

## Paralelizování a Fortran

### Co bude na pořadu?

- (náš) hardware a software vhodný pro intenzivní počítání
- (automaticky) paralelizující (volně dostupné) překladače Fortranu
- (automaticky) paralelizující knihovna MKL
- (ruční) paralelizace pomocí knihovny OpenMP – později
- (ruční) paralelizace pomocí knihovny MPI – později

### HW/SW pro intenzivní počítání

- náš hardware: stroje s vícejádrovými (32bitovými) procesory s 64bitovými prvky (adresováním)
  - 4jádrové procesory (CPU) Intel Pentium Core 2 Quad 2.4 GHz Q6600, RAM 8 GB
  - (hw) schopnost adresovat byty 64bitově, tj. paměť až  $2^{64}$  bytů (pozn.:  $2^{32}$  B =  $2^{22}$  KB =  $2^{12}$  MB =  $2^2$  GB)
  - (hw) schopnost zpracovat integer čísla o velikosti 4 a 8 B a real čísla o velikosti 4 a 8 (ne 16) B (viz IEEE 754r)
- procesory Xeon, Core 2, Dual Core: Intel 64 Architecture neboli EM64T (Extended Memory Technology)
  - <http://en.wikipedia.org/wiki/Em64t>
  - pozn. IA-32 Architecture: 32bitové procesory bez 64bitových rozšíření – Intel Pentium 4
  - pozn. IA-64 Architecture: 64bitové procesory (nekompatibilní s EM64T) – Intel Itanium
- podporované instrukční sady: MMX (Multimedia/Multiple Math/Matrix Math Extensions), SSE (Streaming SIMD=simple-instruction-multiple-data Extensions, 70 instrukcí), SSE2 (+144 instrukcí), SSE3 (+13 instrukcí) a SSSE3 (Supplemental SSE)
  - pozn. SSE4 na (budoucím) CPU Intel Pentium Core 2 Quad Q9xxx
- software: 64bitový operační systém (Linux, Win XP 64b, Vista 64b), 64bitový překladač, 64bitové knihovny
- my teď: (EM64T) vaclav, geof30–40 (Mandriva Linux 2008, Intel Fortran, gfortran, MKL, OpenMP, MPI)
  - (IA-32) karel, geof60–90 (starší Mandriva/Mandrake, Intel Fortran, gfortran, MKL)
  - hw výkon: je-li 1 jádro geof30 100 %, pak naše stroje EM64T 400 % a IA-32 67 %
  - sw verze: Intel Fortran 10.1, MKL 10.0
  - výhledy: geof50 s EM64T, gfortran s MKL, ifort pro Windows, rozhraní BLAS95 a LAPACK95
  - sledování procesorové zátěže a diskového prostoru: guptime (geof-uptime), df (disk free), top (top tasks)

### Paralelizační modely

- vektorizace: instruction-level parallelism (ILP) – využívání SIMD instrukčních sad procesoru, realizuje překladač, užívá se ve vnitřních cyklech
- multithreadový multitasking: thread-level parallelism (TLP) – jeden proces s více vlákny a sdílenou pamětí, užívá se na vnějších cyklech, realizuje překladač automaticky nebo programátor ručně pomocí OpenMP
- multiprocessorový multitasking: procesorové cluster – více procesů s distribuovanou (oddělenou) pamětí, realizuje programátor ručně pomocí MPI
- paralelizované knihovny numerických metod: BLAS/LAPACK, MKL, IMSL, NAG, ...
- Fortran 90+: data parallelism – operace a standardní funkce zobecněné pro práci s poli, dot\_product a matmul, paralelizovatelné cykly forall a where, paralelizovatelné pure a elemental procedury; praxe :-(

### Paralelizující a volně dostupné překladače Fortranu 95

- Intel Fortran Compiler ifort: volný pro Linux, dostupný pro Windows a Mac OS
- gfortran: součást GNU Compiler Collection (GCC), volný pro Linux, Windows a cokoliv
- oba překladače pro 32 i 64bitové prostředí
- oba obsahují paralelizační knihovnu OpenMP
- ifort spolupracuje s knihovnou MKL, gfortran používá knihovnu BLAS (a může spolupracovat s MKL?)
- volný překladač g95, <http://g95.org>, neparalelizuje

## Intel Fortran Compiler (ifort)

- Fortran 95 s prvky Fortranu 2003, obsahuje OpenMP, spolupracuje s MKL
- pro operační systémy: Linux – zdarma, Windows – měsíční demo (ale: vyžaduje min. Win XP a MS Visual C++)
- stručná pomoc a dokumentace:
  - `ifort -help`
  - `man ifort`
  - `/opt/intel/fce/*/doc`
  - <http://www.intel.com> (Software Products, Compilers)

### Vybrané volby překladače

- default překlad: se standardní optimalizací a výstupem do a.out
  - `ifort files.f90` neboli `ifort -O2 -o a.out files.f90`
- překlad s žádnou nebo malou optimalizací
  - `ifort -O0 files.f90` nebo `-O1`
- překlad s maximální optimalizací (včetně flush-to-zero a interprocedural optimizations)
  - `ifort -O3 -ftz -ip files.f90` nebo `-ipo`
- s využitím instrukčních sad MMX, SSE, SSE2, SSE3 a SSSE3
  - `ifort -xT files.f90` (pro Core 2) nebo `-xP` jen do SSE3 (Pentium 4), `-xS` i pro SSE4 (CPU Q9xxx)
- překlad s automatickou paralelizací pomocí OpenMP (vyžaduje -O2 nebo -O3)
  - `ifort -O3 -parallel files.f90`
- další možnosti:
  - `-C, -CB, -CU` kontroly: runtime error checking, out-of-bounds subscripts, uninitialized variables
  - `-i2, -i4, -i8, -r8, -r16` default podtyp pro integer a real/double precision; neúčinné pro popisy real(kind)
  - `-zero, -save` lokální proměnné: automatická inicializace nulou, uchování
  - `-mp, -mp1` zvýšení floating-point přesnosti, snížení optimalizace (větší, menší)
  - `-openmp, -openmp-stubs` připojení knihovny OpenMP v paralelním nebo sériovém režimu
  - `-openmp -parallel` automatická paralelizace zdrojového textu mimo direktivy OpenMP
  - `-cm, -w` potlačení všech komentářů (comment messages) a varování (warnings)
  - `-vec-report0` minimalizace (`-vec-report3` maximalizace) vektorizační diagnostiky (loop was vectorized)
  - `-par-reportN, -openmp-reportN, -opt-reportN` diagnostika (N = 0, 1, ...) paralelizace, OpenMP a optimalizace

### Příkladové skripty

- pro EM64T (vaclav, geof30, geof40)
  - `ifort -O3 -ftz -ip -xT -parallel -vec-report0 -o a.out files.f90`
  - pro OpenMP vložit `-openmp`
  - pro MKL přidat `-L$MKL -Bstatic -lmkl_lapack -Bdynamic -lmkl_em64t -lguide -lpthread`
- pro IA-32 (karel, geof60, geof70, geof80, geof90)
  - `ifort -O3 -ftz -ip -xP -vec-report0 -o a.out files.f90`
  - a pro MKL přidat `-L$MKL -Bstatic -lmkl_lapack -Bdynamic -lmkl_ia32 -lguide -lpthread`

### Direktivy překladače

- vektorizace: direktivy IVDEP, VECTOR ALWAYS, NOVECTOR aj.
- automatická paralelizace: direktivy (!DEC\$) PARALLEL, NOPARALLEL

## gfortran

- Fortran 95 s prvky Fortranu 2003 a Fortranu 2008, pro 32 i 64 bitové systémy, volně dostupný
- součást GNU Compiler Collection (GCC), <http://gcc.gnu.org/wiki/GFortran>
- obsahuje OpenMP, spolupracuje s MKL, automaticky používá BLAS
- pro operační systémy: Linux – zdarma, Windows – měsíční demo (ale: vyžaduje min. Win XP a MS Visual C++)
- stručná pomoc a :
  - `gfortran --help`
  - `man gfortran`
- překladový skript (včetně OpenMP):
  - `gfortran -O3 -fopenmp -o a.out files.f90`

## Intel Math Kernel Library (MKL)

- knihovna frekventovaných algoritmů optimalizovaných a paralelizovaných na procesorech Intel
- pro operační systémy: Linux – zdarma, Windows – měsíční demo
- pro překladače: Intel Fortran, Intel C++, GNU Compiler Collection (GCC, součástí i gfortran)
- obsah:
  - BLAS**, sparse BLAS, CBLAS (Basic Linear Algebra Subprograms)
  - LAPACK** (Linear Algebra Package, soustavy lineárních algebraických rovnic, vlastní čísla a vektory)
  - ScaLAPACK** (Scalable LAPACK, spolupracuje s MPI)
  - Pardiso** (Parallel Sparse Direct Solver, pro přímé nebo iterační řešení velkých řídkých soustav)
  - FFT/DFT** (Fast and Discrete Fourier Transform)
  - VML** (Vector Math Library, pro efektivní vyčíslování funkcí s vektorovým argumentem)
  - VSL** (Vector Statistical Library, mj. pro náhodná čísla)
  - GMP** (GNU Multiple Precision Arithmetic Library, pro integer aritmetiku neomezené přesnosti)
  - intervalová aritmetika**
- dokumentace: `/opt/intel/mkl/*/doc`
  - `userguide.pdf` pro User's Guide (107 stran)
  - `mklqref/index.htm` pro Quick Reference (k verzi 9.0)
  - `mklman.pdf` pro velký Reference Manual (3238 stran)
  - `vmlnotes.htm` pro Vector Math Library Notes
  - `vslnotes.pdf` pro Vector Statistical Library Notes
  - `Release_Notes.htm` pro Release Notes
- připojení MKL k Intel Fortranu (pro Linux):
 

```
ifort files.f90 -L$MKL [-lmkl_solver] [-lmkl_lapack] -lmkl_em64t -lguide -lpthread
```

  - kde `mkl_solver` Pardiso
  - `mkl_lapack` LAPACK
  - `mkl_em64t` BLAS, sparse BLAS, FFT/DFT, VML, VSL, GMP, interval arithmetics (mkl\_ia32: totéž pro IA-32, mkl\_ipf: totéž pro IA-64)
  - `mkl_lapack95`, `mkl_blas95` **f95 rozhraní** pro LAPACK a BLAS (standardně neinstalováno)
  - `guide`, `pthread` threading libraries
  - \$MKL** obsahuje cestu ke knihovním souborům, př. `/opt/intel/mkl/10.0.1.014/lib/em64t`
  - a lze volit statické (**-Bstatic**) nebo dynamické (**-Bdynamic**) připojení knihoven (příklad na předchozí stránce)
- připojení MKL k Intel Fortranu (pro Windows):
  - ...
- připojení MKL ke gfortranu (pro Linux i Windows):
  - ...

### MKL a BLAS: maticové násobení

- podprogram **dgemm** pro výpočet výrazu  $c = \alpha * op(a) * op(b) + \beta * c$ , kde  $a$ ,  $b$ ,  $c$  jsou po řadě  $m \times k$ ,  $k \times n$  a  $m \times n$  matice a operátor  $op$  může matice  $a$ ,  $b$  transponovat
  - integer  $m$ ,  $n$ ,  $k$ ,  $lda$ ,  $ldb$ ,  $ldc$
  - real(8)  $a(lda,*)$ ,  $b(ldb,*)$ ,  $c(ldc,*)$ ,  $\alpha$ ,  $\beta$
  - character(1) ::  $transa='N'$ ,  $transb='N'$  ! možnost 'T' pro transpozici matic
  - call `dgemm(transa, transb, m, n, k, alpha, a, lda, b, ldb, beta, c, ldc)`

- výkonový **benchmark** na Core 2 (2.4 GHz) **bez paralelizace** (viz zdrojový text `benchLA.f90`)

Linux (64bit), Intel Fortran Compiler

CPU time [s]	n = 2500	n = 5000	n = 10000
MKL/BLAS (dgemm)	3	28	223
f90 matmul	122	1290	moc
f77 3 vnořené cykly	122	1282	moc

Windows 2000 (32bit), Microsoft Fortran PowerStation 4.0

CPU time [s]	n = 2500	n = 5000	n = 10000
IMSL/BLAS (dgemm)	77	516	4153
f90 matmul	113	1126	moc
f77 3 vnořené cykly	100	967	moc

**MKL a LAPACK: řešení soustavy algebraických rovnic s plnou maticí pomocí LU rozkladu**

- driver **dgesv** pro řešení soustavy o n rovnicích a nrhs pravých stranách,  $a x = b$   
integer n, nrhs, lda, ldb, ipiv(n), info  
real(8) a(lda,\*), b(ldb,\*)  
call dgesv(n, nrhs, a, lda, ipiv, b, ldb, info)
- podprogramy **dgetrf**, **dgetrs** pro dekompozici mxn matice a zpětnou substituci pro nrhs pravých stran  
character(1) :: trans='N' ! možnost 'T' pro transpozici matice  
call dgetrf(m, n, a, lda, ipiv, info)  
call dgetrs(trans, n, nrhs, a, lda, ipiv, b, ldb, info)

- výkonový **benchmark** na Core 2 (2.4 GHz) **bez paralelizace** (viz zdrojový text benchLA.f90)

Linux (64bit), Intel Fortran Compiler

CPU time [s]	n = 2500	n = 5000	n = 10000
MKL/LAPACK (dgesv nebo dgetrf & dgetrs)	1	10	78
Numerical Recipes (ludcmp & lubksb)	40	428	moc

Windows 2000 (32bit), Microsoft Fortran PowerStation 4.0

CPU time [s]	n = 2500	n = 5000	n = 10000
IMSL (dlftrg & dlfsrg)	9	79	575
Numerical Recipes (ludcmp & lubksb)	37	372	moc

**Paralelizace v MKL**

- založena na (interním) volání knihovny OpenMP (BLAS, LAPACK, Pardiso, FFT, VML)
- proměnné prostředí **MKL\_NUM\_THREADS** pro návrh počtu paralelních vláken (dříve jen **OMP\_NUM\_THREADS**)  
**MKL\_DYNAMIC** pro volné nakládání knihovny s navrženým počtem vláken  
**MKL\_DOMAIN\_NUM\_THREADS** pro nastavení počtu vláken různě pro různé kapitoly
- př. bash: export MKL\_NUM\_THREADS=1  
tsh: setenv MKL\_NUM\_THREADS 1
- **default:** MKL\_NUM\_THREADS = počet procesorů (nově ve verzi 10, dříve až na výjimky = 1)  
MKL\_DYNAMIC = TRUE

- **benchmark paralelizace v MKL** na Core 2 Quad (2.4 GHz, 4 jádra), Intel Fortran Compiler for Linux

Maticové násobení, **dgemv**, n=10000, verze knihovny 10 (v závorkách verze 9)

MKL NUM THREADS	CPU time [s]	wall clock time [s]	poznámka
1	223 (226)	224 (226)	
2	223 (226)	113 (116)	
3	224 (240)	115 (118)	žádný přínos!
<b>4</b>	<b>230 (233)</b>	<b>60 (63)</b>	minimum
8	233 (238)	60 (64)	neproduktivní
64	258 (266)	68 (72)	kontraproduktivní

Řešení soustavy algebraických rovnic s plnou maticí, **dgesv**, n=10000, verze 10 (v závorkách verze 9)

MKL NUM THREADS	CPU time [s]	wall clock time [s]	poznámka
1	78 (81)	79 (81)	
2	95 (84)	43 (45)	
3	91 (86)	31 (33)	
<b>4</b>	<b>88 (91)</b>	<b>25 (27)</b>	minimum
8	88 (92)	25 (28)	neproduktivní
64	99 (137)	28 (66)	kontraproduktivní