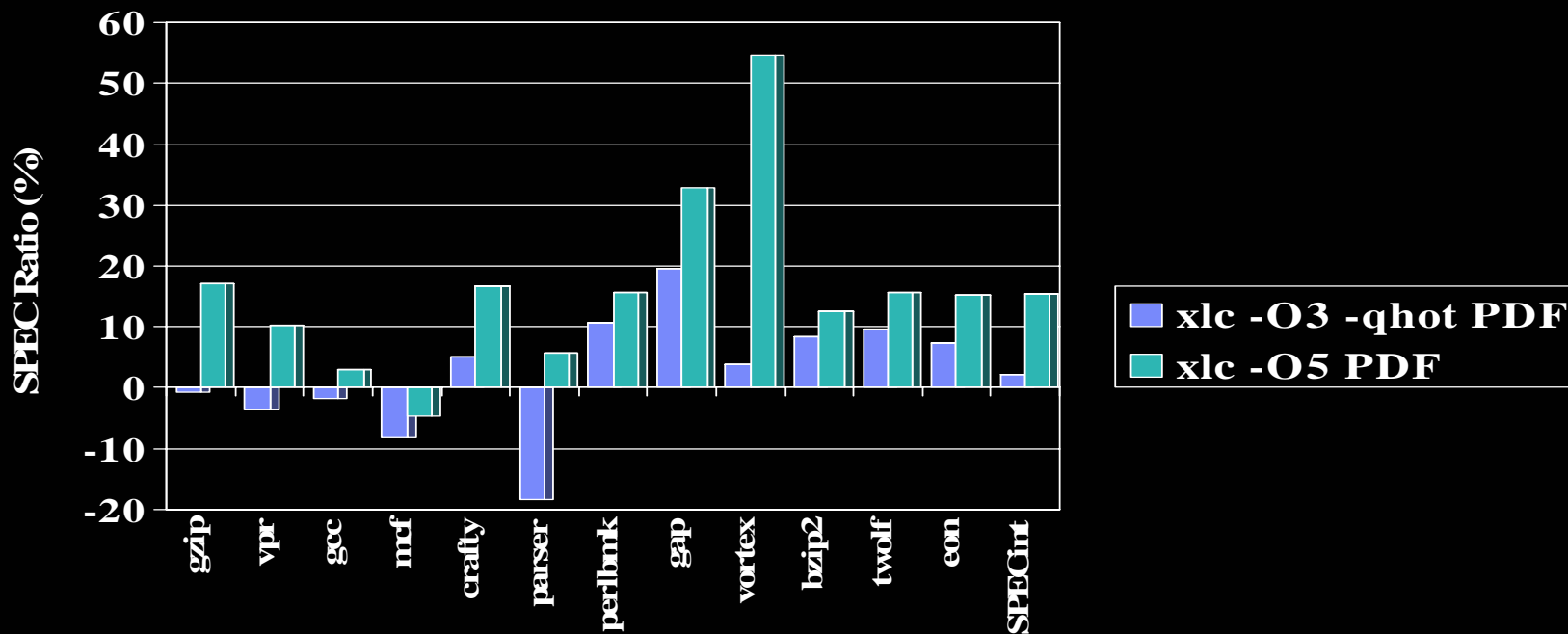
Four stream performance Power5 BUV 1-chip/4SMI 1.6GHz DDR1 266MHz





XL Compilers Vs. gcc high-opt Performance Comparison

SPEC2000int
p520 (SF2) 1.65GHz POWER5, SLES 9, gcc 4.0, xlc V7.0

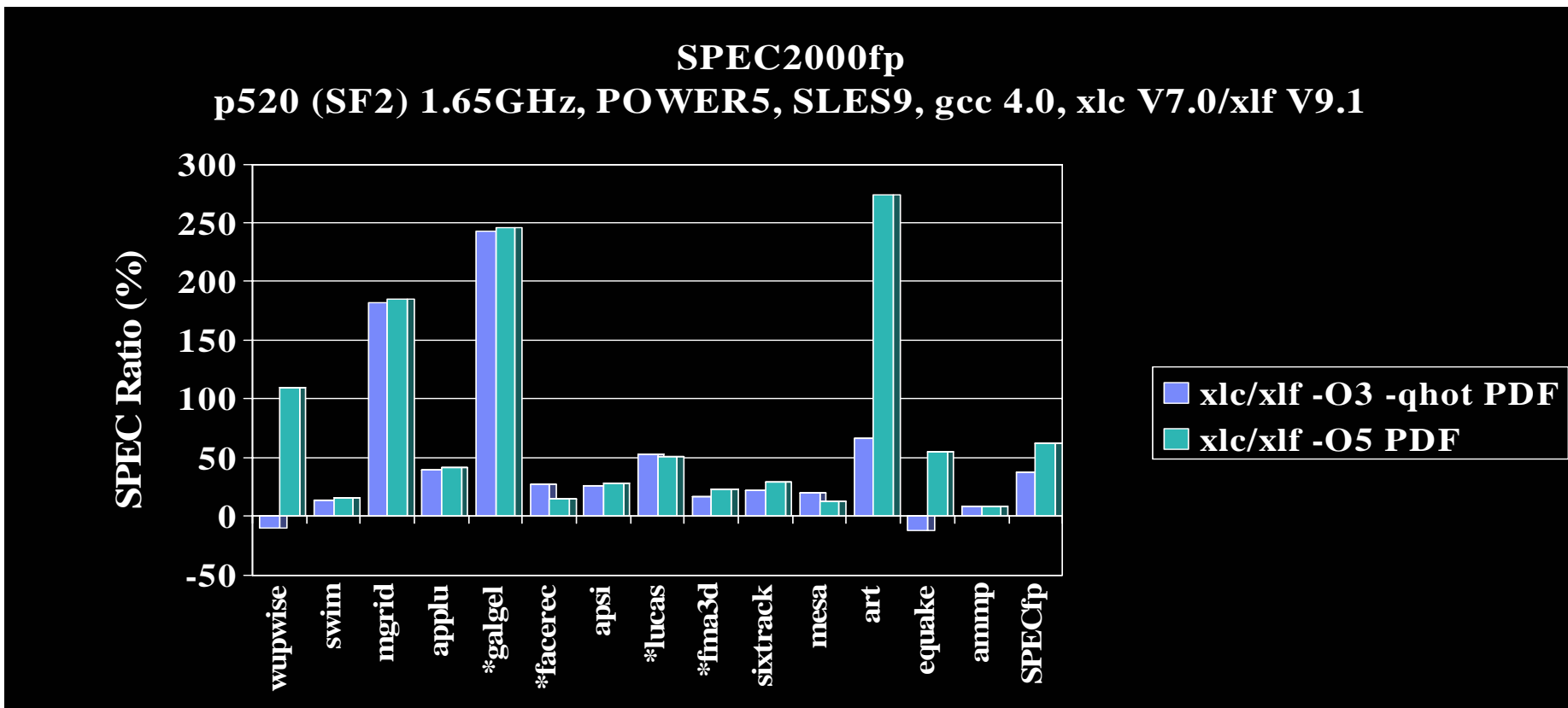


-qarch=pwr5 is used with XL C/C++ v7

-mtune=power5 -mpowerpc-gpopt -mpowerpc-gfxopt -ffast-math -funroll-loops -fpeel-loops -ftree-loop-linear -fprofile-generate/-fprofile-use is used with gcc



XL Compilers Vs. gcc high-opt Performance Comparison

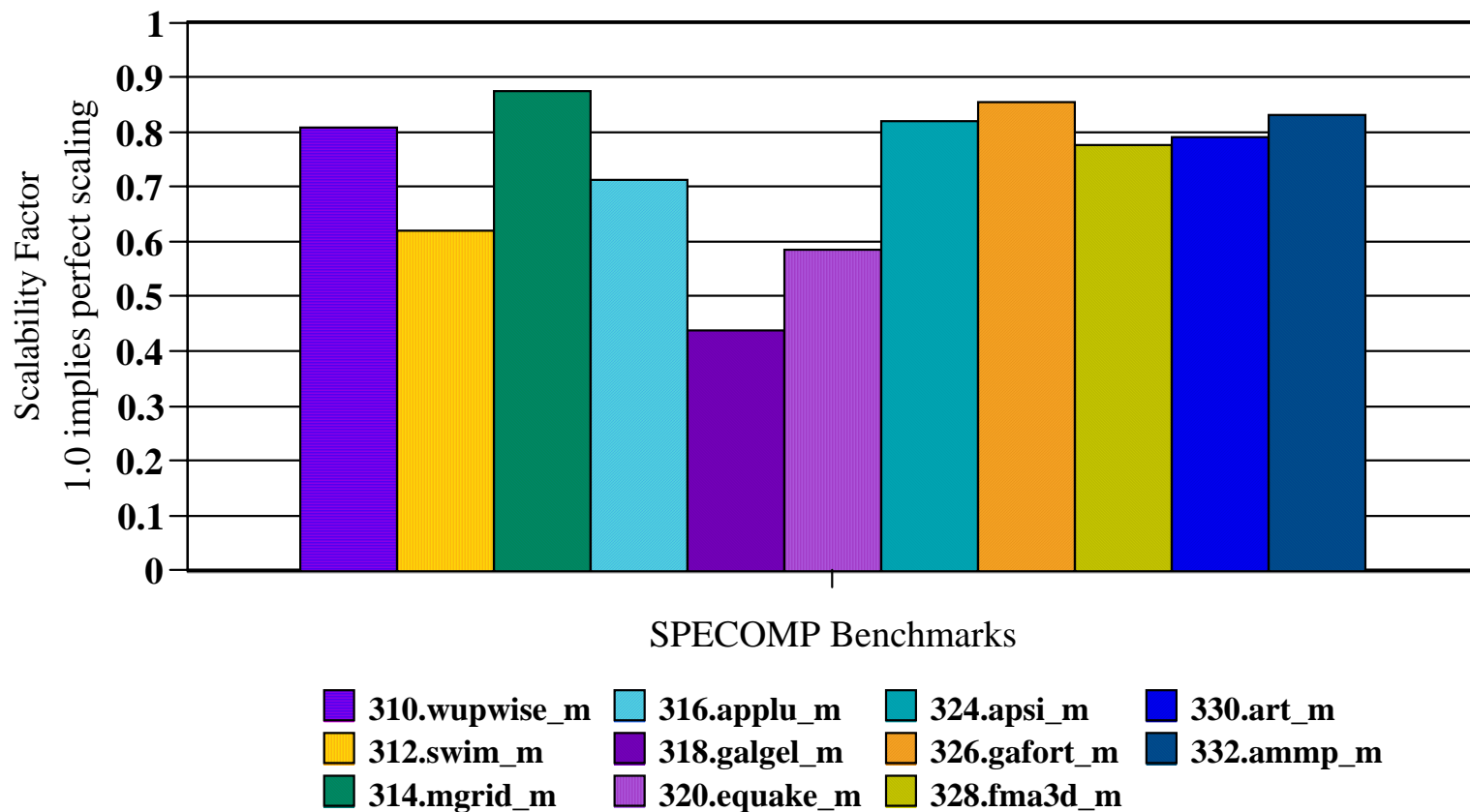


-qarch=pwr5 is used with XL C/C++ v7 and XL Fortran v9.1

-mtune=power5 -mpowerpc-gpopt -mpowerpc-gfxopt -ffast-math -funroll-loops -fpeel-loops -ftree-loop-linear
 -fprofile-generate/-fprofile-use is used with gcc v4

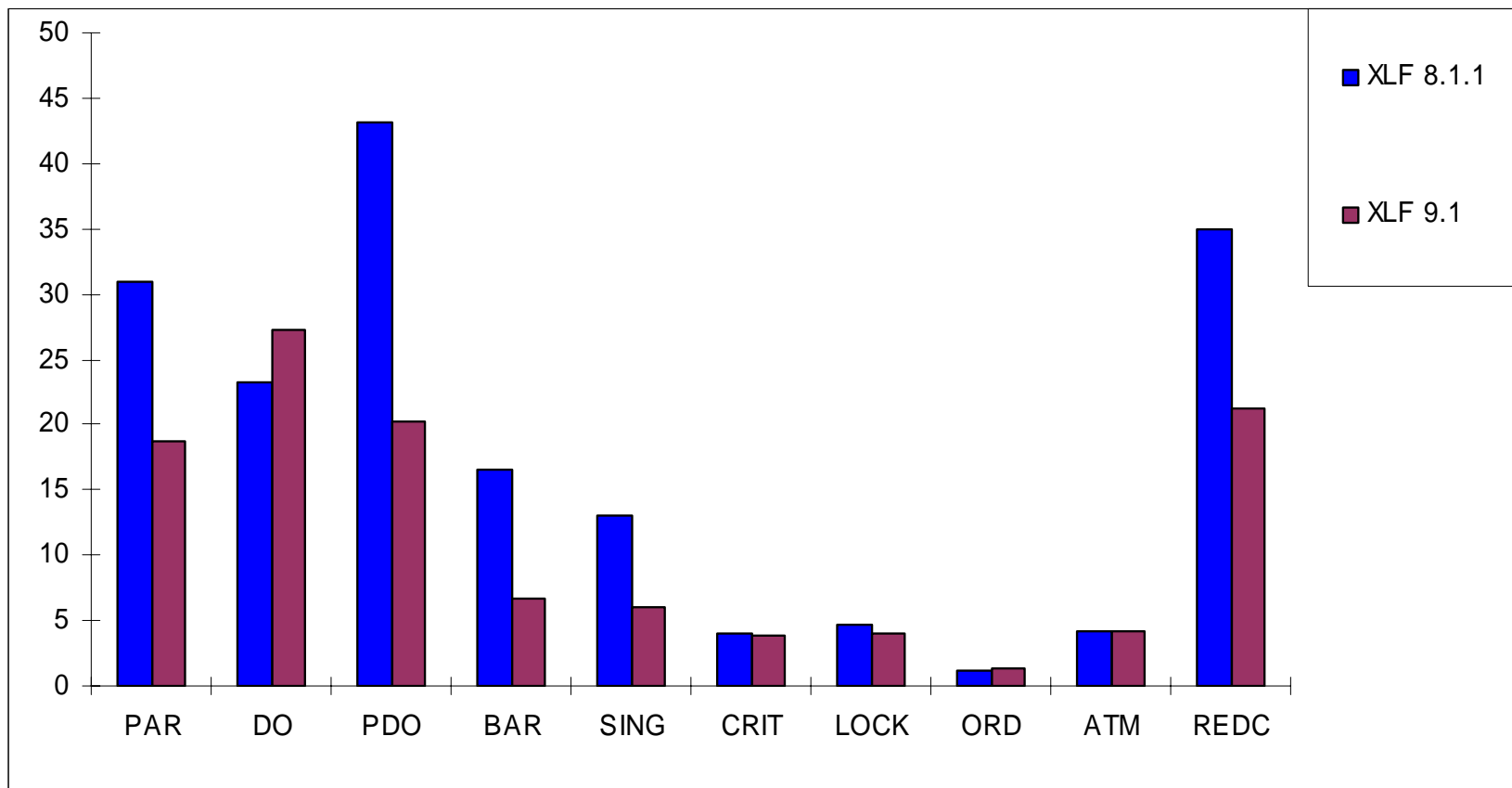


SPECOMP Scalability On POWER4 (16 vs 32 CPUs)





32-way EPCC results on AIX 5.2 p690 system (1.1 GHZ)



Time in micro-seconds - lower is better



SPEC FP Base Auto-Parallelization (2 CPUs, POWER4)

