ACCESS SOFTEK, INC

Experiments with two-phase expression evaluation for a better debugging experience

Ilya Kuklin

Anton Korobeynikov

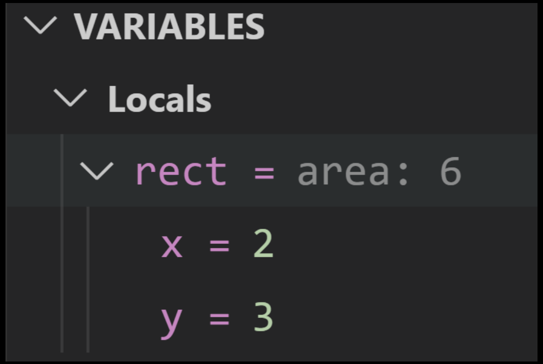
Motivation

- Certain debugging functionality requires expression evaluation

```
// C++:
struct Rectangle {
  unsigned x, y;
};
```

```
# Python:
def RectangleSummaryProvider(valobj, dict):
    Name = valobj.GetName()
    Expr = "%s.x * %s.y" % (Name, Name)
    ValRef = valobj.CreateValueFromExpression('area', Expr)
    Area = ValRef.GetValueAsUnsigned(0)
    return "area: %d" % Area
```

VS Code:

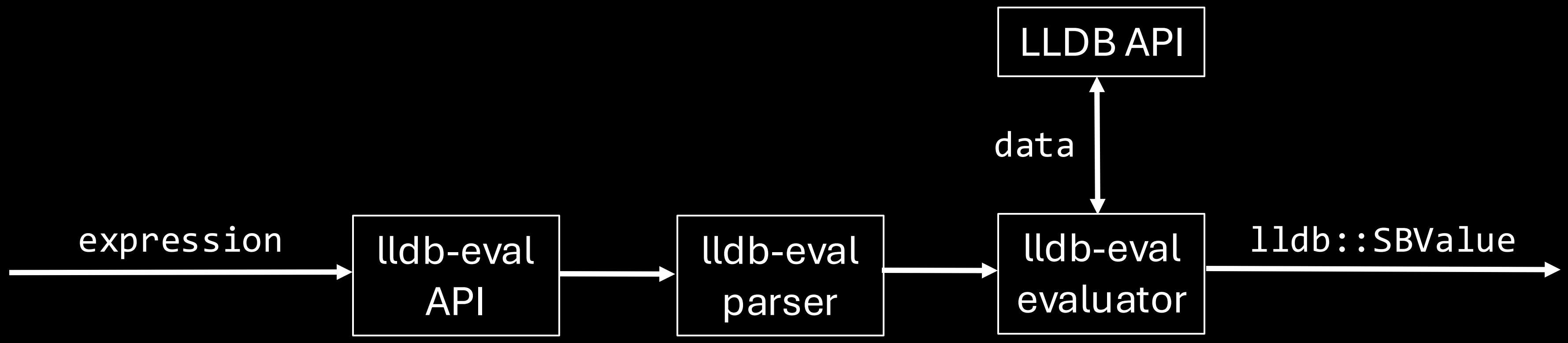


```
VARIABLES
  Locals
    rect = area: 6
      x = 2
      y = 3
```

- These expressions tend to be small and simple
- They can be evaluated in a faster but limited way

lldb-eval

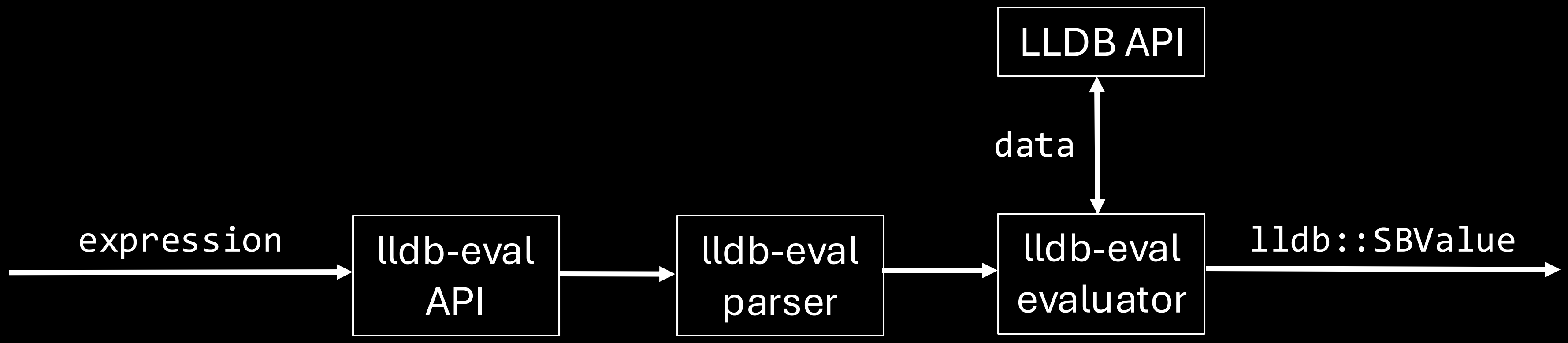
- A very fast C++ interpreter for a limited language subset
- Initially created by Google for Stadia and was presented at LLVM Dev Mtg 2021
- Has its own parser, evaluator, and an API
- Relies only on the debug information retrieved using LLDB API
- Library: cannot be used standalone



lldb-eval: LLDB integration

ACCESS SOFTEK, INC

- Rebased onto LLVM ToT
- Integrated into LLDB to be automatically used in conditional breakpoints, creating values from expressions, and generally any explicit expression evaluation
- Can now be used to evaluate simple expressions quickly and fall back to full LLDB evaluation for complicated expressions



Use cases

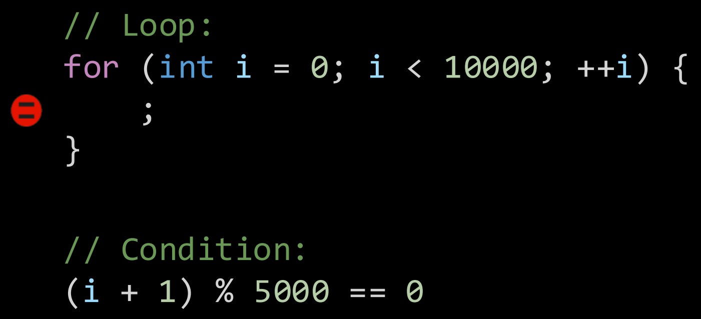
- Custom type formatting

Debugging performance in a large game engine

	lldb-eval (via LLDB)	LLDB	LLDB + lldb-eval overhead
1 expression	0.65 ms	87.9 ms	88.9 ms
Total for all local and global variables	103 ms	2025 ms	2041 ms

Use cases

- Evaluating conditional breakpoints



```
// Loop:
for (int i = 0; i < 10000; ++i) {
    ;
}

// Condition:
(i + 1) % 5000 == 0
```

	LLDB	lldb-eval
Time until breakpoint is triggered	18 seconds	7 seconds
Time spent only on expressions	12 second	1 second

Results

- The approach is viable: use lldb-eval for simple expressions, fallback to LLDB for complex expressions
- The speed up depends entirely on the need for expression evaluation and complexity those expressions
- Code: https://github.com/access-softek/llvm-project/tree/add-lldb-eval

Thank you!