

# Praktické paralelní programování (PPP 2023)

## Počítačové cvičení č. 7: HDF5

---

Jiří Jaroš (jarosjir@fit.vutbr.cz)

### 1 ÚVOD

Cílem dnešního cvičení bude vyzkoušet si práci se vstupem a výstupem pomocí knihovny HDF5. Nejprve si vyzkoušíme vytvoření souboru z jednoho ranku a zápis datasetu. Následně se pokusíme zapsat matici distribuovanou po řádcích do souboru pomocí kolektivních komunikací. Na závěr načteme složitější soubor, který obsahuje dvě skupiny, jednu pro vstupní a jednu pro referenční data. Jednu matici natáhneme po řádcích, druhou po sloupcích. Volitelně můžete otestovat zápas atributů k souborům/datasetům (např. "Created by"), nebo nastavení chunkování či komprese.

### 2 PŘIHLÁŠENÍ NA KAROLÍNU / BARBORU A ALOKACE VÝPOČETNÍHO UZLU

Dnešní cvičení lze vykonat pouze na clusterech Karolína a Barbora, jelikož merlin nemá nainstalovanou paralelní verzi knihovny HDF5. Pokud používáte cluster Karolína:

1. Zažádejte o jeden uzel v interaktivním módu.

```
$ salloc -A DD-23-135 -p qcpu_exp -N 1 --ntasks-per-node 128 -t 01:00:00
```

2. Natáhněte modul s HDF5 a OpenMPI.

```
$ source load_modules.sh
```

Pokud používáte cluster Barbora:

1. Zažádejte o jeden uzel v interaktivním módu.

```
$ salloc -A DD-23-135 -p qcpu_exp -N 1 --ntasks-per-node 36 -t 01:00:00
```

2. Natáhněte modul s HDF5 a OpenMPI.

```
$ source load_modules.sh
```

Pokud pracujete na vlastním počítači, je paralelní HDF5 knihovnu možné nainstalovat pomocí:

- Linux

```
$ sudo apt-get install libhdf5-openmpi-dev hdf5-tools
```

- Mac OS

```
$ brew install hdf5-mpi
```

- Windows - je třeba zkompilevat ze zdrojového kódu.

## 2.1 PŘEKLAD

Vygenerujte překladový skript pomocí cmake a spusťte překlad:

```
$ cmake -Bbuild -S.  
$ cmake --build build
```

## 3 PŘÍKLAD 1. - SÉRIOVÉ I/O A ZÁPIS SKALÁRNÍ HODNOTY DO DATASETU.

Cílem tohoto příkladu je vyzkoušet si tvorbu a uzavření souboru (H5Fcreate, H5Fclose) a zápis z jednoho ranku do vybraného datasetu. Se souborem bude pracovat pouze root rank! Zadání se nachází pod sekcí case 1: ve funkci main.

1. Nejprve si deklarujte objekt HDF5 souboru (hid\_t).
2. Vytvořte soubor se jménem uloženým v konstantě fileName. Použijte takový flag, který zajistí přepis souboru, pokud již existuje.

3. Vytvořte `file`space a `mem`space popisující tvar skalární proměnné (velikost [1]).
4. Zapište hodnotu `value` do datasetu.
5. Uzavřete dataset a soubor.
6. Přeložte soubor.
7. Spust'te výslednou binárku a prohlédněte si obsah souboru:

```
$ mpiexec -np 8 ./hdf5 1
```

8. Prostudujte obsah souboru pomocí cmd rutin `h5ls` a `h5dump`. Vyzkoušejte si různé parametry.

#### 4 PŘÍKLAD 2. - ZÁPIS MATICE DO DATASETU POMOCÍ KOLEKTIVNÍHO I/O.

Mějme matici o velikosti  $16 \times 4$  prvky typu `int`. Tato matice je distribuovaná mezi ranky po blocích řádků. Vaším cílem je tuto matici zapsat do jednoho datasetu uvnitř HDF5 souboru. Zadání se nachází pod sekcí `case 2`: ve funkci `main`.

1. Nejprve si deklarujte objekt HDF5 souboru (`hid_t`).
2. Vytvořte file access property list a zapněte MPI-IO (`H5Pcreate`, `H5Pset_fapl_mpio`).
3. Vytvořte soubor se jménem uloženým v konstatně `fileName`. Použijte takový flag, který zajistí přepis souboru, pokud již existuje.
4. Vytvořte `file`space a `mem`space popisující celkovou velikost datasetu a jeho část v paměti daného ranku.
5. Pomocí hyperslabu vyberte část datasetu, do které budete zapisovat (`H5Sselect_hyperslab`).
6. Vytvořte XFER property list a zapněte kolektivní IO (`H5Pset_dxpl_mpio`).
7. Zapište hodnotu části matice do datasetu.
8. Uzavřete property listy, dataset a soubor.
9. Přeložte soubor.
10. Spust'te výslednou binárku a prohlédněte si obsah souboru:

```
$ mpiexec -np 8 ./hdf5 2
```

11. Prostudujte obsah souboru pomocí cmd rutin `h5ls` a `h5dump`. Vyzkoušejte si různé parametry.

## 5 PŘÍKLAD 3. - ČTENÍ NĚKOLIKA MATIC Z RŮZNÝCH DATASETŮ A SKUPIN.

Tento příklad slouží jako ukázka typického unit testu výpočetního kernelu. Naším cílem je otestovat správnost výpočtu Hadamardova násobení

$$C = A \cdot B'; \quad \forall i, j: c[i][j] = a[i][j] * b[j][i] \quad (5.1)$$

Ve vstupním souboru naleznete 2 skupiny "Inputs" a "Output". První skupina obsahuje matice A a B zatímco druhá skupina referenční hodnotu C. Všechny matice jsou obdélníkové s hranou o velikosti mocniny 2. Matice A a C je třeba distribuovat po blocích řádků, matice B po blocích sloupců.

Zadání se nachází pod sekci case 3: ve funkci main.

1. Nejprve otevřete soubor v režimu pouze pro čtení. Inspirujte se předchozím příkladem.
2. Nyní otevřete vstupní a výstupní HDF5 skupinu.
3. Následně otevřete vstupní datasety s maticemi A, B a Cref.
4. Nyní musíme zjistit velikost matic a spočítat jejich distribuci.
  - a) Nejprve si napíšeme pomocnou lambda funkci, která nám přečte počet dimenzí daného datasetu a uloží je do objektu typu Dim2D. Zde je třeba nejprve získat dataspace H5Dget\_space a následně vhodnou funkcí přečíst velikosti dimenzí H5Sget\_simple\_extent\_dims. Přepis výsledku do objektu Dim2D zajistíte konverzí struktury na pole Dim2D::toArray().
  - b) Následně použijte tuto lambda funkci na zjištění celkových velikostí daných matic.
  - c) Dle počtu ranků v komunikátoru spočtete lokální velikosti matic A, B a C.
5. Následně přečteme požadované části datasetů do připravených polí jednotlivých ranků. Pro tento účel si napíšeme novou lambda funkci, která má jako parametry: id datasetu, pozici slabu, velikost slabu a velikost celého datasetu.
6. Následně načteme dané datasety.
7. Nyní necháme proběhnout výpočet. Pokud byla data distribuována správně, měla by být maximální absolutní chyba rovna 0.
8. Na závěr uzavřete property listy, dataset a soubor.
9. Přeložte soubor v normálním nebo debug módu (vypíše obsah matic).
10. Spust' te výslednou binárku a prohlédněte si obsah souboru:

```
$ mpiexec -np 8 ./hdf5 3
```