High Level Virtual Machine: Status and Goals

LLVM Developers’ Meeting
May 25, 2007
What Is HLVM?

- Right now, nothing much. 0.2 release
- Purpose:
  - Be the “front end toolkit” for LLVM.
- Goals:
  - Make developing new languages “easy”
  - Support languages with runtime compilation features.
  - Provide language inter-operation features
  - Provide a robust virtual machine (may be a separate project)
Features

- **HLLDL: High Level Language Definition Language**
  - XML-based Declarative language for language definition
  - Much “higher” level than Yacc/Lex or ANTLR
  - Only targets the HLVM AST
- **Extensible Abstract Syntax Tree**
  - Languages can use built-in features or extend for language-specific constructs
  - Optional bi-directionality supports language conversion
  - XML representation of AST
- **Built-in support for:**
  - threading, synchronization, continuations, closures, objects, memory management, input/output + more
- **Runtime allows HLVM based languages to be portable**
Design Philosophy

• Much the same as LLVM’s:
  • Toolkit approach (small libraries)
  • Focus on performance (compile & run time)
  • You only pay for what you use.

• For example:
  • AST designed for fast construction
  • Executables just make calls to libraries.
  • If your language doesn’t have objects, your runtime won’t have object support linked in.
Significant Components

• **HLLDLC**
  • Compiler for High Level Language Definition Language

• **Adaptors:**
  • Adaptors for ANTLR and YACC/LEX to make converting existing languages easier.

• **AST:**
  • Abstract Syntax Tree: the heart of HLVM

• **Generators**
  • Convert AST to LLVM, HTML, JIT, etc.

• **HLVM**
  • Reference implementation of the runtime (APR based)
Current Status

- Early: release 0.2
- No languages supported, yet.
- AST supports LLVM constructs (only)
- Generator works (needs 2.0 update)
- Passes significant test suite (XML->XML)
- Much more to be done …
Questions?