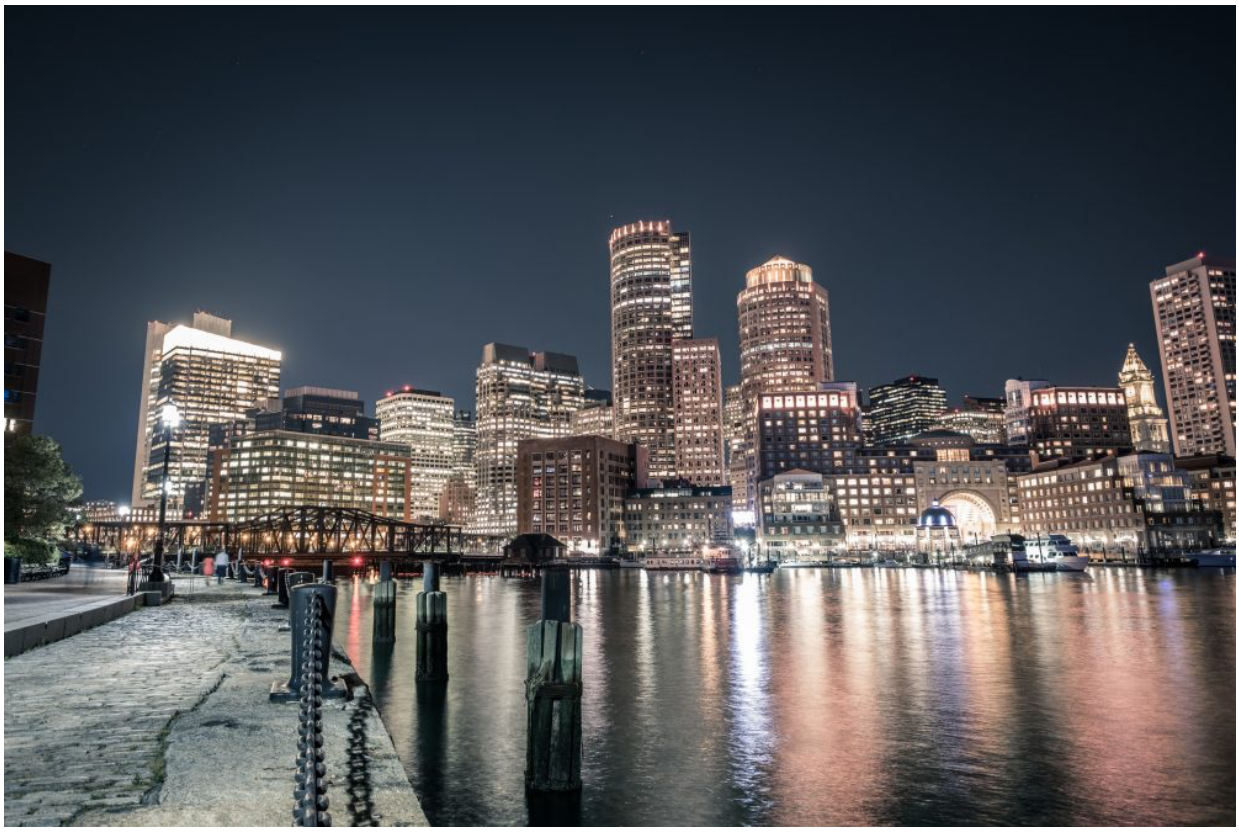
Splash 2018 Trip Report
Boston, USA

고려대학교 소프트웨어 분석 연구실 전민석



들어가며

작년에 이어 올해도 논문 발표를 위해 보스턴에서 열린, OOPSLA 2018 에 참석하였다. 이번에 발표한 연구는 개인적으로 애착을 가지고 했던 연구인데, 그 이유는 이 논문이 제시한 문제가 “다른 사람은 놓쳤지만 내가 보았다는 것” 때문이다. 이는 큰 동기를 부여 했었고, 강력한 동기는 좋은 연구 결과로 이어졌고 이는 좋은 학회에 논문이 발탁되는 것으로 이어졌었다. 학회도 즐겁게 참석하였고, 많은 것을 보고 느낄 수 있었다. 이번 Trip Report 는 SPLASH 2018 을 보고 느낀 바 들을 전하고자 한다.

작년과 비교해서..

참가자는 작년보다 훨씬 많았다. 포스터 세션에선 모든 포스터에 사람이 계속 끊이지 않고 설명을 듣고 있었다. 작년에 봤던 저자들이 이번에도 많이 참가한 걸 볼 수 있었다. 이를 직접 보면 질 수 없다는 생각에 나도 꾸준히 내서 매년 참가 해야겠다는 동기부여가 된다. Venue 는 확실히 작년보다 못하였다. BOSTON 의 물가는 너무나 부담스러웠고 날씨도 비가 와서 활발히 돌아 다니지도 못하였다. 미국의 시차 또한 적응하기 너무 힘들었다(미국 출장을 가게 될 땐, 시차 적응을 어떻게 할 지 계획을 잘 세워야 할 것 같다). 발표장의 의자도 부족하여 세션 중간에 다른 발표장으로 자리를 옮기면 서서 들어야 하는 불편함을 감수 했어야 했다.

논문 발표

이번 논문은 멋지게 발표를 하고 싶었기에, 사실 오래전부터 발표를 준비를 했었다. 처음엔 영어 전달력이 중요하다고 생각해 영어 말하기를 오랬 동안 연습을 했다. 약 10 개월동안 아침과 통학 시간을 투자해 영화 대본을 보고 영화와 똑같이 따라하는 연습을 했었고 영어 전달력을 키웠었는데, 결과적으로 Tunneling 이라는 아이디어에게 미안할 정도로 발표를 망쳤다. 발표를 망친 가장 큰 이유는 준비한 내용의 부족이었다. 준비한 내용이 부족하니 발표가 빨리 끝났고 뭔가 발표를 하다가 만 느낌이었다. 논문에는 재미있는 내용이 풍부했는데, 이러한 내용들을 거의 준비하지 않은 것이 많이 아쉽다. 망친 두번째 이유는 말을 너무 빠르게 했다는 것, 안 그래도 준비한 내용이 부족했는데 말까지 빠르게 해버려서 몇명 빼고는 거의 이해를 못한 것 같았다. 쉬운 아이디어이고 천천히 그리고 자세히 설명했다면 모든 사람이 이해하고 감탄할 만한 아이디어인데 너무 아쉽다. 내년에는 풍부한 내용준비하는 것과 천천히 전달하는 연습을 하여 올해 와 같은 재앙이 일어나지 않도록 할 것 이다.

사람들과의 교류

학회에 저자로서 참석하게 된다면, 최소한 같은 세션에서 발표하는 사람들과는 찾아가 인사하고 많은 이야기를 해보길 권한다. 같은 분야를 연구하는 사람들 이므로 다른 세션 논문 저자들과 이야기 할 때보다 서로의 연구를 이해하기 훨씬 쉽고 좋은 코멘트를 주고받으며 친해질 수 있다. 어차피 또 보게 될 확률이 높은 사람들이니 일찍 친해질 수록 좋다. 올해엔 일 부 사람들과 깊은 교류가 있었는데, 가장 깊게 교류한 사람은 Tian

tan 이다. 비록 나이는 어리지만 Tian tan 은 나와 나 의 연구에 가장 크게 영향을 미친 사람이다. 내 연구는 Tian tan 의 SAS16 발표에서 시작 되었다. 교수님이 16 년도에 Tian tan 의 발표를 들으시고 “아 우리도 포인터 분석을 해야겠다!” 고 생각 하신 후 나의 모든 연구가 시작 되었다. 이런 얘기들을 주고 받다 보니 다음 연구에 대한 얘기로 자연스럽게 이어졌고 이에 대한 co-work 이야기까지도 나왔었다. 이처럼, 학회장에서 열심히 소통하다 보면 자신이 동경하는 사람들과 co-work 할 수 있는 기회가 생길 수도 있다.

목표로 하는 학회가 한국에서 열린다면 꼭 논문을 내라

만약 한국에서 자신이 목표로 하는 좋은 학회가 열린다면, 꼭 내볼 것을 권한다. 조금은 근거 없는 소리지만, 붙을 확률이 조금이라도 올라 갈 것이다. OOPSLA18 에서 느낀 또 다른 특징 중 하나는 MIT 의 paper 가 눈에 띄게 많았다는 것이다. 올해 OOPSLA 학회장은 MIT 에서 버스한번이면 갈 수 있는 거리에 있었는데, 그 때문인지 MIT 학생들이 유난히 많이 참석을 한 것을 느낄 수 있었다. 참석한 것 뿐만 아니라 논문도 많이 accept 되었는데, MIT 학생들도 이번에 MIT 에서 많이 붙었다고 이야기 하더라.

Talks

재미있게 들었던 talk 들을 몇개 소개하겠다.

1. Explicit Direct Instruction in Programming Education

“어떻게 학생들에게 프로그래밍을 가르쳐야 하는가?” 에 대한 이야기를 들을 수 있었다. 학생들에게 프로그래밍을 가르치는 방법은 크게 2 가지 방법이 있다고 한다. 첫번째는, 선생님으로서 최소한의 역할만을 하고 학생들이 스스로 프로그래밍의 재미와 원리를 깨우칠 수 있도록 하는 것이다. 두번째 방법은, 학생이 알아야 하는 모든 것들을 알려주며 가르치는 방법이다. 발표하신 분이 주장은 두번째 방법이 더 효과적이라는 것이다. Programming Education 으로 연구하는 사람들은 한번 들어보는 것도 좋을 것이다.

2. The DaCapo Benchmark Suite: A Methodological, Engineering, and Social Journey

DaCapo Benchmark 라는 최고 권위의 자바 프로그램 벤치마크가 어떻게 만들어지고 관리되는지에 대한 이야기를 들을 수 있었다. 나의 연구는 주로 Java program 을 타겟으로 하기 때문에, 나의 모든 아이디어는 DaCapo Benchmark 를 통해 실험적으로 검증을 거쳤었다. 나 뿐만 아니라 모든 Java program 에 대한 연구에서 DaCapo Benchmark 는 반드시 검증을 거쳐야 하는 또 하나의 reviewer 이다. 이런 권위있는 Benchmark 를 만들었다는 것만으로 발표에서 자부심이 느껴졌었다. 아무 지원없이 스스로의 동기부여로 시작한 프로젝트였고, 좋은 벤치마크로서의 역할을 다할 수 있도록 오랬 동안 연구했음을 알 수 있었다. DaCapo Benchmark 에서 나쁜 성능을 보이는 기술(연구)들은 좋지 못한 기술이라며 다시 한번 생각해 봐야 한다고, 강력하게 얘기하시던 모습이 인상적이었다. 좋은 Benchmark 를 만드는 것은 큰 contribution 임이 분명 하고 좋은 연구이다.

3. Decompiling Ethereum Bytecode and Detecting Gas-Focused Vulnerabilities.

최근 가장 핫 한 연구들 중 하나인 Smart Contract 취약점 탐지에 대한 톡이었는데, EVM Bytecode 를 decompile 하는 테크닉을 다루고 있다. 이 연구는 주어진 EVM Bytecode 를 high level 3-address-code 로 decompile 해주는 데 특이한 점은 declarative logic 을 이용해 구현을 했다는 것. 이 decompiler 는 약 99.98%의 Bytecode 를 cover 할 수 있다고 한다. 나왔던 질문 중 흥미로웠던 것은 “decmopile 된 코드가 실행이 되는가?” 였는데, decompile 된 코드가 실행할 수 있는 코드는 아닐 수 있다고 한다.

논문 소개

1. Precision-Guided Context Sensitivity for Points-to Analysis

Precision critical 한 method 들의 특징을 제시하고 이를 전 분석을 이용해 찾고자 했던 연구이고, 나의 연구들과 가장 관련이 깊은 논문이다. 실험에 따르면 약 20%정도의 method 들에게 2-object sensitivity 를 적용하였지만 모든 method 들에게 적용한 것의 98% precision 을 유지했음을 보였다. 나의 작년 연구는 precision critical 한 method 들을 machine learning 을 이용해 찾는 방법을 제시했는데, 작년 연구의 단점

중 하나가 성능은 좋은데 왜 잘되는지 설명을 할 수 없었다는 것이었다. 반면 이 논문의 가장 큰 장점은 왜 잘 되는지를 설명을 잘 해 놓음으로써 독자들이 왜 잘 되는지 이해할 수 있다는 것이다. 작은 criticize 를 하자면, 이 연구에서 제시한 heuristic 은 큰 프로그램에 대해선 아직 unscalable 하다는 것이다. Dacapo Benchmark 에서 2-obj 가 제한시간 안에 분석을 끝내지 못하는 프로그램이 있는데(jython), 이 연구가 제시한 heuristic 은 그 프로그램에 대해선 분석이 제한시간 안에 끝나지 않는다. 선택적으로 context sensitivity 를 적용하는 것의 큰 목적 중 하나는 큰 프로그램을 분석 가능하게 만드는 것이기에, jython 에 대해 분석이 끝나지 않는다는 점은 약점이라 할 수 있다. 우리의 작년 연구에서는 jython 조차 빠르게 끝나는 very scalable 한 heuristic 을 제시했었다.

2. MadMax: Surviving Our-of-Gas Conditions in Ethereum Smart Contracts

이 논문은 Ethereum Byte code 로부터 gas 와 관련된 취약점을 정적 분석을 통해 자동으로 detect 하고자 하였다. Smart Contract 는 요즘 가장 핫 한 주제 중 하나이다. Smart Contract 의 취약점은 큰 경제적 피해로 이어지기 때문이다. 좋은 주제이고, distinguished paper award 까지 받았기에 발표장에 사람도 가장 많았었다. 이 논문의 특이한 점은 byte code 를 분석 한다는 것이다. 주어진 byte code 를 우선 3-address code 형태의 중간 언어로 decompile 한 후, 그 위에서 정적 분석을 통해 취약점을 detect 한다. 정적 분석은 다양한 취약점을 검출해 내는데 그 중에는, unbounded mass operation(user 의 input 이 매우 큰 loop 의 iteration 을 초래해 많은 양의 Gas 를 소모하게 하는 것), Non-Isolated External Calls(외부 함수때문에 문제가 발생하는 것) 그리고 우리가 잘 알고 있는 integer overflow 등이 있다. 이들을 검출해 내는 data-log logic 을 디자인 했으며 많은 취약점을 검출해 냈다. 이 tool 은 Ethereum block chain 상의 대부분의 byte 코드를 분석 할 수 있을 정도로 Scalability 가 높은 것이 장점이다.

3. Format Abstraction for Sparse Tensor Algebra Compilers

이 연구도 우리처럼 사람이 손으로 하기 힘들어 하는 것을 자동화 한 연구이다. Tensor algebra 는 대중적으로 많이 쓰이는 다차원 데이터 계산용 tool 로서, 머신 러닝, 데이터 분석 등에 사용된다. Tensor algebra 에서는 다양한 데이터 format 이 존재하는데, SCR, COO, DIA, ELL, HASH 등이 있고, 각 format 마다 장단점이 존재하고 적합한 상황이 따로 있다고 한다. 문제는 사용자가 어떤 format 을 사용하다가 다른 format 으로 바꿔서 Tensor algebra 를 계산하려면 코드를 완전히 새로 짜야 한다는 것이고 이 과정이 어렵다는 것이다. 이 연구는 이 과정을 안전하게 자동으로 바꾸어 주는 방법을 제시

했다. 우리의 Data-driven 연구와 motivation 이 비슷해 자세한 내용은 정확히 이해하지는 못했지만 motivation 에 대한 공감은 많이 할 수 있었다. 주 저자가 MIT 학생인데, 이번 학회에서 점심 시간에 우연히 계속 같이 앉게 되어 얘기도 많이 했었다. MIT 나 Harvard 에 대해 많은 정보를 주었고, 마지막 날 점심시간 후 "내년 OOPSLA19 Athena 에서 보자!" 하며 헤어졌다.

후기

연구 시작부터 학회 발표(전)까지 굉장히 즐거웠던 연구였다. Tunneling 이라는 멋진 아이디어를 만들어 냈을 때 느낀 짜릿함, Tunneling 이라는 문제를 탐구하면서 느낀 재미, Tunneling 이 실제로 엄청난 성능 향상을 가져옴을 evaluation 을 통해 확인할 때의 통쾌함 등이 기억에 남는다. 내년에도 Context Sensitivity 로 연구를 할 것이다. 내년에는 발표까지 완벽하게 마무리 할 것이다.

올해 역시 연구가 성공적으로 끝난 가장 큰 이유는 생산적인 연구 미팅이다. 생산적인 연구 미팅이 이어진다면 멋진 연구가 완성된다. 초기 연구미팅에선 좋은 문제를 찾아 discuss 를 하며 강력한 motivation 을 갖추었고, 중반 연구미팅에선 approach 에 대한 깊은 discuss 를 해가며 논문을 탄탄하게 만들었으며, 후반 연구미팅에선 새로운 발견 등을 정리하며 논문의 내용을 풍성하게 만들었다. 모든 미팅이 성공적인 것은 아니었지만, 모든 미팅이 성공적이 되도록 노력하는 것이 중요한 것 같다. 매 미팅 마다 최소 하나의 새로운 포인트라도 만들어 갈 수 있도록 노력 하면 연구는 목표로 하는 곳의 due 안에 반드시 완성할 수 있다. 매번 생산적이고 적극적인 미팅으로 이번 연구를 이끌어 주신 오학주 교수님께 감사하다고 전해드리고 싶다. 또한 나의 research mate 세훈이 형에게도 격한 감사를 전하고 싶다. 내 다음 연구 또한 OOPSLA 를 목표로 하고 있다. 처음 대학원에 입학하면서 세운 목표가 OOPSLA 개근하기 였는데 목표달성을 할 수 있도록 노력 할 것이다.

이쯤에서 나의 두번째 Trip report 를 마무리 하겠다.