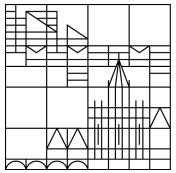
Universität Konstanz



XDS news March 2017

Kay Diederichs, HDRMX@Lund, 16.3.2017

Native reading of data files using a dynamically (i.e. at runtime) loaded library

(this is old news; for more information see talks at previous HDRMX meetings)

- A generic interface for this purpose was implemented
- Currently, it is only used by the Dectris HDF5 Plug-In
- Removes the overhead of H5ToXds (or similar programs)
- Can be used to read arbitrary files, if the Producer provides a library
- XDSwiki will soon show source code of a minimal example (in Fortran)

XDS on Xeon Phi “Knights Landing” (KNL)

- KNL usage depends on environment variables, and these may influence several overlapping areas and override each other. There are several such “families”, e.g. **GOMP**_CPU_AFFINITY, **OMP**_PLACES, **OMP**_PROC_BIND, **KMP**_SETTINGS, **KMP**_AFFINITY, **KMP**_PLACE_THREADS
- "Intel Xeon Phi Processor High Performance Programming, Knights Landing Edition” by Reinders, Jeffers and Sodani is useful. Usage of MPI+OpenMP seems well documented, but not scheduling of concurrent multi-core processes (XDS!) on KNL. Overall, confusing and inconsistent documentation.
- **use `OMP_PROC_BIND=false` (or `KMP_AFFINITY=verbose,none`, or explicit `numactl`)**
- KNL benefits from latest versions of BIOS, operating system (CentOS7.3), compiler (ifort 17)
- quadrant mode, with MCDRAM configured as cache, gives a good baseline for performance tuning.

9/2016: Xeon Phi “Knights Landing” (KNL)

- 64, 68 or 72 cores in a single socket; 4-fold (efficient!) Hyper-Threading; ≤ 384 GB ECC-DDR4; 16 GB built-in high-bandwidth (5 times faster) MCDRAM
- Boots and runs Linux (e.g. RHEL7) natively; has many new features e.g. AVX512, a vector instruction w/ 512 bits (AVX=256 bits; SSE=128 bits); needs recompilation for highest performance
- Vector peak performance: >3 TeraFlops Double Precision; >6 TeraFlops SP

System tried: 1 KNL 7210 processor; 64*4=256 threads; 192 GB RAM; 750W power supply; (e.g. workstation SuperServer 5038K-i ~6.000€). Note A: until now, systems reaching 3 TeraFlops cost >60.000€ (8 * Xeon E7). Note B: SuperServer 5028TK-HTR: 2U-rackmount; 4 KNL 7210 (>1.000 threads): ~20.000€

Robust findings with Eiger data on KNL

- KNL needs some tuning (last slide). It is a pain that by default all processes land on core 0 !
- As expected and hoped for, processing is fastest with Dectris' plugin library (talk by Stefan et al.). Roughly 15% faster than w/ H5ToXds-script, and twice as fast as pure H5ToXds (w/ my setup).
- Experiments and results are documented in Eiger article of XDSwiki. A single KNL gives a similar overall processing time as a dual-processor Xeon workstation, but may be cheaper.
- CBF is still 10% faster than HDF5 but there is a tradeoff in CPU versus I/O - CBF has bigger files but needs less CPU. Therefore, if reading data only once, HDF5 is faster than CBF.

9/2016: First findings on KNL

Test data: 3600 Eiger 16M frames; HDF5 (using `H5ToXds`)

- XDS runs "out of the box"; re-compiled for <=72 threads (old: <=32 threads)
- Compiling (ifort v16 or v17 beta) with `-xMIC-AVX512` : ~30% faster
- Using MCDRAM instead of normal RAM in "Flat" mode: 5% faster in COLSPOT; >10% faster in INTEGRATE (`memkind` not yet tried; "Cache" and "Hybrid" mode not yet tried);
- NUMA configuration & environment variables need special attention
- Work in progress; system looks promising but too early to summarize

References: http://colfaxresearch.com/get-ready-for-intel-knights-landing-3-papers/

Current version of XDS

- INIT parallelization speeds up as expected
- latest BUILT 20170215 does not correctly construct name of master file; use symlink as workaround `ln -s my_master.h5 my_data_000001.h5` (should be obvious from error message). Next BUILT will fix this.
- special versions (e.g. compiled with -xMIC-AVX512 for KNL) can be obtained from me

9/2016: Parallelization

- COLSPOT: reads several degrees of data; for speedup parallelized on two levels (threads + shell-level) i.e. can use many cores / several computers
- INTEGRATE: as COLSPOT but reads all data
- INIT: reads (typically) only the first 5 degrees of data. Used to be **the only serial part of XDS data processing**. For big machines/clusters this meant that up to 1/3 of wallclock time was spent in INIT.

INIT was rewritten; it is now parallel (OpenMP threads) - the speedup depends on the number of threads. For typical 180° data sets of which 5° are used for INIT, the wallclock time spent in INIT can now be considered insignificant.

XDS: managing bottlenecks

has effect on *increasing*	CPU	I/O	Memory
MAXIMUM_NUMBER_OF_JOBS	-	-	++
NUMBER_OF_IMAGES_IN_CACHE	-	--	++
DELPHI	-	o	++ (if using NUMBER_OF_IMAGES_IN_CACHE, else o)
1/OSCILLATION_RANGE	++	++	++ (if using NUMBER_OF_IMAGES_IN_CACHE, else o)

++: raises strongly +: raises o: no change -: reduces --: reduces strongly

Personal perspective on best HDRMX practices

- Exposure time: if data quality is unimportant (screening), speed of data collection may be high - but for structure solution and refinement, **data quality is the most important factor** → minimize fluctuations!
- Default experiments should use 0.1° oscillation range (except very low-mosaicity crystals, e.g. viruses). Don't go overboard - just because it is *possible* to do 0.01° oscillation range, does not mean it is *necessary/good*.
- **best way to obtain accurate (not just precise) data: "true multiplicity"**
- flux variations on short timescale are mitigated by attenuation/long exposure
- monitoring data quality goes a long way towards good data quality
- beamline staff should know about precision *vs* accuracy, and establish best compromises; some users may need education in aspects of data quality.

Thank you!

questions?