Příloha č. 5 ZD

Výkonnostní testy

Úvod

Tato příloha ZD popisuje soubor výkonnostních testů. Software ALADIN bude použit v předepsaných konfiguracích k ověření schopnosti vysoce výkonného výpočetního serveru splnit mandatorní požadavky na výkon, tak jak jsou specifikovány ve SPEC_12, SPEC_16, SPEC_30, SPEC_39 a SPEC_40 v Příloze č. 3 této ZD.

Všechny výstupy z testů, včetně *stdout*, *stderr* a také všechny *job logs*, musí být poskytnuty ČHMÚ v elektronické podobě buďto jako jednotlivé soubory, anebo seskupené do souboru *tar*, na nosiči DVD. Binární výstupy (In Core Memory Spectral History soubory ICMSH*, Large Scale Coupling soubory ELSCF*, atd.) nejsou požadovány.
Výstup musí obsahovat:

- Název testu;
- Časové značky umožňující určení startu a konce reálného času potřebného pro exekuci úlohy;
- Množství času CPU použitého pro exekuci úlohy;
- Běh úlohy, včetně zdrojového kódu a vstupních souborů jako *fort.4*, aby ČHMÚ mohl zjistit jakékoliv změny buďto ve zdrojovém kódu nebo ve vstupních datech;
- Výstupy z testu.

Všechny testy musí být provedeny s pomocí spustitelného kódu, vytvořeného ze stejného setu knihoven.

Tentýž spustitelný kód musí být použit ve všech testech (specifické rekompilace pro provedení některého z testů NEJSOU povoleny). Spustitelný kód použitý v testech musí být generován s použitím překladačů a knihoven, které jsou nabízeny a podporovány jako odpověď na požadavky ZD.

Každý test musí být spuštěn dvakrát (aniž je stanoveno jinak) a poskytnout reprodukované, bit identické výsledky.

V každém testu je úhrnný reálný čas stanoven jako rozdíl časů, kdy první úloha testu započne a kdy poslední úloha testu skončí. Aby to bylo zřejmé, tyto body začátku a konce budou ve skriptu shell označeny pomocí volání nástroje „date“ s řetězci „start of test XX“ a „end of test XX“. Vstupní soubory před začátkem testu a výstupní soubory před koncem testu musí být na sdíleném diskovém systému.

Modifikace kódu nesmí změnit numerické výsledky natolik, že by byla ovlivněna jejich meteorologická kvalita. Spektrální diagnostika počítaná modelem ALADIN bude porovnána se standardním referenčním setem, generovaným na platformě s aritmetikou IEEE v plovoucí čárce, používající verzi kódu ALADIN dodanou s výkonnostními testy jako součást ZD.

Uchazeč dodá přehled výsledků výkonnostních testů podle vzoru Tabulky 1, jako součást zprávy o výkonnostních testech.

Instalace softwaru

Zdrojový kód a data nutná pro provedení výkonnostních testů jsou k dispozici ke stažení v ftp schránce ČHMÚ. Seznam souborů je zobrazen na Obrázku 1.

```
ICMSHALADINIT
ELSCFALADALBC000
ELSCFALADALBC001
ELSCFALADALBC002
ELSCFALADALBC003
ELSCFALADALBC004
ELSCFALADALBC005
ELSCFALADALBC006
ELSCFALADALBC007
ELSCFALADALBC008
Sourcesetc.tar:
        src/aladin43.tar
        srcadd/auxlibs_installer.2.3.tar
        srcadd/grib_api-1.10.4.tar.gz
        srcadd/lapack-3.6.1.tgz
        script/morgane
        script/fullpos
        namelist/fullpos.namel
        namelist/morgane.namel
        listing/morgane.refer.out
        listing/fullpos.refer.out
```

Obrázek 1. Seznam souborů pro výkonnostní testy.

Sestavení programu ALADIN

Tato část obsahuje podrobné informace o přípravě výkonnostních testů s modelem ALADIN. Zdrojový kód vychází z cyklu 43 knihoven ARPEGE/ALADIN.
Software ALADIN je strukturován do několika adresářů v závislosti na částech celého numerického systému. Navíc také knihovny grib_api, gribex a knihovny algebraických výpočtů LAPACK/BLAS jsou nutné pro sestavení spustitelného kódu.
Zdrojový kód knihoven se nachází v souborech *tar* v adresářích `src/` a `srcadd/`.

Kompilace

Knihovny obsahují subrutiny, funkce a moduly (*.F90) napsané ve volném formátu v jazyce Fortran 90 a „include“ soubory (*.h) obsahující namelisty, interface a další sdílené části. Knihovny jsou vysoce portabilní, a tak by neměla vzniknout potřeba měnit zdrojový kód kromě „inline“ direktiv pro kompilátor.

Arp – kód globálního modelu ARPEGE;
Ald – kód ALADIN, který je verzí ARPEGE na omezené oblasti;
Tf – kód spektrálních transformací v modelu ARPEGE;
Ta – kód spektrálních transformací v modelu ALADIN;
Sur – kód schématu pro zemský povrch;
Sat – kód nástrojů pro asimilaci družicových dat;
Bip – kód algoritmu bi-periodizace;
Cou – kód párování modelu na omezené oblasti;
Ecf, Odb, Mse, Mpa a Bla jsou z větší části potřebné kvůli „include“ souborům a modulům potřebným pro provedení kompilace. Knihovna Xla obsahuje nástroje lineární algebry a některé další matematické funkce.

Knihovna Xrd obsahuje technické funkce a subrutiny v jazyce C a Fortran. Některé z nich závisejí na systému a mohou být modifikovány uchazečem. Některé z těchto subrutin jsou ve volném formátu jazyka Fortran 90 (*.F90), některé mají fixní formát jazyka Fortran (*.F).

Rutiny ve fixním formátu jazyka Fortran (*.F) jsou též obsaženy v knihovně Xla. Oba typy by měly být normálně kompilovatelné stejným překladačem jazyka Fortran 2003.

Při kompilaci kódu je třeba brát zřetel na to, že rutiny v jazyce FORTRAN mohou používat moduly z jiných pod-knihoven, než kam samy patří, a také že existují moduly vnořené do modulů.

Pro sestavení spustitelné aplikace je třeba instalovat následující podpůrné knihovny, které jsou v adresáři `srcadd/`:
- *grib_api* verze 1.10.4;
- *lapack/blas* matematické knihovny;
- *gribex* (soubor auxlibs_installer.2.3.tgz).
Tyto knihovny jsou instalovány zvlášť.

Definice „kind“

ALADIN je napsán tak, že umožňuje explicitní definici kind pro proměnné. Numerické řešení vyžaduje vysokou přesnost operací v plovoucí čárce, které musí být z velké části provedeny ve dvojité přesnosti. V subrutinách modelu jsou proměnné deklarovány jako tiny (`KIND=JPRT`) pro `REAL`, (`KIND=JPIT`) pro `INTEGER`, small (`KIND=JPRS`), (`KIND=JPIS`), medium (`KIND=JPRM`), (`KIND=JPIM`), big (`KIND=JPRB`), (`KIND=JPIB`) nebo huge (`KIND=JPRH`), (`KIND=JPIH`). Viz soubory `Xrd/module/parkind1.F90` a `Xrd/module/parkind2.F90`, kde je příslušný kind těchto typů definován.
Ovšem z důvodu heterogenního obsahu některých knihoven (například Xrd , Xla, Mpa, Mse etc.) není zde tento systém plně implementován. Proto je nutné kompilovat příslušné Fortran položky s automatickým povýšením real proměnných na 8 bytů (REAL(KIND=8) nebo REAL*8), zatímco integer proměnné jsou ponechány tak, jak jsou deklarovány.

Linkování

Když jsou vytvořeny všechny knihovny, mohou být linkovány pomocí ld. Linkovací skript není poskytnut. Je důležité, aby knihovny byly linkovány ve správném pořadí pod-knihoven:

1. `Ald`
2. `Arp`
3. `Sat`
4. `Sur`
5. `Ta`
6. `Tf`
7. `Bip`
8. `Cou`
9. `Mpa`
10. `Mse`
11. `Xla`
12. `Xrd`
13. `Lapack/Blas (liblapack.a,librefblas.a)`
14. `Grib_api (libgrib_api_f90.a, libgrib_api.a)`
15. `Gribex (libgribex_370R64.a)`

Položky v knihovnách s nižším pořadím musí mít při linkování přednost před položkami v knihovnách s vyšším pořadím. Během linkování položka `master.o` musí být použita jako „main entry“ spustitelného programu.
Pro jednodušší portování kódu ALADIN pro výkonnostní testy nejsou některé knihovny modelu zahrnuty do zadávací dokumentace. Tyto obsahují subrutiny nebo funkce, které by stejně nebyly volány v požadovaných konfiguracích modelu. Uchazeč tyto funkce nahradí tzv. dummy verzemi s tiskem zprávy ‘function <name> should not be called’.

Spuštění programu ALADIN

Po vytvoření spustitelného programu `ALADIN.exe` lze spustit různé úlohy a konfigurace modelu. Příklad skriptů je poskytnut v adresáři `script/`po rozbalení souboru „`sourcesetc.tar`“. Konkrétní práce programu ALADIN.exe je definována parametry nastavenými v souboru namelist. Tento soubor namelist musí existovat v adresáři, ze kterého je program volán a musí mít jméno “`fort.4`”. Adresář `namelist/` obsahuje prototyp souboru namelist pro konfiguraci MORGANE a pro konfiguraci FULLPOS.
Vstupní data pro úlohu MORGANE jsou tvořena počáteční podmínkou integrační úlohy ICMSHALADINIT a okrajovými podmínkami (soubory ELSCF*).
Kód ALADIN umožňuje paralelní výpočet pomocí MPI a OpenMP. Nastavení pro paralelizaci pomocí MPI se nacházejí v namelist blocích `NAMPAR0` a `NAMPAR1`, viz níže.
Adresář `listing/` obsahuje referenční výstupní soubory.

Popis testů

Set běhů ASIS

Zdrojový kód nemůže být modifikován s výjimkou změn níže popsaných. Pokud si uchazeč přeje provést jakékoliv další změny, musí k tomu získat souhlas ČHMÚ, jinak bude jeho nabídka vyloučena. Následující kategorie změn v kódu jsou povoleny:

a) Zdrojový kód v jazyce FORTRAN může být zpracován jedním nebo více obecně dostupnými preprocesory, jejichž konečný výstup musí být použit jako vstup pro překladač jazyka FORTRAN.
b) Direktivy pro překladač mohou být umístěny do kódu za účelem řízení některých funkcí překladače, které by jinak překladač neprovedl, jako například "ignore vector dependencies", "unroll a DO-loop", "align arrays on different cache lines".
c) Je povoleno na úrovni překladače nebo preprocesoru provést tzv. "inline" rutin, a to pomocí direktiv nebo automaticky.
d) Je povoleno vkládat nebo měnit direktivy OpenMP.
e) Je povoleno optimalizovat subrutiny v adresáři `src/Xrd/support/` anebo nahradit jejich volání voláním knihoven dodaných uchazečem s tím, že tyto knihovny se stanou součástí podporovaného software.
f) Je povoleno optimalizovat knihovnu LAPACK/BLAS anebo ji nahradit knihovnami dodanými uchazečem s tím, že tyto knihovny se stanou součástí podporovaného software.

Všechny změny musí být dokumentovány a snadno identifikovatelné ve zdrojovém kódu, například pomocí „zakomentování“ původního kódu zařazením řetězce jako "!#xxx#" kde “xxx” je řetězec znaků snadno identifikující uchazeče (“atos”, “ibm”, “nec”, “sgi”, “cray”, atd.).
Následující parametry v souboru „fort.4“ mohou být modifikovány:

V namelistu NAMDIM:
NPROMA: faktor délky bloků pro cyklus výpočtů v uzlových bodech. Jeho hodnotu je potřeba zadat se záporným znaménkem, jinak bude přepsána default hodnotou.

V namelistu NAMPAR0:
NPROC: celkový počet MPI tasks použitých pro výpočet.
NPRGPNS, NPRGPEW, NPRTRW, NPRTRV: upřesnění distribuce počtu MPI tasks.
LOPT_SCALAR: optimalizace pro skalární počítače.

V namelistu NAMPAR1:
NSTRIN, NSTROUT: celkový počet MPI tasks použitých pro input and output processing.

Porovnání spektrálních norem

Za účelem ověření správnosti výsledků testů na zkoušeném vysoce výkonném výpočetním serveru je nutné srovnat obdržené spektrální normy s referenčními výstupy. Z důvodu vysoké citlivosti spektrálních norem na rozdíly v přesnosti operací v plovoucí čárce je v praxi obvykle nemožné přesně reprodukovat tyto normy na různých systémech.
Normy vypočítané pro několik meteorologických parametrů jsou vypsány ve výstupním listingu (NODE.001_01). Jsou počítány v každém pátém kroku a vypadají podle ukázky na Obrázku 2.

Odpovídající referenční výstup konfigurace MORGANE je v adresáři v listing/., soubor "morgane.refer.out".

Uchazeč porovná spektrální normy s referencí pro integrační kroky 0 až 40 s intervalem 5 kroků (Test MORGANE), a to pro 3 veličiny: „VORTICITY", „DIVERGENCE" a „d4 = VERT DIV + X". Výsledky jsou správné, pokud odchylka „VORTICITY" a „DIVERGENCE" od reference nepřesáhne jedno promile (1 ‰) a odchylka pole „d4 = VERT DIV + X" od reference nepřesáhne dvě promile (2 ‰) během ověřovaného intervalu prvních 40 časových kroků integrace modelu.

```
NORMS AT NSTEP CNT4 (PREDICTOR)    5
 SPECTRAL NORMS -   LOG(PREHYDS)    0.114979521644852E+02
 LEV     VORTICITY              DIVERGENCE
TEMPERATURE             HUMIDITY             KINETIC ENERGY
 AVE 0.459375177193074E-04 0.539773251492018E-04
0.263014820068540E+03 0.385130815676301E-02
0.778323684407708E+02
 LEV   LOG(PRE/PREHYD)       d4 = VERT DIV + X
 AVE 0.680041507485121E-05 0.368197585528870E-04
```

Obrázek 2. Ilustrace spektrálních norem ve výstupním listingu MORGANE

Test vytvoření absolutního binárního kódu

Hlavním účelem tohoto testu je ukázka schopnosti předzpracovat, kompilovat a sestavit absolutní binární kód modelu ALADIN. Za tímto účelem všechny kroky, jako jsou pre-

processing, kompilace, linkování atd. nezbytné pro vytvoření absolutního programu ze zdrojových souborů dodaných jako součást ZD pro provedení výkonnostních testů modelu ALADIN, musí být provedeny.
Uchazeč předá v elektronické podobě zdrojový kód použitý pro vytvoření absolutního binárního kódu včetně dokumentace změn (viz výše) a spolu s výstupy z kompilace (kompilační listingy, hlášení překladače). Tento zdrojový kód by byl použit pro akceptační zkoušky.
Uchazeč popíše přepínače použité při kompilaci a linkování (požadavek SPEC_38) a uvede seznam všech knihoven použitých pro sestavení absolutního binárního kódu.

Test Morgane

Test MORGANE provádí integraci modelu ALADIN na 24 hodin. Skript (příklad `script/morgane`) používá vstupní data `ICMSHALADINIT` (počáteční podmínka úlohy) a `ELSCFALADALBC0$NUM` (`NUM=00, 01, 02, …, 08;` okrajové podmínky úlohy) a soubor namelist `namelist/morgane.namel`.
Na výstupu je vytvořeno 25 souborů `ICMSHALAD+00$hh`, které obsahují průběžný stav v každé hodině předpovědi.
Uchazeč spustí jednu kopii, označenou jako copy0, testu MORGANE na Systému fáze A.
Uchazeč spustí jednu kopii, označenou jako copy0, testu MORGANE na Systému fáze B.
Uchazeč spustí současně 4 kopie testu MORGANE na Systému fáze A.
Uchazeč spustí současně 8 kopií testu MORGANE na Systému fáze B.
Všechny kopie testu MORGANE spuštěné v tomto testu musí mít identické spektrální normy.
Uchazeč porovná spektrální normy s poskytnutým referenčním výstupem pro ověření správnosti výsledků, viz oddíl „Porovnání spektrálních norem“.
Výsledné reálné časy (wall-clock times) budou vzaty jako míra výkonu Systému pro posouzení splnění mandatorního požadavku SPEC_12.
Výstupní listingy budou uloženy v `listing/morgane.perf.number_of_the_copy.phaseA(B).`
Uchazeč spustí krátkodobou 1h předpověď MORGANE (v namelistu se nastaví `CUSTOP='h1'`) se třemi různými hodnotami délky bloků `NPROMA`. Výstupní listingy budou uloženy v `listing/morgane1h.NPROMA_$`. Výsledky testu budou použity pro ověření požadavku SPEC_39. Tato část testu MORGANE nemusí být spuštěna dvakrát.
Uchazeč spustí krátkodobou 1h předpověď MORGANE (ve namelistu se nastaví `CUSTOP='h1'`) se třemi různými počty procesorů. Výstupní listingy budou uloženy v `listing/morgane1h.nproc_$`. Výsledky testu budou použity pro ověření požadavku SPEC_40. Tato část testu MORGANE nemusí být spuštěna dvakrát.

Test Fullpos

Test FULLPOS provádí post-processing výstupu modelu ALADIN pro délku předpovědi tři hodiny a slouží pro ověření výsledků. Skript (příklad `script/fullpos`) používá vstupní data spočtena v testu MORGANE, kdy se výsledný soubor ICMSHALAD+0003 (předpověď na tři hodiny) použije jako počáteční podmínka `ICMSHALADINIT` úlohy post-processingu.
Dalším vstupem je soubor namelist `namelist/fullpos.namel`.
Na výstupu je vytvořen soubor `PFALADMODL+0000`, který obsahuje výsledky post-processingu.
Uchazeč spustí test MORGANE (předpověď na 24h) na polovině výpočetních uzlů systému fáze A.

V okamžiku, kdy úloha vytvoří soubor ICMSHALAD+0003, uchazeč spustí test FULLPOS souběžně s testem MORGANE, který normálně pokračuje.
Uchazeč porovná výsledné průměrné (AVERAGE) normy „FULL-POS GPNORMS“ s referenčním výstupem (`listing/fullpos.refer.out`) pro modelovou hladinu číslo 87, viz Obrázek 3:

```
S087WIND_U_COMPO/MODL    : 0.7773019667227 53E+00
S087WIND_V_COMPO/MODL    : -.1915771094074 41E+00
S087TEMPERATURE /MODL    : 0.2929564828453 14E+03
S087GEOPOTENTIEL/MODL    : 0.2547520417735 00E+04
S087WIND_VELOCIT/MODL    : 0.2054830169306 42E+01
S087HUMI_RELATIV/MODL    : 0.7470958166956 14E+00
S087THETA_P_W   /MODL    : 0.2904836511330 16E+03
S087VIRT_P_TEMPE/MODL    : 0.3210672372313 76E+03
S087PRESSURE    /MODL    : 0.9842743490640 94E+05
```

Obrázek 3. Normy „FULL-POS GPNORMS v hladině číslo 87 ve výstupním listingu FULLPOS

Výsledky jsou správné, pokud relativní rozdíl norem v porovnání s referencí nepřesáhne hodnotu 4e-03 pro parametry proudění (`WIND_U_COMPO`, `WIND_V_COMPO` a `WIND_VELOCIT`), hodnotu 2e-06 pro parametry teploty (`TEMPERATURE`, `THETA_P_W` a `VIRT_P_TEMPE`), hodnotu 5e-05 pro parametr relativní vlhkosti (`HUMI_RELATIV`) a hodnotu 1e-06 pro parametry pole hmoty (`GEOPOTENTIEL` a `PRESSURE`).

Uchazeč zopakuje tento test stejným postupem pro fázi B. Výstupní listingy budou uloženy v `listing/fullpos_output.phaseA(B)`. Tento test není třeba opakovat dvakrát.

Test ověření paměti

Systém fáze B musí mít dostatek interní paměti na současné spuštění alespoň 20 kopií deterministické předpovědi MORGANE. Reálný (wall-clock) čas potřebný k výpočtu testu ověření paměti musí být menší nebo roven 20/8-násobku reálného času potřebného pro současný výpočet 8 kopií testu MORGANE.
Uchazeč spustí současně 20 kopií testu MORGANE na Systému fáze B.
Test ověření paměti nemusí být spouštěn dvakrát. Zato všechny kopie testu MORGANE počítané v těchto testech musí mít identické spektrální normy jako individuální výkonnostní testy MORGANE.
Výstupní listingy budou uloženy v `listing/morgane.memory.number_of_the_copy.phaseB`.
Výsledky budou vzaty jako míra výkonu Systému pro posouzení splnění mandatorního požadavku SPEC_16.

Test operativního přepnutí SWITCHOVER

Při tomto testu bude spuštěna jedna předpověď modelu ALADIN (úloha MORGANE) s normální prioritou, která bude používat všechna procesorová jádra na všech Početních nódech Systému HPCS; když model dosáhne 12 hodin předpovědi (“času modelu”), musí být spuštěna druhá předpověď modelu ALADIN, která musí běžet s nejvyšší prioritou a také používat všechna procesorová jádra na všech Početních nódech Systému HPCS. Tím se simuluje vysokoprioritní operativní úloha, která se počítá na úkor úlohy s normální prioritou.

Po skončení vysokoprioritní úlohy první úloha s normální prioritou získá zpět zdroje systému a bude pokračovat až do dokončení. Celkový reálný čas pro tyto 2 výpočty v paralelním spuštění by neměl překročit součet reálných časů při samostatném individuálním spuštění za sebou. Reálný čas vysokoprioritní úlohy nesmí přesáhnout více jak o 5% reálný čas výkonnostního testu jedné kopie MORGANE.
Uchazeč musí uvést, jakým způsobem tohoto bylo dosaženo, zejména jak byla úloha s normální prioritou ošetřena (suspended, swapped out nebo check-pointed), anebo zda bylo postačující využít mechanismu priorit. Ukončení (abort) úlohy s normální prioritou a její následné spuštění po úloze s vysokou prioritou za účelem splnění tohoto testu není pro ČHMÚ přijatelné.
Výsledky budou vzaty jako míra výkonu Systému pro posouzení splnění mandatorního požadavku SPEC_30.

Výsledky

Tabulka 1 **Formulář pro výsledky testů.**

TEST FÁZE A						
Test	Název	Počet současně spuštěných kopií	Reálný (wall-clock) čas nutný pro výpočet daného počtu kopií	Počet procesorových jader a MPI tasks v 1 kopii: NCORES/NPROC	Maximum paměti na nód	Celková paměť
1	MORGANE	1				
1R	Repeat	1				
2	MORGANE	4				
2R	Repeat	4				
3	SWITCHOVER	Vysoká priorita				
		Normální priorita				
		Celkově 2 úlohy		X		
3R	Repeat	Vysoká priorita				
		Normální priorita				
		Celkově 2 úlohy		X		
TEST FÁZE B						
Test	Název	Počet současně spuštěných kopií	Reálný (wall-clock) čas nutný pro výpočet daného počtu kopií	Počet procesorových jader a MPI tasks v 1 kopii: NCORES/NPROC	Maximum paměti na nód	Celková paměť
1	MORGANE	1				
1R	repeat	1				
2	MORGANE	8				
2R	repeat	8				
3	MORGANE memory	20				
4	SWITCHOVER	Vysoká priorita				
		Normální priorita				
		Celkově 2 úlohy		X		
4R	repeat	Vysoká priorita				
		Normální priorita				
		Celkově 2 úlohy		X		