SW 오류 자동 검출 및 검증기술

이 광근

서울대학교 교수

소프트웨어무결점연구센터 소장

04/12/2010 @ CEEDS/KAOC/ROSAEC 공동웍샵

ROSAEC center

Research On Software Analysis for Error-free Computing

소프트웨어 무결점 연구센터 KOSEF ERC

SW 분야의 근본 문제

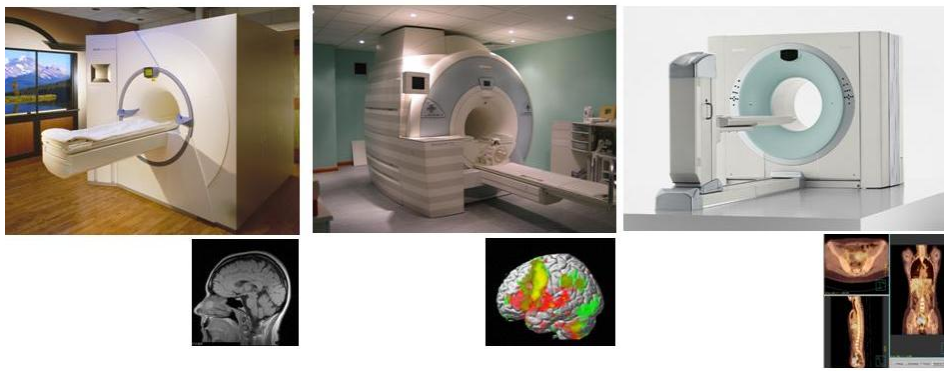
작성한 SW의 오류를 자동으로 미리 모두 찾아주거나, 없으면 없다고 확인해주는 기술들은 있는가?

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 KOSEF ERC

SW 오류 자동 검출 및 검증 기술

SW 소스(C, assembly, binary 등)를 자동 분석

"소프트웨어 MRI" "소프트웨어 fMRI" "소프트웨어 PET"



ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 KOSEF ERC

센터의 원천 기술

정적 프로그램 분석(static program analysis)[1]

- 엄밀히 예측
- 테스팅의 단점을 보완
- 실용성 확인됨
  - 산업화 완료한 SPARROW를 통해

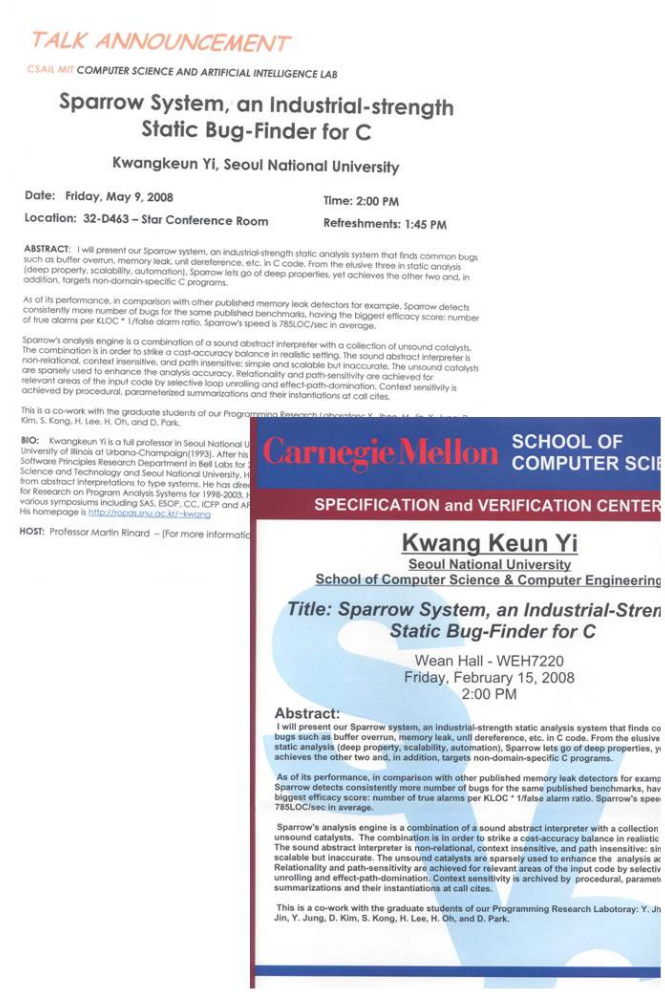
[1] *static analysis, abstract interpretation, type system, program logic, theorem proving, model checking*

원천기술 실용성 예시: SPARROW

대상: C, memory leak/buffer overrun, 오류 검출률 6/KLOC, 속도 100Loc/sec

Memory leak detection (SPEC2000 and open sources) (as of 01/04/2008)

Programs	Size KLOC	Time (sec)	True Alarms	False Alarms
art	1.2	0.68	1	0
equake	1.5	1.03	0	0
mcf	1.9	2.77	0	0
bzip2	4.6	1.52	1	0
gzip	7.7	1.56	1	4
parser	10.9	15.93	0	0
ammp	13.2	9.68	20	0
vpr	16.9	7.85	0	9
crafty	19.4	84.32	0	0
twolf	19.7	68.80	5	0
mesa	50.2	43.15	9	0
vortex	52.6	34.79	0	1
gap	59.4	31.03	0	0
gcc	205.8	1330.33	44	1
gnuchess-5.07	17.8	9.44	4	0
tcl8.4.14	17.9	266.09	4	4
hanterm-3.1.6	25.6	13.66	0	0
sed-4.0.8	26.8	13.68	29	31
tar-1.13	28.3	13.88	5	3
grep-2.5.1a	31.5	22.19	2	3
openssh-3.5p1	36.7	10.75	18	4
bison-2.3	48.4	48.60	4	1
openssh-4.3p2	77.3	177.31	1	7
fftw-3.1.2	184.0	15.20	0	0
httpd-2.2.2	316.4	102.72	6	1
net-snmp-5.4	358.0	201.49	40	20
binutils-2.13.1	909.4	712.09	228	25


TALK ANNOUNCEMENT

CSAIL MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LAB

Sparrow System, an Industrial-strength Static Bug-Finder for C

Kwangkeun Yi, Seoul National University

Date: Friday, May 9, 2008 Time: 2:00 PM

Location: 32-D463 – Star Conference Room Refreshments: 1:45 PM

ABSTRACT: I will present our Sparrow system, an industrial-strength static analysis system that finds common bugs such as buffer overrun, memory leak, unil dereference, etc. in C code. From the elusive three in static analysis (deep property, scalability, automation), Sparrow lets go of deep properties, yet achieves the other two and, in addition, targets non-domain-specific C programs.

As of its performance, in comparison with other published memory leak detectors for example, Sparrow detects consistently more number of bugs for the same published benchmarks, having the biggest efficacy score: number of true alarms per KLOC * 1/false alarm ratio. Sparrow's speed is 785LOC/sec in average.

Sparrow's analysis engine is a combination of a sound abstract interpreter with a collection of unsound catalysts. The combination is in order to strike a cost-accuracy balance in realistic setting. The sound abstract interpreter is non-relational, context insensitive, and path insensitive: simple and scalable but inaccurate. The unsound catalysts are sparsely used to enhance the analysis accuracy. Relationality and path-sensitivity are achieved for relevant areas of the input code by selective loop unrolling and effect-path-domination. Context sensitivity is achieved by procedural, parameterized summarizations and their instantiations at call cites.

This is a co-work with the graduate students of our Programming Research Laboratory: Y. Jhee, M. Jin, Y. Jung, D. Kim, S. Kong, H. Lee, H. Oh, and D. Park.

BIO: Kwangkeun Yi is a full professor in Seoul National U... University of Illinois at Urbana-Champaign(1993). After his... Software Principles Research Department in Bell Labs for 2... Science and Technology and Seoul National University. H... from abstract interpretations to type systems. He has dire... for Research on Program Analysis Systems for 1998-2003. H... various symposiums including SAS, ESOP, CC, ICFP and A... His homepage is http://ropas.snu.ac.kr/~kwang

HOST: Professor Martin Rinard – (For more informatio...

Carnegie Mellon SCHOOL OF COMPUTER SCI...

SPECIFICATION and VERIFICATION CENTER

Kwang Keun Yi

Seoul National University

School of Computer Science & Computer Engineering

Title: Sparrow System, an Industrial-Stren... Static Bug-Finder for C

Wean Hall - WEH7220

Friday, February 15, 2008

2:00 PM

Abstract:

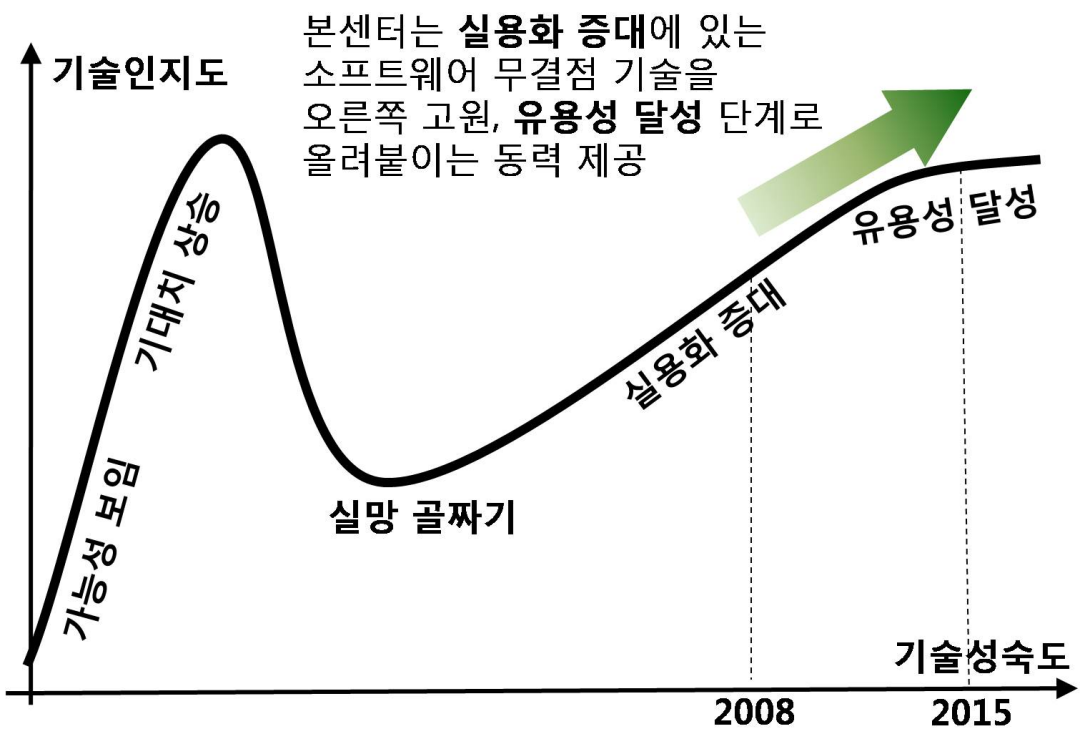
I will present our Sparrow system, an industrial-strength static analysis system that finds co... bugs such as buffer overrun, memory leak, unil dereference, etc. in C code. From the elusive ... static analysis (deep property, scalability, automation), Sparrow lets go of deep properties, y... achieves the other two and, in addition, targets non-domain-specific C programs.

As of its performance, in comparison with other published memory leak detectors for exam... Sparrow detects consistently more number of bugs for the same published benchmarks, hav... biggest efficacy score: number of true alarms per KLOC * 1/false alarm ratio. Sparrow's spee... 785LOC/sec in average.

Sparrow's analysis engine is a combination of a sound abstract interpreter with a collection ... unsound catalysts. The combination is in order to strike a cost-accuracy balance in realistic ... The sound abstract interpreter is non-relational, context insensitive, and path insensitive: sim... scalable but inaccurate. The unsound catalysts are sparsely used to enhance the analysis a... Relationality and path-sensitivity are achieved for relevant areas of the input code by selecti... unrolling and effect-path-domination. Context sensitivity is archived by procedural, paramet... summarizations and their instantiations at call cites.

This is a co-work with the graduate students of our Programming Research Labotoray: Y. Jh... Jin, Y. Jung, D. Kim, S. Kong, H. Lee, H. Oh, and D. Park.

기술 위치


본센터는 실용화 증대에 있는
소프트웨어 무결점 기술을
오른쪽 고원, 유용성 달성 단계로
올려붙이는 동력 제공
기술인지도
가능성 보임
기대치 상승
실망 골짜기
실용화 증대
유용성 달성
기술성숙도
2008
2015

정적 프로그램 분석(static program analysis)

프로그램의 실행 성질을
실행전에 자동으로
안전하게 어림잡는
일반적인 방법

"정적 분석" "static analysis"

응용: sw 오류검증, sw 테스트, sw 최적화, sw 관리, 등등 ∞

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 KOSEF ERC

- "실행전": 프로그램을 실행시키지 않고
- "자동으로": 프로그램이 프로그램을 분석
- "안전하게": 모든 가능성을 포섭
- "어림잡는": 실제 이외의 것들이 포함됨
  - 어림잡지 않으면 불가능
- "일반적": 소스언어와 성질에 무제한
  - C, Java, ML, Z, binary, etc.
  - "buffer overrun?", "memory leak?",
    "x=y at line 2?", "memory use $\leq$ 2K?", etc.

정적 프로그램 분석 기술의 한계 (SW 오류 분석에서)

허위 경보(false alarm)

- 오류가 아닌 것을 오류라고 판별
- 이론적으로 불가능: 허위경보가 항상 0

정적 프로그램 분석 기술의 실용성 (SW 오류 분석에서)

오류 검출과 무결점 검증

- 오류 검출(bug-finding)
  - 허위경보가 항상 적음 ($\leq 20\%$, non domain-specific)
  - 오류를 모두 찾지 못함
- 무결점 검증(verification)
  - 허위경보가 거의 0
  - 오류를 모두 찾는 것이 보장
  - 특정 SW에 대해서만 (domain-specific)

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 KOSEF ERC

오류 검출기(bug-finder)

있는 오류 대부분을 검출

- not exhaustive
- a few false-alarm
- domain-independent

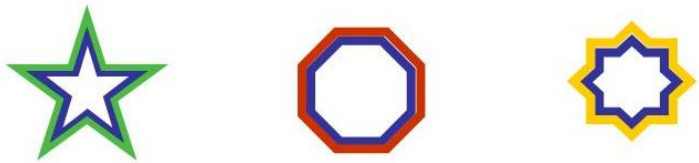
허위 경보율 $\leq 20\%$

무결점 검증기(verifier)

오류가 없으면 없다고 확인

- exhaustive
- zero false-alarm
- domain-specific

허위 경보율 $\leq 1\%$



서울대학교 소프트웨어 무결점 연구센터

Research On Software Analysis for
Error-free Computing (ROSAEC) Center

rosaec.snu.ac.kr

- 교육과학기술부/한국연구재단 지정 우수연구센터(ERC)
- 2008년 9월 설립. 향후 최장 10년간 지원받음.
- 교수 15명내외
  참여(고려대/서울대/한양대/KAIST/POSTECH 등)

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 KOSEF ERC

결론

- 정적 프로그램 분석 기술
  - 필요했던 기술(technology pull)
  - 실용적인 답이 가능할 만큼 성숙(technology push)
    - 센터의 경험

- 협력: 소프트웨어무결점연구센터 + CEEDS + 한국인공장기센터
  - 인공장기 시스템 SW에 특화된
  - 무결점 검증기(verifier) 공동개발
  - 한국인공장기센터에 제공

ROSAEC center
Research On Software Analysis for Error-free Computing
소프트웨어 무결점 연구센터 KOSEF ERC