Hartree Centre
Science & Technology Facilities Council

Mixed precision:
Is it the Holy Grail for Software Efficiency?

Luke Mason, Andrew D. Taylor, Sue Thorne
sue.thorne@stfc.ac.uk

Software Outlook

CoSeC
Computational Science Centre for Research Communities

Single vs Double Precision

- Most scientific codes are written using double precision floating point arithmetic
- Single precision
  - less data to be stored and moved
  - single precision arithmetic (usually) faster and uses less energy than double precision

Single vs Double Precision

- Jacobi Test Code: Developed as part of CCP-ASEArch
- 3D Jacobi solver benchmark: MPI, OpenMP, CUDA, OpenCL and OpenACC
- Stencil approach:
  - Each entry in vector v_i computed by averaging the values in v_{i-1} at neighbouring grid points

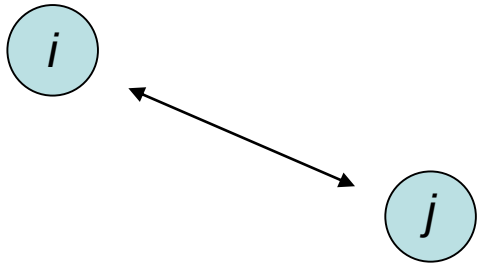
		Single		Double	
	Nthr	Time(s)	Energy(kJ)	Time(s)	Energy(kJ)
ARMv8.1 - Cavium ThunderX	48	266.1	69.8	482.3	128.2
	96	169.7	48.3	324.8	93.7
Intel (IvyBridge) Xeon ES-2650v2	8	110.0	6.62	233.4	14.2
	16	97.4	7.36	235.0	17.3
Blue Gene/Q – IBM PowerPC	1	3333.7	181.3	2707.1	144.4
	8	419.4	23.0	428.5	23.8
	16	239.0	13.1	421.2	22.8

OpenMP version. Energy per node

Single vs Double Precision

- Most scientific codes are written using double precision floating point arithmetic
- Single precision
  - less data to be stored and moved
  - single precision arithmetic (usually) faster and uses less energy than double precision
- Underlying problem may be much lower accuracy than double precision
- Might be possible to use mixed precision within code

DL_POLY – molecular dynamics code CCP5



Initialise problem

Calculate forces $F(t=0)$

For *it=1,…,nTimeSteps* **do**

Calculate velocity $v(t+\frac{\nabla t}{2})$ and update positions $r(t+\nabla t)$

Calculate long-range contributions to force $F(t+\nabla t)$

For *i=1,…,nAtoms* **do**

For *j* in neighbours of *i* **do**

Update system energy

Update $F(t+\nabla t)$ with short-range contributions to forces on *j* due to *i*

End for

End for

Calculate velocity $v(t+\nabla t)$

End for

Hartree Centre

Science & Technology Facilities Council

DL_POLY – molecular dynamics code CCP5

- Ionic interaction energies described by Coulomb sum
  - Replaced by Ewald sum

$$E = E^L + E^S + E^{SELF}$$

    - E^L long-range interactions → uses discrete (complex) Fourier transform
    - E^S short-range interactions → uses **erfc** function
    - E^{SELF} self interaction term → simple summation

- Force on ion *j* (position $\boldsymbol{r}_j$) due to ionic interaction energies

$$f_j = -\frac{\nabla E}{\nabla \boldsymbol{r}_j} = -\frac{\nabla E^L}{\nabla \boldsymbol{r}_j} - \frac{\nabla E^S}{\nabla \boldsymbol{r}_j}$$

Ewald Sum - Long-range interactions

- Discrete Fourier transform in 1D:

  $Y_k = \sum_{n=0}^{N-1} w_N^{k,n} y_n$, where $w_N^{k,n} = e^{-i2\pi kn/N}$

  - Initialisation
    - $w_N^{k,n}$ computed in dp but stored in sp
  - Calculation
    - Input: y_n computed in dp, converted to sp
    - Output: Y_n converted to dp before use in energy/forces calculation

Ewald Sum - Long-range interactions

		fft			total			
Problem	n_p	**mixed**	**dp**	**ratio**	**mixed**	**dp**	**ratio**	**fft/total**
02b	8	*5.10*	6.77	0.75	*49.9*	51.7	0.97	0.13
	16	*2.75*	4.13	0.67	*26.4*	27.8	0.95	0.15
	32	*1.38*	2.30	0.60	*13.8*	14.8	0.93	0.16
02c	48	*27.7*	46.1	0.60	*174.1*	192.8	0.90	0.24
	96	*14.6*	29.1	0.50	*91.8*	106.6	0.86	0.27
	144	*9.8*	17.2	0.57	*62.7*	70.2	0.89	0.25
04b	8	*1.00*	1.25	0.80	60.4	60.6	1.00	0.021
	16	*0.49*	0.84	0.58	32.5	32.7	0.99	0.026
	32	*0.24*	0.44	0.55	16.8	16.9	0.99	0.026
04c	96	*3.06*	5.32	0.58	*107.2*	109.2	0.98	0.049
	144	*2.37*	4.27	0.56	*72.3*	73.8	0.98	0.058
	192	*1.34*	2.60	0.52	*53.4*	54.4	0.98	0.048

Ewald Sum - Long-range interactions

		fft			total			
Problem	n_p	**mixed**	**dp**	**ratio**	**mixed**	**dp**	**ratio**	**fft/total**
02b	8	*5.10*	6.77	0.75	*49.9*	51.7	0.97	0.13
	16	*2.75*	4.13	0.67	*26.4*	27.8	0.95	0.15
	32	*1.38*	2.30	0.60	*13.8*	14.8	0.93	0.16
02c	48	*27.7*	46.1	0.60	*174.1*	192.8	0.90	0.24
	96	*14.6*	29.1	0.50	*91.8*	106.6	0.86	0.27
	144	*9.8*	17.2	0.57	*62.7*	70.2	0.89	0.25
04b	8	*1.00*	1.25	0.80	60.4	60.6	1.00	0.021
	16	*0.49*	0.84	0.58	32.5	32.7	0.99	0.026
	32	*0.24*	0.44	0.55	16.8	16.9	0.99	0.026
04c	96	*3.06*	5.32	0.58	*107.2*	109.2	0.98	0.049
	144	*2.37*	4.27	0.56	*72.3*	73.8	0.98	0.058
	192	*1.34*	2.60	0.52	*53.4*	54.4	0.98	0.048

Ewald Sum - Long-range interactions

		fft			total			
Problem	n_p	**mixed**	**dp**	**ratio**	**mixed**	**dp**	**ratio**	**fft/total**
02b	8	*5.10*	6.77	0.75	*49.9*	51.7	0.97	0.13
	16	*2.75*	4.13	0.67	*26.4*	27.8	0.95	0.15
	32	*1.38*	2.30	0.60	*13.8*	14.8	0.93	0.16
02c	48	*27.7*	46.1	0.60	*174.1*	192.8	0.90	0.24
	96	*14.6*	29.1	0.50	*91.8*	106.6	0.86	0.27
	144	*9.8*	17.2	0.57	*62.7*	70.2	0.89	0.25
04b	8	*1.00*	1.25	0.80	60.4	60.6	1.00	0.021
	16	*0.49*	0.84	0.58	32.5	32.7	0.99	0.026
	32	*0.24*	0.44	0.55	16.8	16.9	0.99	0.026
04c	96	*3.06*	5.32	0.58	*107.2*	109.2	0.98	0.049
	144	*2.37*	4.27	0.56	*72.3*	73.8	0.98	0.058
	192	*1.34*	2.60	0.52	*53.4*	54.4	0.98	0.048

Ewald Sum - Short-range interactions

$$E^S = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{\mathbf{n}}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}\operatorname{erfc}\left(\frac{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}{\sqrt{2}\sigma}\right)$$

Known precisely

Known at uniform points - interpolate

Hartree Centre
Science & Technology Facilities Council

Ewald Sum - Short-range interactions

$$E^S = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{\mathbf{n}}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}\,\mathrm{erfc}\left(\frac{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}{\sqrt{2}\sigma}\right)$$

Summation simplified to just nearby neighbours of i

Known precisely

Known at uniform points - interpolate

Hartree Centre
Science & Technology Facilities Council

Ewald Sum - Short-range interactions

$$E^S = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{\mathbf{n}}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}\,\mathrm{erfc}\left(\frac{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}{\sqrt{2}\sigma}\right)$$

Summation simplified to just nearby neighbours of i

Known precisely

Known at uniform points - interpolate

Convert q_i, q_j and $|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|$ to single precision

Store erfc at uniform points in single precision

Ewald Sum - Short-range interactions

$$E^S = \frac{1}{4\pi\epsilon_0}\frac{1}{2}\sum_{\mathbf{n}}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{q_i q_j}{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}\,\mathrm{erfc}\left(\frac{|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|}{\sqrt{2}\sigma}\right)$$

Summation simplified to just nearby neighbours of i

Known precisely

Known at uniform points - interpolate

Convert q_i, q_j and $|\mathbf{r}_i - \mathbf{r}_j + \mathbf{n}L|$ to single precision

Store erfc at uniform points in single precision

Convert each term in summation back to double precision and add to E^S

Similarly for the force calculation $\frac{\nabla E^S}{\nabla \mathbf{r}_j}$

Ewald Sum - Short-range interactions

		Ewald_real			total			
Problem	n_p	**mixed**	**dp**	**ratio**	**mixed**	**dp**	**ratio**	**Ewald_real/total**
02b	1	65.2	*55.3*	1.18	325.6	*317.6*	1.025	0.17
	2	32.2	*27.4*	1.17	169.2	*162.4*	1.042	0.17
	4	16.7	*14.2*	1.18	85.4	*83.0*	1.029	0.17
	8	8.39	*7.29*	1.15	44.7	*43.3*	1.031	0.17

Ewald Sum - Short-range interactions

		Ewald_real			total			
Problem	n_p	**mixed**	**dp**	**ratio**	**mixed**	**dp**	**ratio**	**Ewald_real/total**
02b	1	65.2	*55.3*	1.18	325.6	*317.6*	1.025	0.17
	2	32.2	*27.4*	1.17	169.2	*162.4*	1.042	0.17
	4	16.7	*14.2*	1.18	85.4	*83.0*	1.029	0.17
	8	8.39	*7.29*	1.15	44.7	*43.3*	1.031	0.17

- Forces update – after update calculation, result immediately converted to dp and added
- $6 \times nTimeSteps \times nAtoms^2$ conversions from sp to dp (worst case)

- Keep force updates in sp array and add to dp force array at end of each time iteration
  - Requires restructuring of code
  - $3 \times nTimeSteps \times nAtoms$ conversions from sp to dp
  - $3 \times nTimeSteps$ extra array allocations in current restructured code

Ewald Sum - Short-range interactions

		Ewald_real			total		
Problem	n_p	**mixed**	**dp**	**ratio**	**mixed**	**dp**	**ratio**
02b*	1	*49.1*	50.3	0.98	337.8	*328.5*	1.03
	2	*29.5*	29.7	0.99	202.5	*201.0*	1.01
	8	*7.41*	7.43	0.99	53.7	*52.2*	1.03

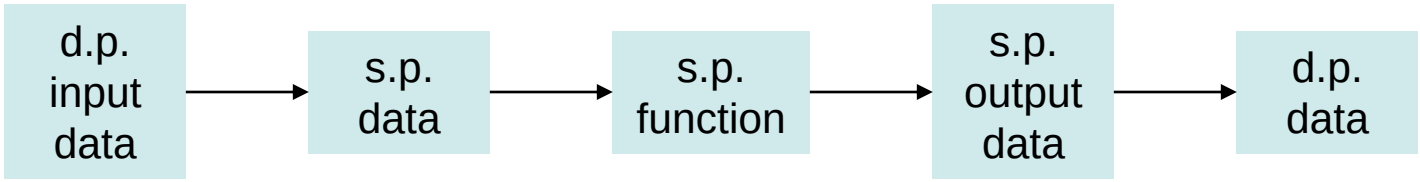
- Restructured code
- Small gains in short-range interaction calculation
- Overheads of extra allocations and adding the single precision update back to the forces at the end of each time step outweigh the gains
  - Significant change to the fundamental structure of DL_POLY required to reduce these overheads

Other energy contributions..

- Many contributions to overall energy of the system
  - Intermolecular potential functions
    - van der Waals, metal, Tersoff,…

- Van der Waals (short-range)
  - 12-6 potential: $\frac{A}{(r_i - r_j)^{12}} - \frac{B}{(r_i - r_j)^{6}}$
  - Lennard-Jones potential: $4\epsilon \left[\left(\frac{\sigma}{r_i - r_j} \right)^{12} - \left(\frac{\sigma}{r_i - r_j} \right)^{6} \right]$
  - …
  - Accumulating related forces in single precision update array not possible due to large numerical range of entries being added together

Conclusions



- **The good…**
  - Single precision component sufficient for overall accuracy
  - Large amount of data being operated on by the function
  - Function only called a few times
  - Fairly simple to implement
- **The bad…**
  - Single precision not sufficient
- **The ugly…**
  - Small amount of data being operated on by the function
  - Function called many times within a loop
  - Major restructure of code may be needed to utilize possible gains e.g. use s.p. update arrays