Proof Carrying Code

Native Client

Untrusted

Programmer

↓

Code

↓

Compiler (Inserted checks to

↓ make verification easy)

Binary

↓

Trusted

Verifier

↓

User

Verifier Checked 2 Properties

- R/W were safe
- J MPs safe

PCC

Programmer

↓

Code

↓

Compiler

↓

Binary, Proof (Correction Condition)

↓

Verifier

↓

User

- Verification Simpler to Code
- Verification can be faster to perform
- No/little run-time checking
- Proof may be derived from source code

Int foo(int a)
{
    Int b = a;
    If(b>0)
        b = -b;
    return b;
}
Post foo(x) >= 0

Def) wp(s, p)   s: program statement
     p: post condition
     is the weakest predicate Q such that if Q is true before executing S, then P will be true afterwards.

EX)
wp(x := x+1, x==7)
   = x==6 (x+1==7)
Wp(x:=x+1, y==5)
   = y == 5

Wp(x:=e,p)
   = P[e/x]

Wp(s1;s2,p)
   =wp(s1,wp(s2,p))

Wp(if(e) then S1 else S2, p)
   (e => wp(s1,p)) ^ (~e => wp(S2,p))

Pre: |a| <=10
Int foo(int a)
{
    Int b = a;
    If (b<0);
b = -b;
return b;
}

Post : 10 >= foo(a) >= 0

Wp(b=a; if(b<0) then b=-b else{}, b>0)
=wp(b=a, b<0 => -b>=0 ^ b>= b>=0)
=(a<0 => -a>=0) ^ (a>=0 => a>=0)

Compiler

{pre} s {post}
  ↓
Verification Condition Generation
  ↓
VC:Pre=>wp(s,post)
  ↓
Theorem Prover
  ↓
Proof

User

{pre} s {post}
  ↓
VCgem
  ↓
VC
  ↓
Verifier <=P
  ↓
Valid/Invalid