

Porovnání efektivity různých programovacích jazyků v numerických výpočtech

Jan Mucha

školitel:

Ing. Tomáš Oberhuber, Ph.D.

11.2.2011

Cíle bakalářské práce

- Porovnat programovací jazyky Fortran, C, C++, Java, Python na některých numerických algoritmech
- Zhodnotit efektivitu:
 - Jednotlivých jazyků na rozdílných platformách
 - Při překladu pomocí různých překladačů
 - Při různých stupních optimalizace

Kompilované jazyky

- Před spuštěním jsou nejprve kompletně přeloženy kompilátorem do instrukcí procesoru.
- Mohou být spouštěny samostatně, bez přítomnosti interpretu.
- Výsledkem je větší rychlost.
- Delší editační cyklus.

Interpretované jazyky

- Napsaný kód překládá do tzv. bajtového kódu.
- Tento kód pak provádí interpret jazyka.
- Obecně pomalejší, ale vysoce přenositelné.

Statická typová kontrola

- Typová kontrola probíhá již v době kompilace.
- Datový typ se přiřazuje k proměnné při její deklaraci.
- Vynechání typové kontroly za běhu umožňuje efektivnější vykonání programu a další optimalizace.

Dynamická typová kontrola

- Dynamická typová kontrola probíhá za běhu programu.
- Není požadována specifikace datového typu u proměnných.
- To zlepšuje flexibilitu a snadnost programování, ale ztěžuje optimalizaci programu na výkon.

Fortran

- Vznikl v roce 1957; ve své době revoluční.
- Kompilovaný jazyk, silné statické typování.
- Dodnes používaný ve vědecko-technických výpočtech – ohromné množství již implementovaných algoritmů.

C, C++

- Vznikl v roce 1972 resp. 1979.
- Kompilované programovací jazyky.
- Slabé statické typování.

Java

- Vznikl v roce 1995.
- Interpretovaný programovací jazyk.
- Silné statické typování.
- JIT - Just In Time kompilátor - v době provádění programu ho přeloží přímo do nativního strojového kódu počítače, na kterém je prováděn.
- Výrazně urychlí běh programu, ale způsobuje prodlevu při spouštění, kterou interpret stráví překladem do nativního kódu.

Python

- Vznikl v roce 1991.
- Interpretovaný programovací jazyk.
- Silné dynamické typování.

Numerický algoritmus

- Implementace explicitní Mersonovy varianty Rungeovy-Kuttovy metody.
- Pětikroková metoda 4. řádu s automatickou volbou časového kroku.
- Je řešena nelineární difúzní rovnici vedení tepla.

Pseudoanalytické řešení

- Implementace pseudoanalytického řešení téhož problému.
- Jde o řešení pomocí konvoluce.
- Umožní zkontrolovat správnost výsledků a ověřit řád konvergence Mersonovy metody.

Překladače

- GCC - GNU Compiler Collection
 - Sada překladačů vytvořených v rámci projektu GNU.
 - Zahrnuje širokou paletu jazyků včetně jazyka Fortran, C a C++.
 - Standardní překladače ve většině operačních systémů unixového typu.
- ICC a IFORT
 - ICC - Intel C++ Compiler, IFORT - Intel Fortran Compiler.
 - Překladače jazyků Fortran, C a C++ vyvinuté společností Intel.
 - Podporuje procesory s architekturou IA-32, Intel 64 a Itanium 2 a některé kompatibilní procesory od jiných společností.

Výsledky na HP xw9400

Algoritmus konvoluce na platformě HP xw9400

Programovací jazyk	bez	O1	O2	O3	xO
Fortran - ifc	7.185	3.573	3.012	2.950	3.012
C - gcc	9.152	6.527	6.711	6.293	-
C - icc	25.983	22.834	3.479	3.337	3.380
C++ - gcc	7.652	6.712	6.886	6.134	-
C++ - icc	25.810	23.245	3.397	3.507	3.262

Výsledky na HP xw9400

Mersonův algoritmus na platformě HP xw9400

Programovací jazyk	bez	O1	O2	O3	xO
Fortran - ifc	103.751	32.059	32.881	23.128	33.402
C - gcc	101.662	20.506	15.574	14.243	-
C - icc	67.265	21.978	21.937	21.596	21.649
C++ - gcc	101.771	20.652	16.919	14.542	-
C++ - icc	67.443	21.866	22.153	21.837	21.929

Výsledky na HP xw9400

Výsledky pro jazyk Java a Python na platformě HP xw9400

Programovací jazyk	CPU čas konvoluce	CPU čas Merson
Java	17.665	17.883
Python	318.954	4867.223

Výsledky na C8000 HP UX

Algoritmus konvoluce na platformě C8000 HP UX

Programovací jazyk	O1	O2	O3	xO	fast
Fortran - g77	16.989	14.248	14.131	-	-
Fortran - f90	17.510	13.110	8.510	22.900	-
C - gcc	39.530	39.110	37.050	-	-
C - cc	52.470	52.470	46.640	-	13.340
C++ - gcc	39.460	39.200	36.930	-	-

Děkuji za pozornost.