

ISSN 2287-4348
Vol. 5 No. 1

한국스마트미디어학회 & 한국전자거래학회
2016 춘계학술대회 논문집

P R O C E E D I N G S

일시 : 2016. 04. 29 (금) ~ 30 (토)
장소 : 신라대학교 종합강의동

주최 : 한국스마트미디어학회
한국전자거래학회
신라대학교

포스터 발표순서 (4월 30일 토요일 13:00-14:30)

P18 **제목 :** K-평균 클러스터링 기반 앞서귀 질병 검출
272Page **저자 :** 박정현, 이성근, 고진광(순천대)

P19 **제목 :** Spark기반의 농업 빅데이터 분석 플랫폼 설계
275Page **저자 :** 뉴엔 신 녹, 뉴엔 반 쿼엣, 김경백(전남대)

P20 **제목 :** PO7(의사소통능력):프로그램 학습성과 평가 체계 모형 연구
279Page **저자 :** 오수열(목포대)

P21 **제목 :** IoT 기반 지능형 환기 히팅 시스템 설계
283Page **저자 :** 조동욱, 김민지, 이성근(순천대), 김강이((주)제노텍)

P22 **제목 :** 저전력 관련 코드 메카니즘과 소프트웨어 가시화 접목
286Page **저자 :** 이근상, 김영철(홍익대)

P23 **제목 :** 역공학을 통한 소스 코드로부터 유스케이스 설계 추출
289Page **저자 :** 권하은, 박보경, 김영수, 박지훈, 김영철(홍익대)

P24 **제목 :** 기존 오픈 소스 도구들 비교 분석을 통한 정적 분석 및 가시화 도구 구축
292Page **저자 :** 서채연, 박보경, 변은영(홍익대), 박용범(단국대), 김영철(홍익대)

P25 **제목 :** 프로파일러를 이용한 소프트웨어 메모리 성능 가시화 방법
296Page **저자 :** 강건희, 이진협(홍익대), 이근상(전북 TP), 김영철(홍익대)

P26 **제목 :** 의사결정기법(AHP)을 통한 한국형 테스트 성숙도 모델 요소의
298Page **적절성 검증에 관한 연구**
저자 : 박보경, 변은영(홍익대), 김기두(한국정보통신기술협회), 김영철(홍익대)

프로파일러를 이용한 소프트웨어 메모리 성능 가시화 방법

강건희^{1*}, 이근상^{2*}, 이진협^{3*}, 김영철^{4*}

홍익대학교 소프트웨어공학연구소, 전북 TP^{2}

e-mail : {¹kang, ³bob}@selab.hongik.ac.kr

e-mail : yi@jbtp.or.kr^{2*}, jinyub4649@naver.com^{2*}

Visualization Method for Software Memory Performance using Profiler

Geon-hee Kang^{1*}, Keunsang Yi^{2*}, Lee Jinhyub^{3*}, R. YoungChul Kim^{3*}

SE Lab., Hongik University, Jeonbuk TP^{2}

요 약

오늘날의 소프트웨어 산업은 점점 커지며 고품질에 대한 이슈가 대두되고 있다. 하지만 커지는 산업에 비해 시장 출하 기간의 단축상황으로 산업현장에서는 빠른 개발을 위한 코드 중심의 개발을 하게 된다. 그 결과 저 품질의 소프트웨어가 양산된다. 경쟁력을 가진 고품질의 소프트웨어를 생산하기 위한 인력과 비용이 우리나라는 부족하다. 그래서 소프트웨어 가시화가 필요하다. 본 논문에서는 소프트웨어의 성능(반응속도)가시화를 위해, 툴 체인을 구성과 방법을 제안한다. 1) 기존 가시화기법의 툴 체인과 프로파일러를 통해 소프트웨어 메모리 성능정보 접목한다. 2) 추출한 소프트웨어 구조정보를 통해 소프트웨어의 가시화를 하게 된다. 이 제안한 방법은 개발자뿐 아니라 다양한 이해관계자들이 소프트웨어의 성능에 대한 이해가 쉬워 질 것으로 예상된다.

1. 서 론

국내 대기업의 소프트웨어 개발은 SW공학적으로 접근하여 개발을 하고 있다. 하지만 중소기업은 SW공학적용을 위한 전문 인력과 개발시간, 자본 등의 부족 때문에 구현중심의 개발을 하고 있다. 이러한 우리나라의 소프트웨어 산업의 대부분인 중소기업의 상황이 변화하기 위해서는 프로젝트의 시작부터 소프트웨어 공학과 다양한 소프트웨어 개발 프로세스의 적용과 그것을 실행 시킬 인력을 배치가 필수다. 하지만 이미 개발이 시작된 프로젝트 중간에 적용하기가 쉽지 않고, 레가시 소프트웨어에는 적용할 수 없다. 우리는 기존연구[1,2]에서 역 공학을 통해 기존의 소프트웨어의 소스코드를 통해 품질지표의 적용과 가시화에 초점을 두었다. 기존 소프트웨어의 가시화를 통해 소프트웨어의 구조를 쉽게 파악하고 소프트웨어 품질지표도 같이 적용하여 소프트웨어의 품질판단과 어느 부분이 품질상의 문제점을 가지고 있는지 파악 가능하다. 그래서 본 논문에서는 프로파일러를 통해서 소프트웨어 메모리 성능정보를 얻고 이를 가시화하는 방법을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장 관련연구는 소프트웨어 가시화와 메모리 정보를 구하기 위한 Hprof(자바프로

파일러)에 대해 설명한다. 3장은 소프트웨어 메모리 성능 가시화 방법에 대해 기술한다. 마지막으로 4장에서는 결론을 언급한다.

2. 관련 연구

2.1 소프트웨어 가시화

소프트웨어의 특징 중 한 가지로 비가시성을 예로 들 수 있다. 이는 소프트웨어가 물리적인 형상을 갖지 않는 것을 의미한다. 소프트웨어 가시화는 이를 해결하기 위한 여러 기법을 의미한다. 이러한 기법에는 구조 가시화, 런타임 행동 가시화, 코드 가시화 등이 존재한다. 이 중 코드 가시화는 텍스트(Text) 형태의 코드를 쉽게 이해하도록 하는 기법이다. 예를 들어 객체지향 코드에서 클래스 간 관계를 코드에 나타내거나, 변수 간 참조 관계를 화살표로 표현하는 것 등이다. 그리고 런타임 행동가시화는 소프트웨어가 실제로 구동되는 상태에서 정보(CPU, Memory 등)를 얻어 해당 정보를 그래프로 표현하는 것이다[3]. 본 연구에서는 코드가시화와 행동가시화를 접목하여 소스코드의 구조와 모듈의 메모리 정보를 가지고 가시화하고자 한다.

2.2 Hprof

Hprof는 Oracle Java SDK에 같이 동봉되어 제공되는 프로파일러이다. 기본적으로 CPU사용률, Heap Allocation

* 본 연구는 2015년 교육부와 한국연구재단의 지역혁신창의인력양성사업(NRF-2015H1C1A1035548)과 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A2011601).

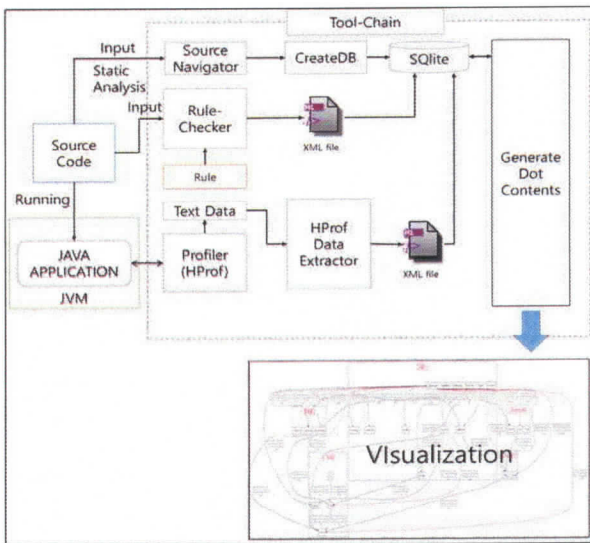
통계, monitor contention profiles 등의 정보를 얻을 수 있다[4]. 아래 그림1은 Hprof의 결과 데이터이다.

rank	self	accum	count	trace	method
1	40.42%	40.42%	10	301786	Stamp.stampTest
2	7.66%	48.07%	10000000	301784	SampleDataClass.getC
3	7.51%	55.59%	10000000	301783	SampleDataClass.getB
4	7.41%	62.99%	10000000	301785	SampleDataClass.getD
5	7.37%	70.37%	10000000	301782	SampleDataClass.getA
6	4.00%	74.37%	1466670	301944	java.lang.Character.digit
7	2.58%	76.95%	1466694	300653	java.lang.CharacterDataLatin1.digit
8	1.26%	78.24%	100000	302034	java.lang.Integer.parseInt
9	1.26%	79.52%	100000	301946	java.lang.Integer.parseInt
10	1.26%	80.79%	100000	301935	java.lang.Integer.parseInt
11	1.11%	81.90%	1466694	300651	java.lang.CharacterData.of
12	1.10%	83.00%	1466694	300652	java.lang.CharacterDataLatin1.getProperties
13	0.88%	83.88%	488890	301945	java.lang.Character.digit
14	0.87%	84.76%	488890	302033	java.lang.Character.digit
15	0.87%	85.63%	488890	301994	java.lang.Character.digit
16	0.86%	86.48%	1	302119	SampleMain.main
17	0.73%	87.21%	1	301973	External1.RandomNumberRead
18	0.65%	87.86%	1	302012	External2.RandomNumberRead
19	0.52%	88.38%	200000	302072	Common1.CommonTest
20	0.44%	88.83%	588890	301943	java.lang.Strings.charAt
21	0.44%	89.27%	588890	301993	java.lang.Strings.charAt
22	0.43%	89.70%	588890	302032	java.lang.Strings.charAt
23	0.36%	90.06%	100000	302076	Common2.CommonTest2

(그림 1) Hprof 결과 데이터

그림 1과 같이 프로파일링이후에 결과가 텍스트파일로 자동으로 출력 되게 된다. 'Rank, self, accum, count, method'순으로 정보가 표시된다. HProf가 커맨드라인으로 구동이 되고 자동으로 데이터파일이 생성되기 때문에 소프트웨어 메모리 성능 가시화에 최적화된 프로파일러라고 판단하여 메모리 성능 가시화에 적용을 하고자 한다.

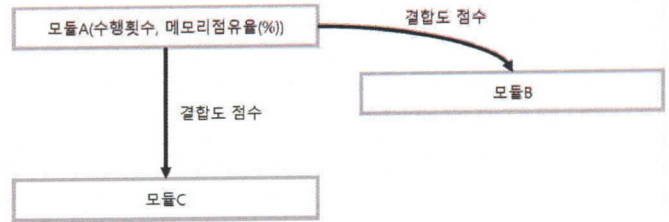
3. 프로파일러를 이용한 메모리 성능 가시화 방법



(그림 2) 메모리 성능가시화를 위한 툴 체인 구성도

그림2는 소프트웨어 메모리 성능 가시화를 위한 툴 체인의 구성도이다. 소프트웨어 성능 가시화를 위한 툴체인에는 기존 소프트웨어 성능 가시화를 위한 툴체인[1]에 소프트웨어 메모리 성능의 좋지 않은 패턴을 추출하기 위한 RuleChecker(PMD[5])와 소프트웨어의 동적분석을 위한 Profiler(Hprof[4]), Profiler에서 추출된 데이터를 Xml데이터로 정제하기 위한 HprofDataExtractor를 추가하였다. 이를 통해 메모리 성능에 위배되는 코드의 정보와 기존에 설정한 소프트웨어 구조정보(결합도, 응집도), 소프트웨어

모듈의 메모리 사용정보 등을 DB에 저장 하고 GenerateDotContents를 통해 가시화 그래프를 그리게 된다. 다음 그림3은 소프트웨어 메모리 성능 가시화 이미지의 구성을 설명하는 그림이다.



(그림 3) 메모리 성능가시화 이미지 구성

그림3을 보면 모듈을 표현하는 박스 안에 모듈의 이름과 모듈의 수행횟수, 메모리 점유율이 표시된다. 그리고 모듈과 모듈 사이에는 모듈간의 결합도가 표현된다. 가시화된 이미지를 통해 실제 모듈의 수행횟수와 메모리 점유율을 보면서 쉽게 메모리를 많이 사용하는 부분을 구분할 수 있다.

4. 결론

단발성의 소프트웨어 개발, 시장 출하기간의 단축 등으로 인한 구현 중심의 개발 프로세스와 잦은 개발 인력의 이동으로 개발 문서의 부재, 잦은 패칭으로 국내 소프트웨어의 품질저하의 악순환이 지속되고 있다. 그래서 본 논문에서는 프로파일러를 통한 메모리 성능 가시화 방법을 제안하였다. 제안하는 방법을 통해 이전의 소프트웨어 가시화 과정과 동시에 자바 성능측정도구(HProf)를 통한 동적 분석을 수행하여 정적 분석에서 얻을 수 없는 실제 수행횟수와 메모리 점유율의 정보를 얻어 가시화 할 수 있을 것이다. 이를 통해 실제 개발자뿐 아니라 많은 이해관계자들도 소프트웨어의 메모리 성능 지표와 구조를 쉽게 이해할 수 있을 것으로 보인다.

참고문헌

[1] 강건희, 박보경, 장우성, 황준순, 권하은, 이한솔, 이현준, 김영철. "소프트웨어 성능 가시화를 위한 툴 체인 개발", 『한국정보과학회 소프트웨어공학회』 Vol. 18, No. 1, 2016. pp.395-398.
 [2] 강건희, 이근상, 김동호, 황준순, 김영수, 박용범, 김영철. "절차식 언어 기반의 코드 정적 분석을 위한 툴 체인 사례 연구," 『한국정보과학회 2014 한국 컴퓨터 종합학술대회 논문집』, 2014, pp. 559-561
 [3] Thomas Ball, Stephen G. Erik, Bell Laboratories, "Software Visualization in the Large", IEEE Computer Society, Volume 29, Issue 4, April 1996.
 [4] HPROF: A Heap/CPU Profiling Tool, <https://docs.oracle.com/javase/7/docs/technotes/samples/hprof.html>
 [5] PMD, <https://pmd.github.io/>

한국스마트미디어학회 & 한국전자거래학회
2016 춘계학술대회 학술발표 논문집

Proceedings of KISM & SEB Spring Conference 2016

제 5권 제 1호
2015년 4월 25일 발행

발행인 / 차준섭, 김훈태 대회장

편집인 / 김병기, 이홍주, 이성근, 김영철 학술위원장

발행처 / (사) 한국스마트미디어학회

광주 남구 송암로 60 광주CGI센터 기업동 309호 (송하동)

전화 : 062)655-3507 / 팩스 : 062)655-3510

홈페이지 : www.kism.or.kr

E-Mail : kism1122@kism.or.kr

디자인 및 편집 / 장 영 우(한국스마트미디어학회 연구원)

후원 / LG CNS, 아이티센, NICE R&C, 콤텍시스템, 비온시이노베이터

한국IT비즈니스진흥협회, 소프트캠프(주), (주)피앤피시큐어, 함소아한의원

중앙대 의료보안연구소

2016
SPRING
CONFERENCE
OF
KISM & SEBS



Smart Media
KOREAN INSTITUTE OF SMART MEDIA



Society for
e-Business Studies



NICE R&C 주식회사
NICE Research & Consulting, Inc



BonC Innovators



함소아 한의원