DIVINE 4

Vladimír Štill

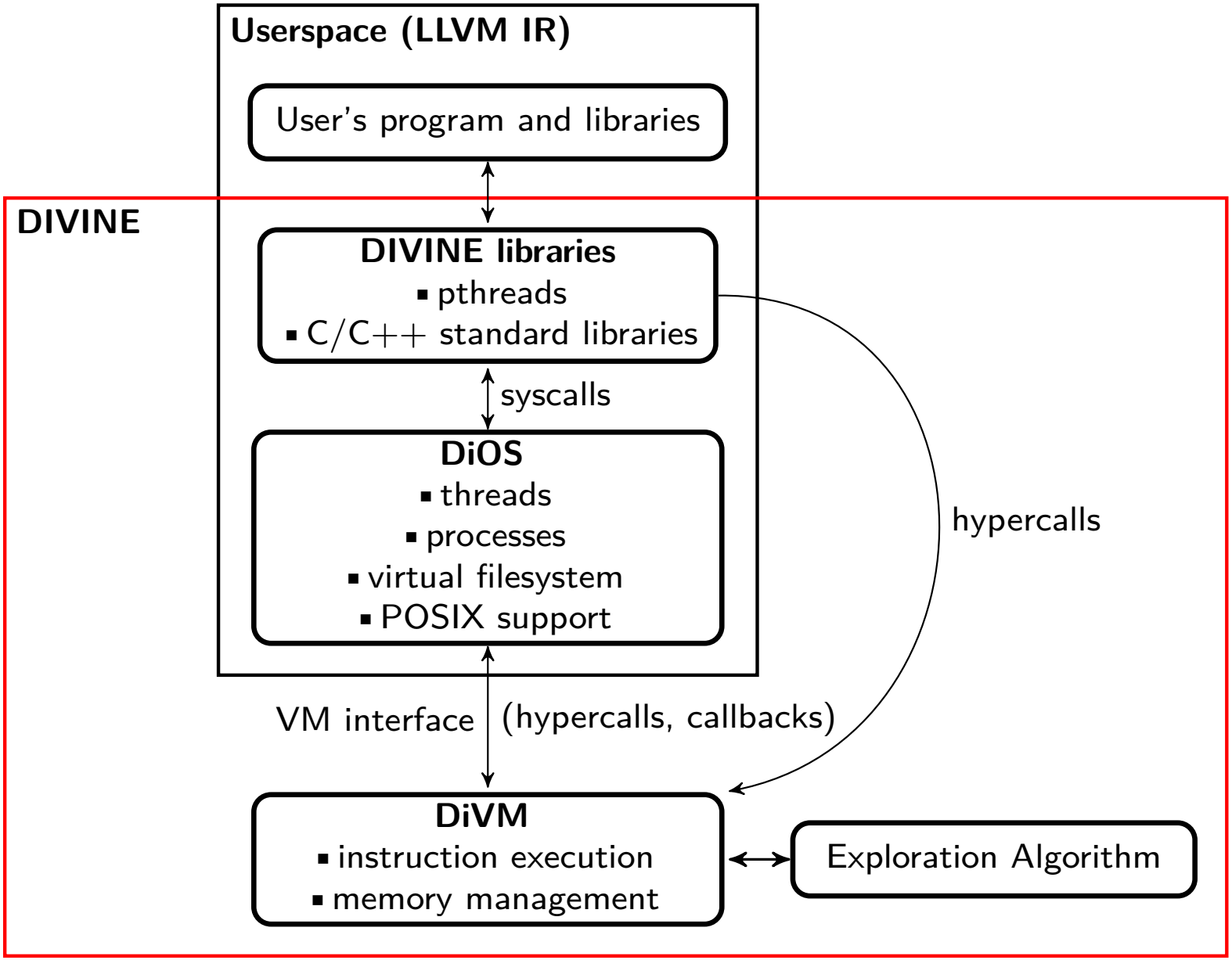
Masaryk University

Brno, Czech Republic

Co je DIVINE?

- nástroj na testování a verifikaci programů v C a C++
- se zaměřením na paralelní programy a detailní detekci problémů s pamětí
- využívá LLVM IR (jednodušší programovací jazyk používaný při překladu C/C++ na nativní kód)

Architektura DIVINE 4



Hlavní myšlenkou DIVINE 4 je oddělení velké části kódu do DiOS

- odpovídá operačnímu systému
- DiVM (interpret) nemusí řešit plánování vláken, jen nedeterministickou volbu
- DiOS běží uvnitř DIVINE takže se snáze testuje než interpret
- DiOS lze v případě potřeby vyměnit za jiný systém s jiným plánováním (například synchronní paralelismus)
  - případně vyměnit jeho části (plánovač)
- snadněji rozšiřitelný než DiVM

DiOS: plánovač

Plánovač vláken (a procesů) je základní součástí DiOS

- samotné spouštění uživatelského kódu
- řeší prokládání vláken
- a spouštění procesů (`fork`, `exec`, případné sdílení paměti mezi procesy)
- při přerušení uživatelského kódu DiVM spustí plánovač
- uživatelský kód s DiOS komunikuje pomocí systémových volání (syscallů)

DiVM

DiVM (interpret LLVM) poskytuje rozhraní na velmi nízké úrovni (odpovídá hypervizoru)

funkce (hyperally) pro:

- správu paměti (alokace, dealokace, změna a zjištění velikosti)
- nedeterministickou volbu
- anotaci hran grafu (akceptující hrany, trace)
- oznamování cyklů a práce s pamětí (bude vysvětleno)
- ovládání control flow (bude vysvětleno)

dále definuje:

- layout rámce a volací konvence
- start programu
- způsob předávání parametrů z příkazové řádky DIVINE do programu (DiOSu)

DiVM: oznamování cyklů a práce s pamětí

DIVINE provádí redukci stavového prostoru skrýváním instrukcí, které nepřistupují k paměti viditelné jinými vlákny

- je třeba aby DIVINE vědět, kdy dochází k čtení/zápisu z/do paměti
- může detekovat přímo DiVM – ale někdy lze staticky poznat, že manipulovaná paměť je privátní
- program musí oznámit DiVM kdy přistupuje k (potenciálně) viditelné paměti
- oznámení zajistí instrumentace

DiVM: oznamování cyklů a práce s pamětí

DIVINE provádí redukci stavového prostoru skrýváním instrukcí, které nepřistupují k paměti viditelné jinými vlákny

- je třeba aby DIVINE vědět, kdy dochází k čtení/zápisu z/do paměti
- může detekovat přímo DiVM – ale někdy lze staticky poznat, že manipulovaná paměť je privátní
- program musí oznámit DiVM kdy přistupuje k (potenciálně) viditelné paměti
- oznámení zajistí instrumentace

obdobně pro cykly v control flow

- třeba detekovat kvůli terminaci hledání následníka

DiVM: oznamování cyklů a práce s pamětí

DIVINE provádí redukci stavového prostoru skrýváním instrukcí, které nepřistupují k paměti viditelné jinými vlákny

- je třeba aby DIVINE vědět, kdy dochází k čtení/zápisu z/do paměti
- může detekovat přímo DiVM – ale někdy lze staticky poznat, že manipulovaná paměť je privátní
- program musí oznámit DiVM kdy přistupuje k (potenciálně) viditelné paměti
- oznámení zajistí instrumentace

obdobně pro cykly v control flow

- třeba detekovat kvůli terminaci hledání následníka

V návaznosti na tato oznámení DIVINE provádí interrupt aktuálního výpočtu a předání řízení do plánovače.

DiVM: ovládání control flow

DiOS (případně knihovny) musí být schopny:

- vytvářet a spravovat zásobníky
- nastavovat program counter
- zakazovat interrupt (vytvářet atomické sekce)
- nastavit, která funkce řeší plánování
- nastavovat handler chyb
- nastavovat globální proměnné a konstanty (kvůli procesům)

K tomu DiVM poskytuje sadu registrů a hypercall, který je umí modifikovat a číst.

Protipříklady

Protipříklad je posloupností fixovaných hodnot nedeterministických voleb

- k dispozici je interaktivní debugger
  - umožňuje krokovat
  - umožňuje inspekci programu v daném bodě výpočtu
  - podobně jako GDB, ale umí spolehlivě chodit i zpět