Release Updates

Ray Chen
University of Maryland
rchen@cs.umd.edu

Madhavi Krishnan
University of Wisconsin
madhavi@cs.wisc.edu

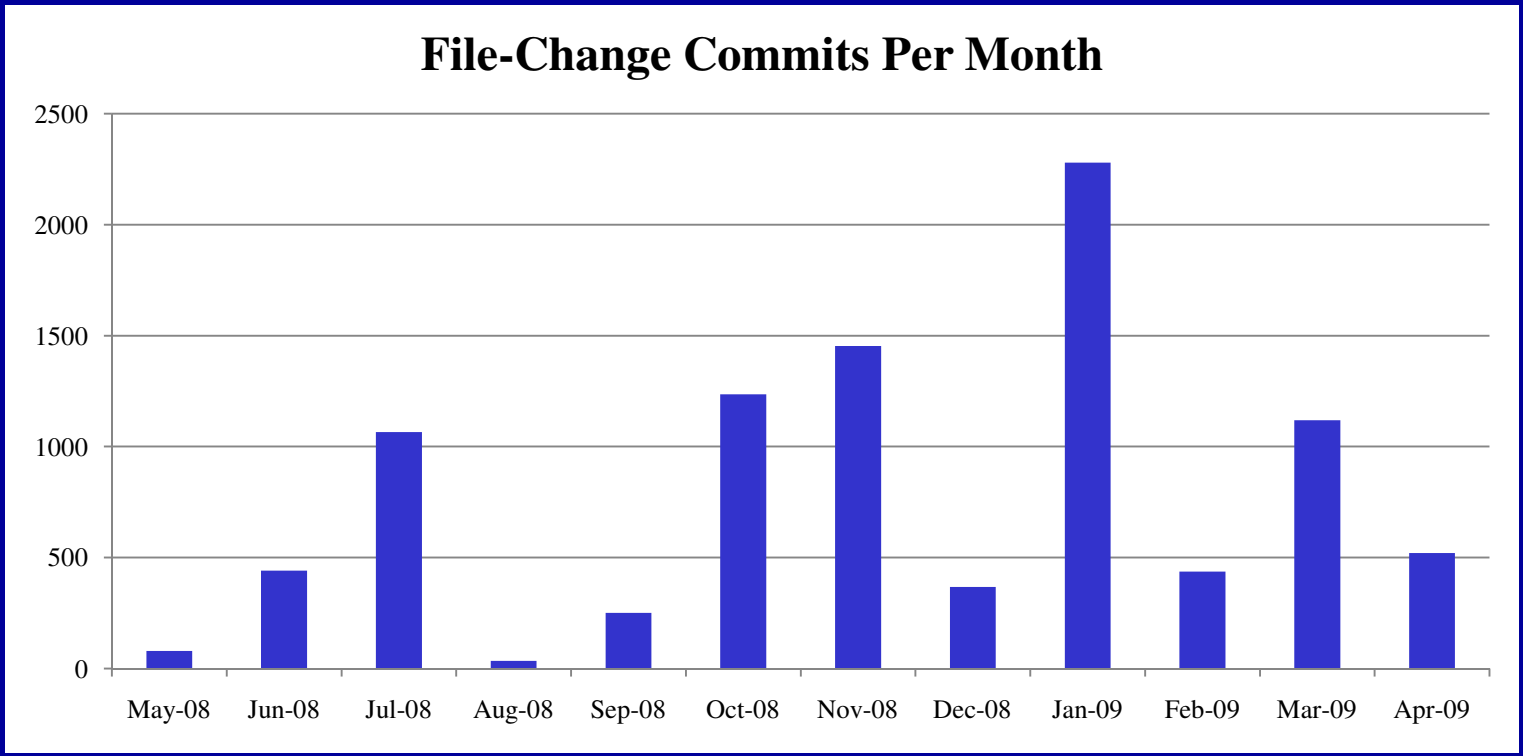
http://www.paradyn.org

Talk Outline

- Updates to Dyninst 6.0
  - Code analysis
  - Dynamic instrumentation
  - Binary rewriting
- Deconstructing Dyninst
  - SymtabAPI
  - StackwalkerAPI
  - InstructionAPI
  - DepGraphAPI
- Updates to MRNet 2.1
- Conclusion

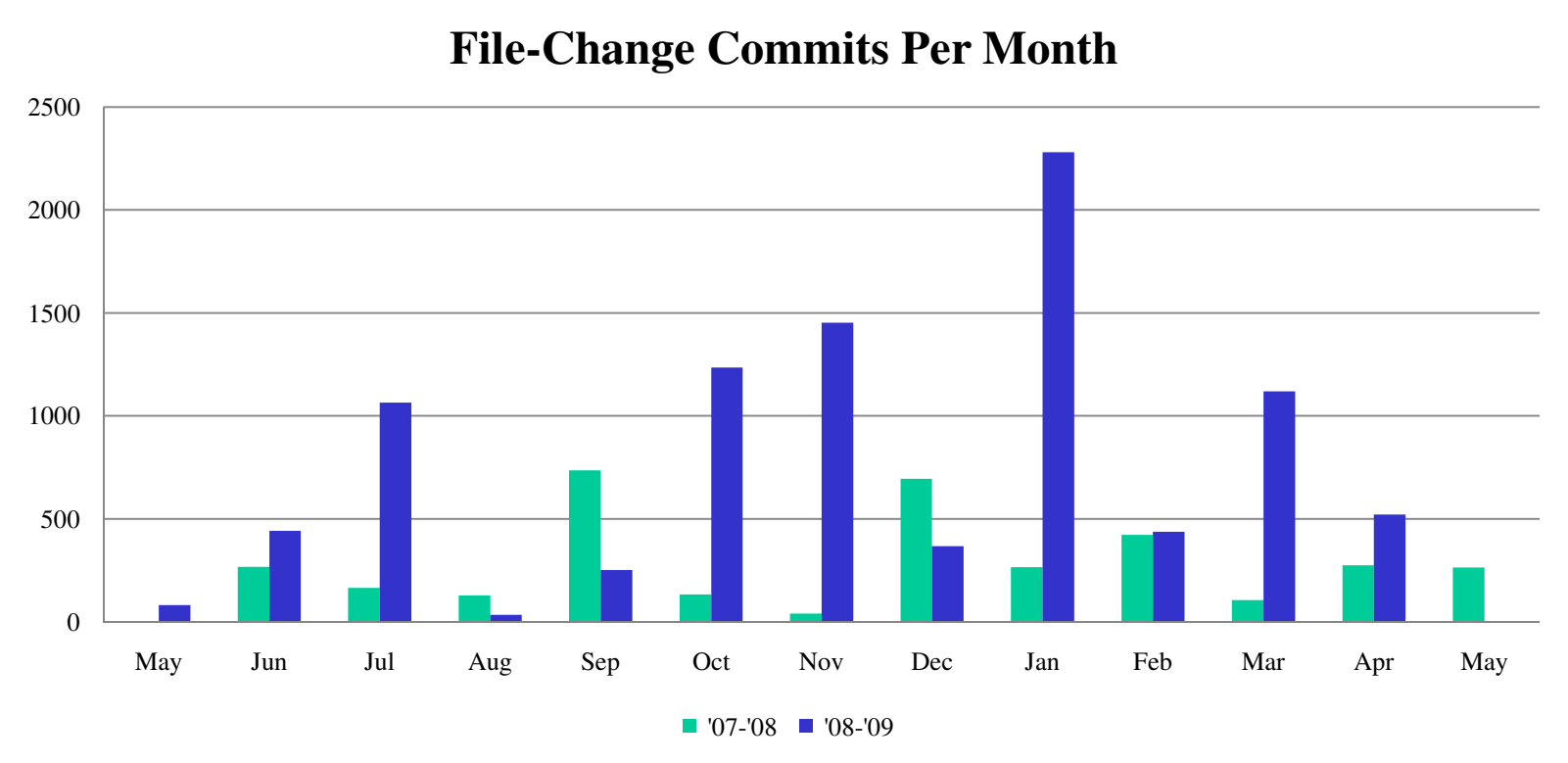
Introduction

- Upcoming Dyninst 6.0 Release
- Working hard over the past year



Introduction

- Upcoming Dyninst 6.0 Release
- Working hard over the past year



Project Updates

- New nightly test result web interface
  - Flexible infrastructure
  - Designed for 3rd party input
  - http://www.dyninst.org/testdb
- Migrated from CVS to Git
  - Chose Git over SVN
  - http://git.dyninst.org

Dyninst 6.0 Updates

- New platform support
  - Linux PPC64
- New compiler support
  - Microsoft VS 2008
- Overhauled test suite
  - More descriptive test names
  - Better sandboxing for problem isolation
- AutoConf assisted source builds

Dyninst 6.0 Updates

- BPatch_register construct
  - Change register values dynamically
  - Modify with BPatch_registerExpr
- Support for external Linux debug info
  - Functionality introduced in Fedora Core 9
- Function replacement on PPC64
  - Difficultly comes from arbitrary branch
    - Requires 28 bytes of code and 2 free registers
  - Basic block instrumentation was the key

Deprecated Features

- #define IBM_COMPAT
  - No longer support building libdyninstAPI.so with xlC
- Save the world
  - Obsoleted by Binary Rewriter

Code Analysis

- Improved slicing internals
  - More accurate information returned
  - Backed by InstructionAPI
- Liveness optimization
  - Includes floating point registers
- New parsing algorithm
  - Recursive traversal parser

Binary Rewriting

- Using the Binary Rewriter
  - Designed to leave API intact
  - Write Dyninst mutator as in dynamic case
    - Replace
      - bpatch->processCreate() with bpatch->openBinary()
      - continueExecution() with writeFile(outFile)
      - Run resulting outFile
  - Makes porting existing Dyninst program easy

Deconstructing Dyninst

Component	Release Version	Platforms
SymtabAPI	6.0	ELF - IA-32, IA-64, x86, x86_64, SPARC PE - Windows
StackwalkerAPI	1.0	x86/Linux x86_64/Linux PowerPC/Linux Blue Gene L
InstructionAPI	1.0	x86, x86_64
DepGraphAPI	Beta	x86, x86_64

SymtabAPI

SymtabAPI	Version 6.0	ELF - IA-32, IA-64, x86, x86_64, SPARC PE - Windows

New Features

- Port to Compute Node Linux
- Function and Variable abstraction
- Binary rewriting

SymtabAPI Binary Rewriting

- DyninstAPI – code rewriting
  - Instrumentation
  - Code modification
- SymtabAPI – object file rewriting (exe/shared libraries)
  - Add sections and symbols
  - Write ELF/PE/XCOFF file

StackwalkerAPI

StackwalkerAPI	Version 1.0	x86/Linux x86_64/Linux Compute Node Linux PowerPC/Linux BlueGene/L

- First and third party interfaces
  - Support for multi-threaded programs
- Walk through stack frames
  - Regular, Optimized, Signal handler …
- Customizable plug-ins to callback interface

InstructionAPI

InstructionAPI	Version 1.0	x86 x86_64

- Get opcode of instructions
- Find memory access
- Get register read sets and write sets
- Disassembly
- Evaluate and update instructions
- Expression Bind call
- Exposed to Dyninst BPatch interface

Integrating InstructionAPI to Dyninst

- Get instructions in basic block, parallel regions and instrumentation points
  - Replaces old BPatch_instruction interface
- Uses
  - Binary analysis
  - Smarter instrumentation

DepGraphAPI

DepGraphAPI	Beta	x86 x86_64

- Provides data dependence graph (DDG), control dependence graph (CDG), and program dependence graph (PDG)
- DDG provides def-use and use-def chains
- PDG provides forward and backwards program slicing

Dyninst 6.1

- New APIs
  - FlowGraphAPI – for gap parsing using machines learning
- Extended Functionality
  - Binary rewriter on Windows
  - InstructionAPI on POWER
  - StackwalkerAPI on BG/P, Solaris/Sparc, AIX/POWER

MRNet 2.1

- Platforms
  - x86/Linux, Windows
  - x86_64/Linux
  - POWER/Linux, AIX
  - SPARC/Solaris
  - Cray XT - coming soon...
- Building MRNet and applications
  - gcc, native compilers for AIX, Solaris, Windows
  - icc, pgCC

MRNet 2.1 New Features

- Stream based performance metric collection
  - CPU usage, memory usage for a rank
  - Number of bytes/packets send/received/processed by a filter/rank
- New k-nomial tree topology, in addition to balanced and generic trees

Conclusion

- Release in 2 weeks
- Downloads and documentation
  www.paradyn.org/html/downloads.html
  www.paradyn.org/html/manuals.html