
# The Need for Parallel I/O in Classical Molecular Dynamics

I.T. Todorov
I.J. Bush, A.R. Porter

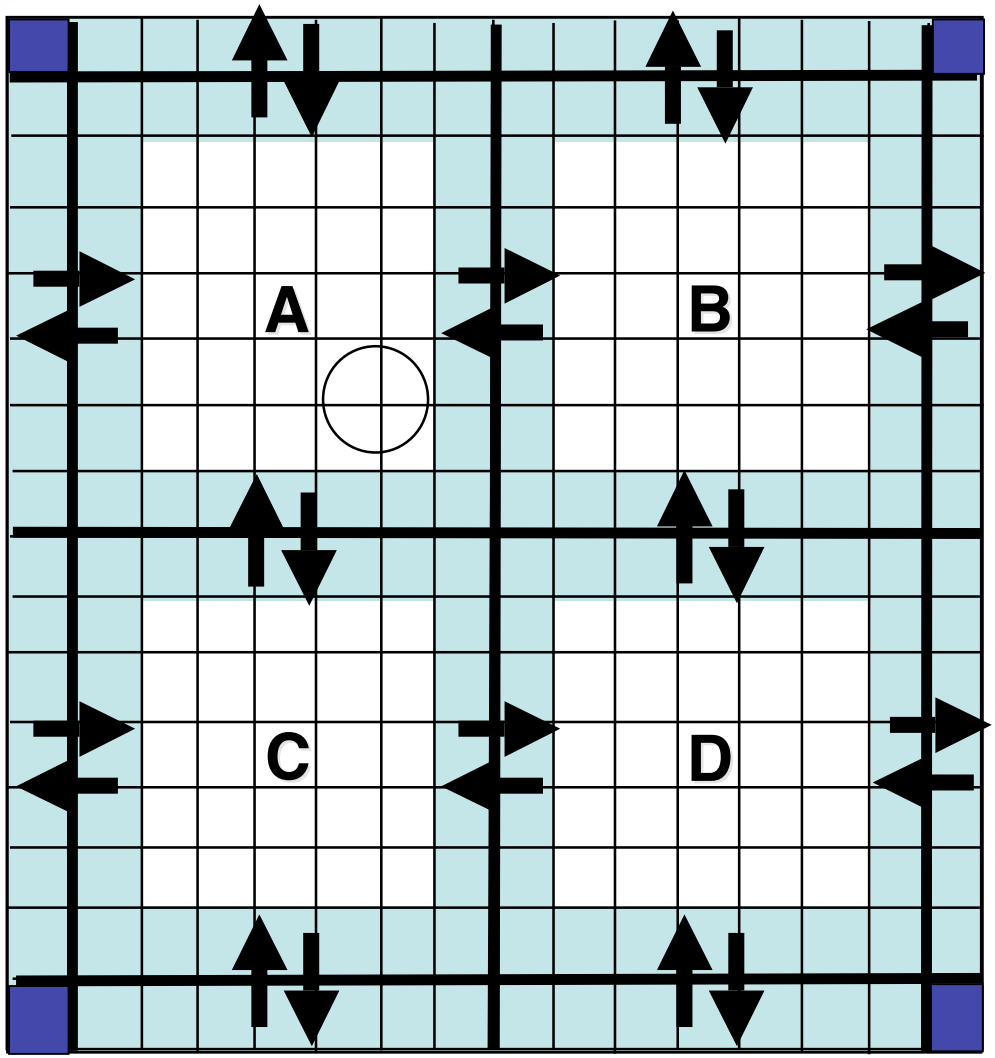
Advanced Research Computing, CSE Department
STFC Daresbury Laboratory, Warrington WA4 4AD, UK

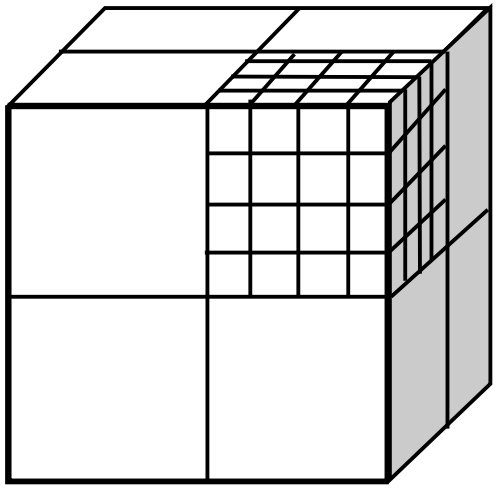

# The DL_POLY_3 MD Package

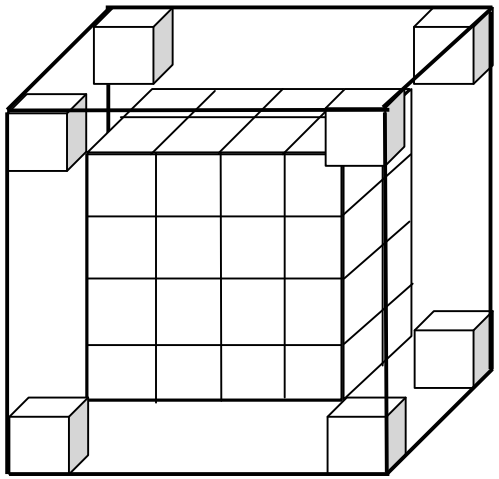
- General purpose MD simulation package
- Written in modularised free formatted FORTRAN90 (+MPI2) with rigorous code syntax (FORCHECK and NAGWare verified) and no external library dependencies
- Generic parallelisation (for short-ranged interactions) based on spatial domain decomposition (DD) and linked cells (LC)
- Long-ranged Coulomb interactions are handled by SPM Ewald employing 3D FFTs for k-space evaluation - limited use to $2^k$ CPUs
- Maximum particle load ≈ $2.1 \times 10^9$ atoms
- Full force field and molecular description but no rigid body description yet (as in DL_POLY_2)
- Free format semantically approached reading with some fail-safe features (but fully fool-proofed)

Science & Technology Facilities Council

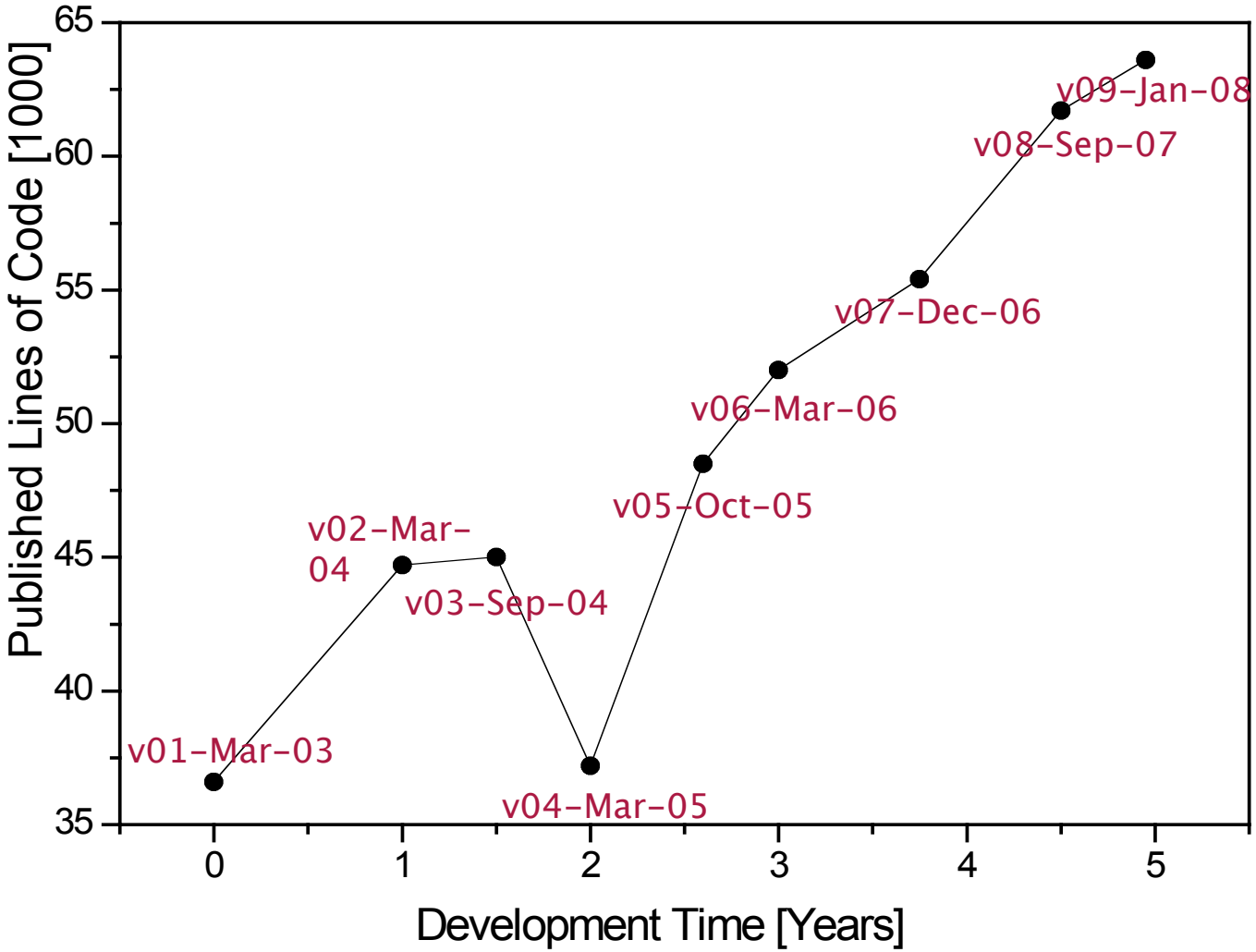
# Domain Decomposition Parallelisation



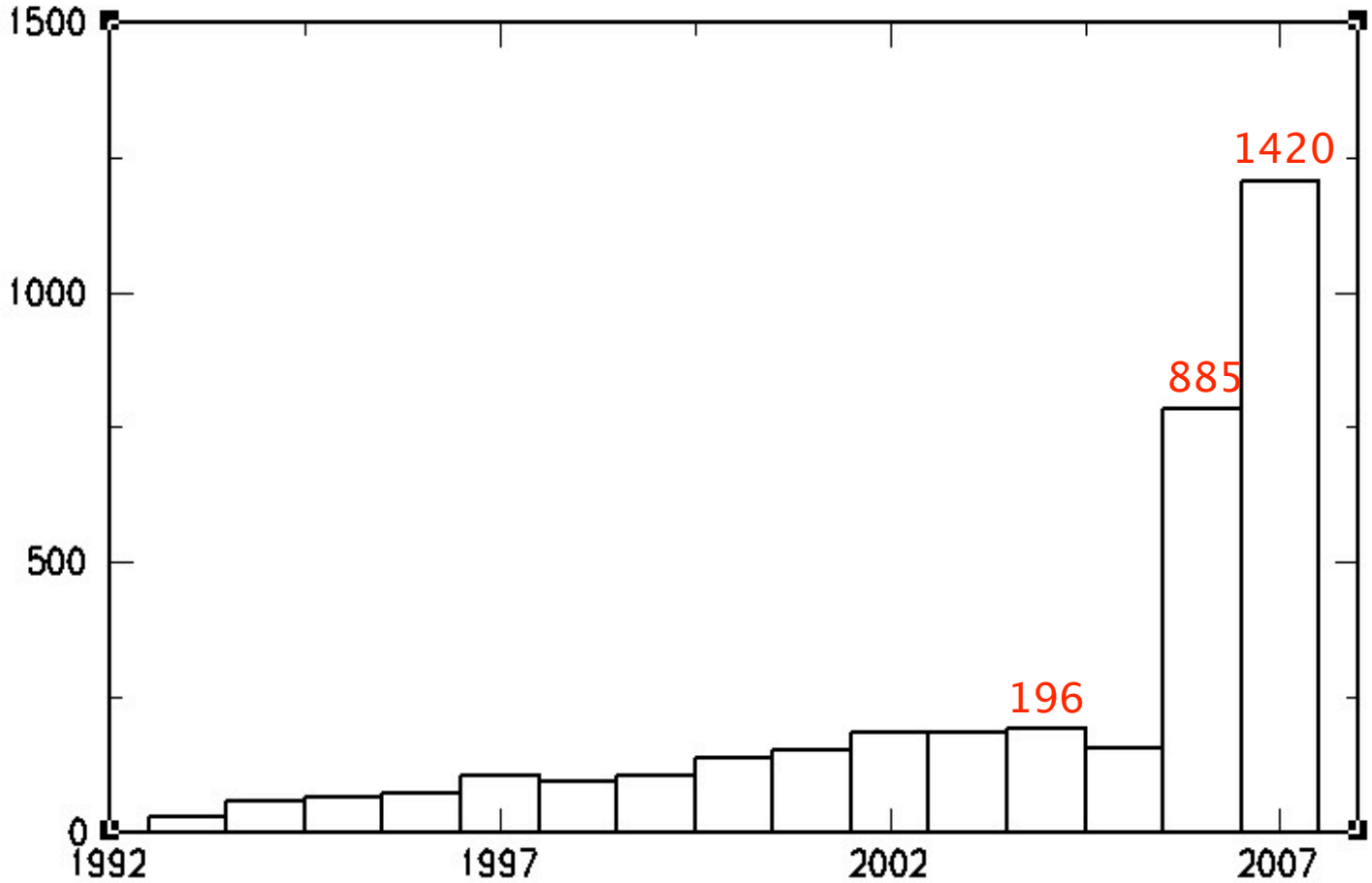


# Development



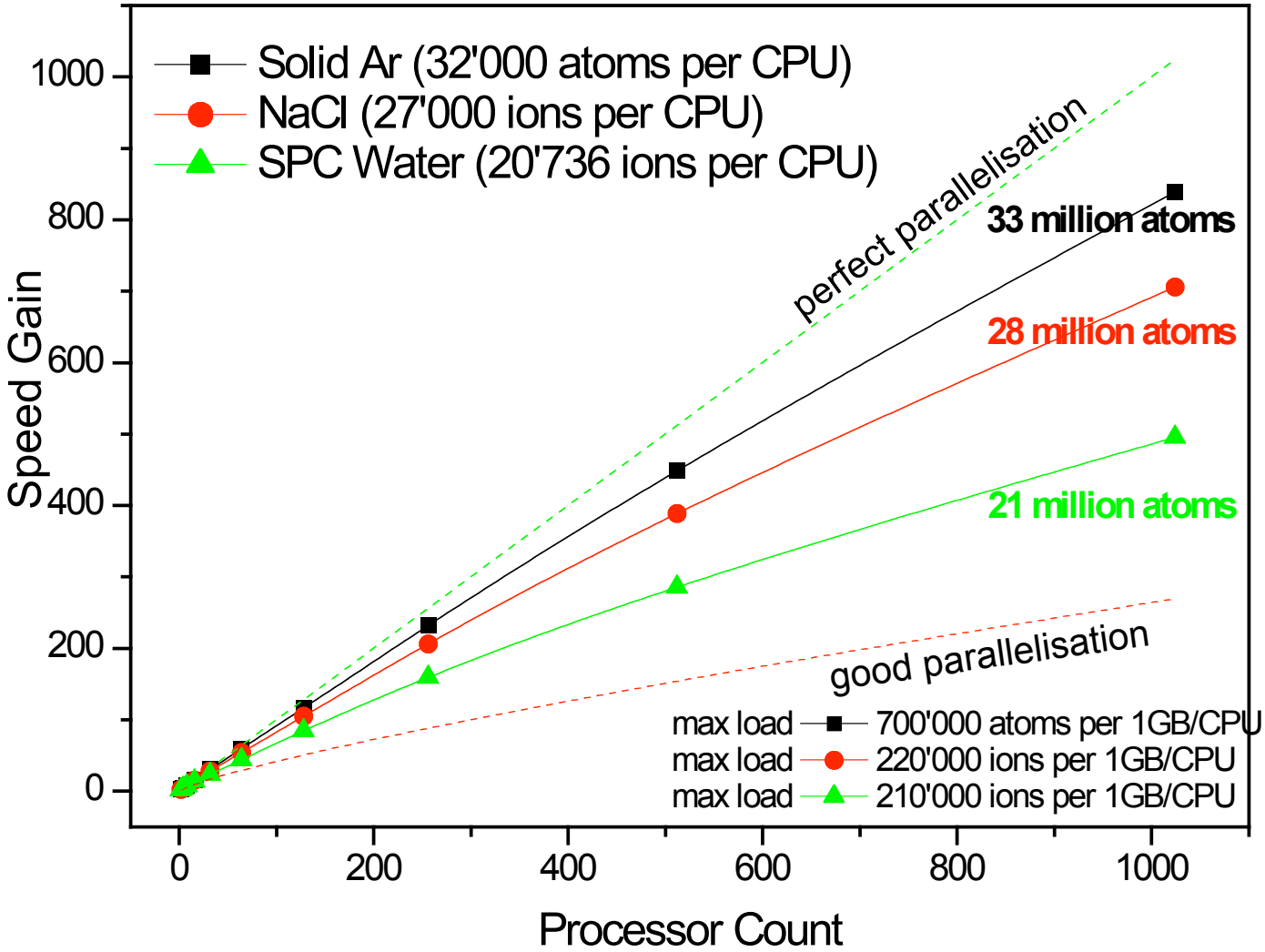
Science & Technology Facilities Council

# DL_POLY Licence Statistics



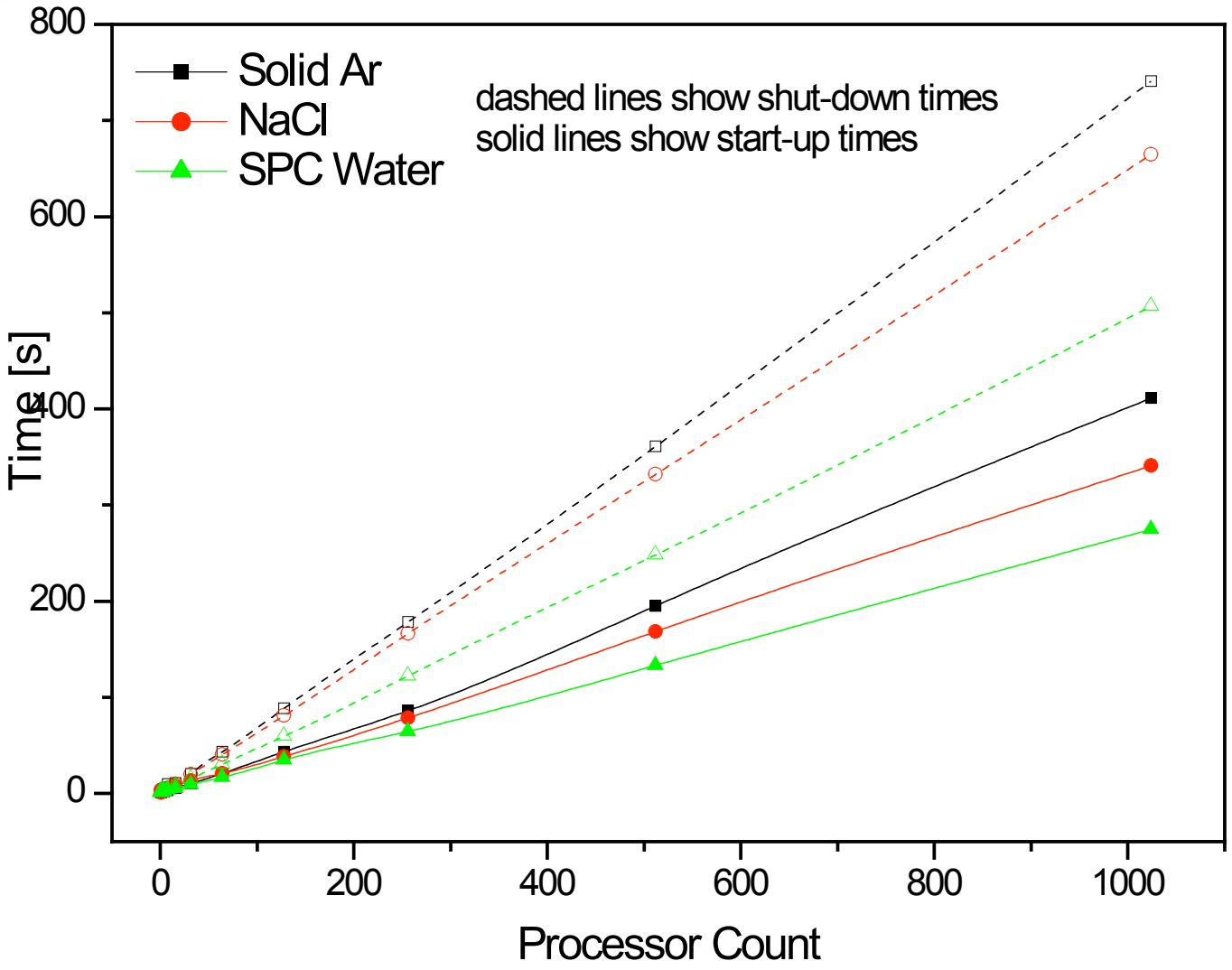
Science & Technology Facilities Council

# Performance Weak Scaling on IBM p575



Science & Technology Facilities Council

# I/O Weak Scaling on IBM p575




# Proof of Concept on IBM p575

300,763,000 NaCl with full SPME electrostatics evaluation on 1024 CPU cores

| | |
|---|---|
| Start-up time | ≈ 1 hour |
| Timestep time | ≈ 68 seconds |
| FFT evaluation | ≈ 55 seconds |

In theory ,the system can be seen by the eye. Although you would need a very good microscope – the MD cell size for this system is 2μm along the side and as the wavelength of the visible light is 0.5μm so it should be theoretically possible.

# Importance of I/O - I

Types of MD studies most dependent on I/O

- Large length-scales ($10^9$ particles), short time-scale such as screw deformations
- Medium big length-scales ($10^6$–$10^8$ particles), medium time-scale (ps-ns) such as radiation damage cascades
- Medium length-scale ($10^5$–$10^6$ particles), long time-scale (ns-$\mu$s) such as membrane and protein processes

| Types of I/O: | portable | human readable | loss of precision | size |
|---|---|---|---|---|
| • ASCII | + | + | – | – |
| • Binary | – | – | + | + |
| • XDR Binary | + | – | + | + |


# Importance of I/O - II

**Example**: 15 million system simulated with 2048 MPI tasks

MD time per timestep ~0.7 (2.7) seconds on Cray XT4 (BG/L)

Configuration read ~100 seconds (once during the simulation)

Configuration write ~600 seconds for 1.1 GB with the fastest I/O method – MPI-I/O for Cray XT4 (parallel direct access for BG/L)

**I/O in native binary is only 3 times faster and 3 times smaller**

Some unpopular solutions

- Saving only the important fragments of the configuration
- Saving only fragments that have moved more than a given distance between two consecutive dumps
- Distributed dump – separated configuration in separate files for each MPI task (CFD)

Science & Technology Facilities Council

# ASCII I/O Solutions in DL_POLY_3

1. **Serial direct access write** (abbreviated as SDAW) – where only a single node, the master, prints it all and all the rest communicate information to a master in turn while the master completes writing a configuration of the time evolution.

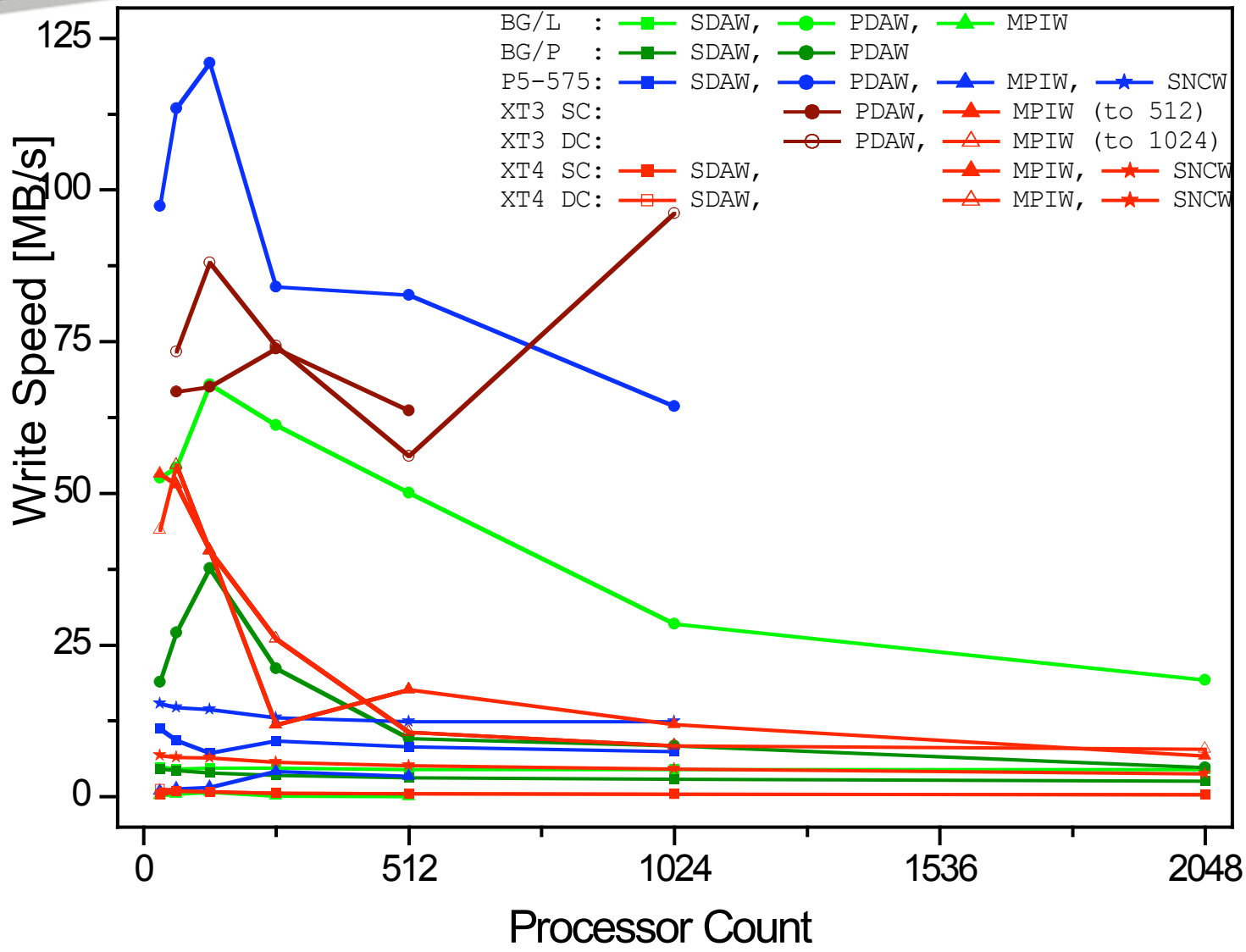
2. **Parallel direct access write** (PDAW) – where all nodes print in the same file in an orderly manner so no overlapping occurs using Fortran direct access files. However, it should be noted that the behaviour of this method is not defined by the Fortran standard, and in particular we have experienced problems when disk cache is not coherent with the memory.

3. **MPI-I/O write** (MPIW) which has the same concept as the PDAW but is performed using MPI-I/O rather than direct access.

4. **Serial NetCDF write** (SNCW) using NetCDF libraries for machine-independent data formats of array-based, scientific data (widely used by various scientific communities)

Science & Technology Facilities Council

# Overall I/O Performance

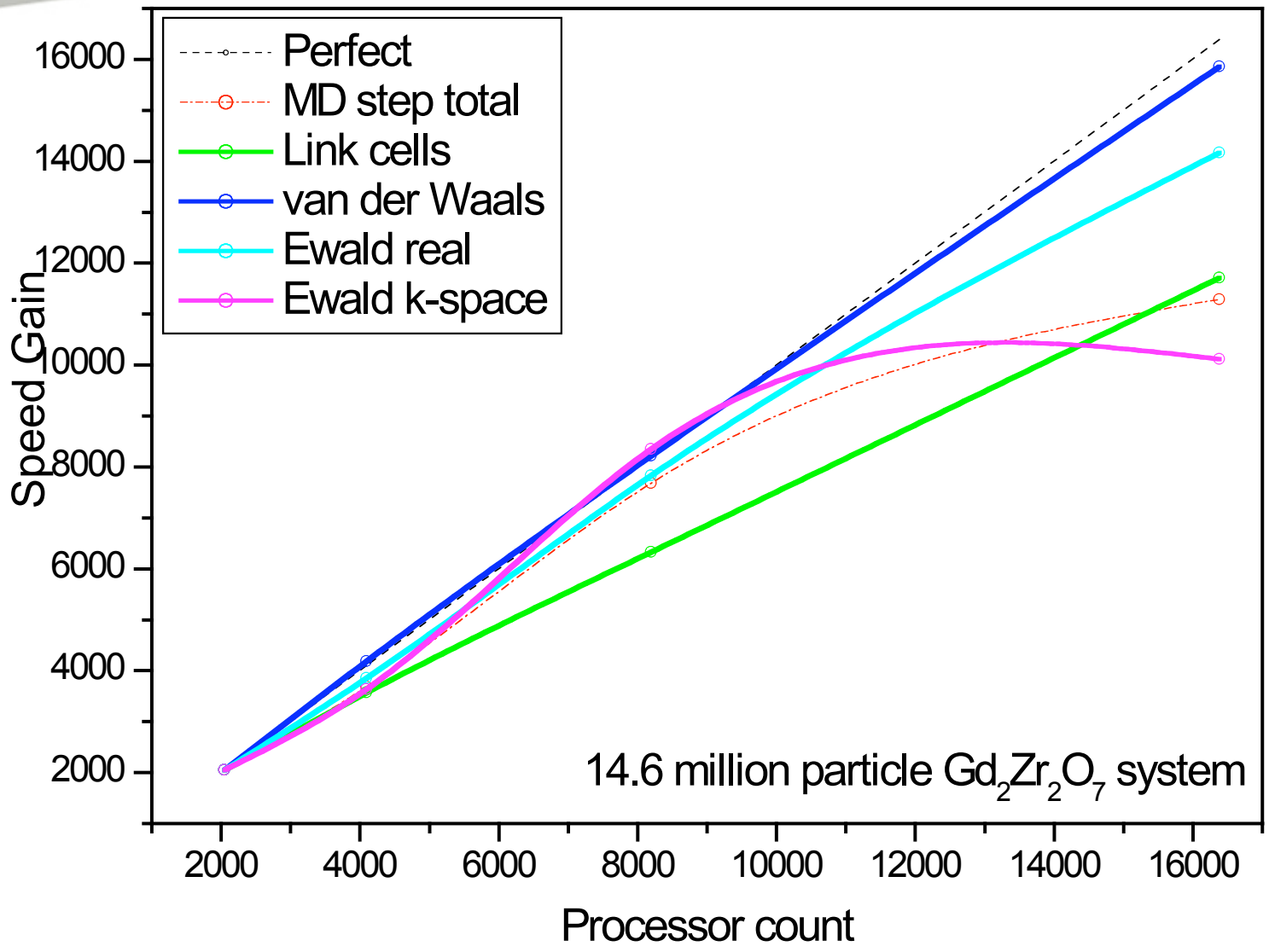


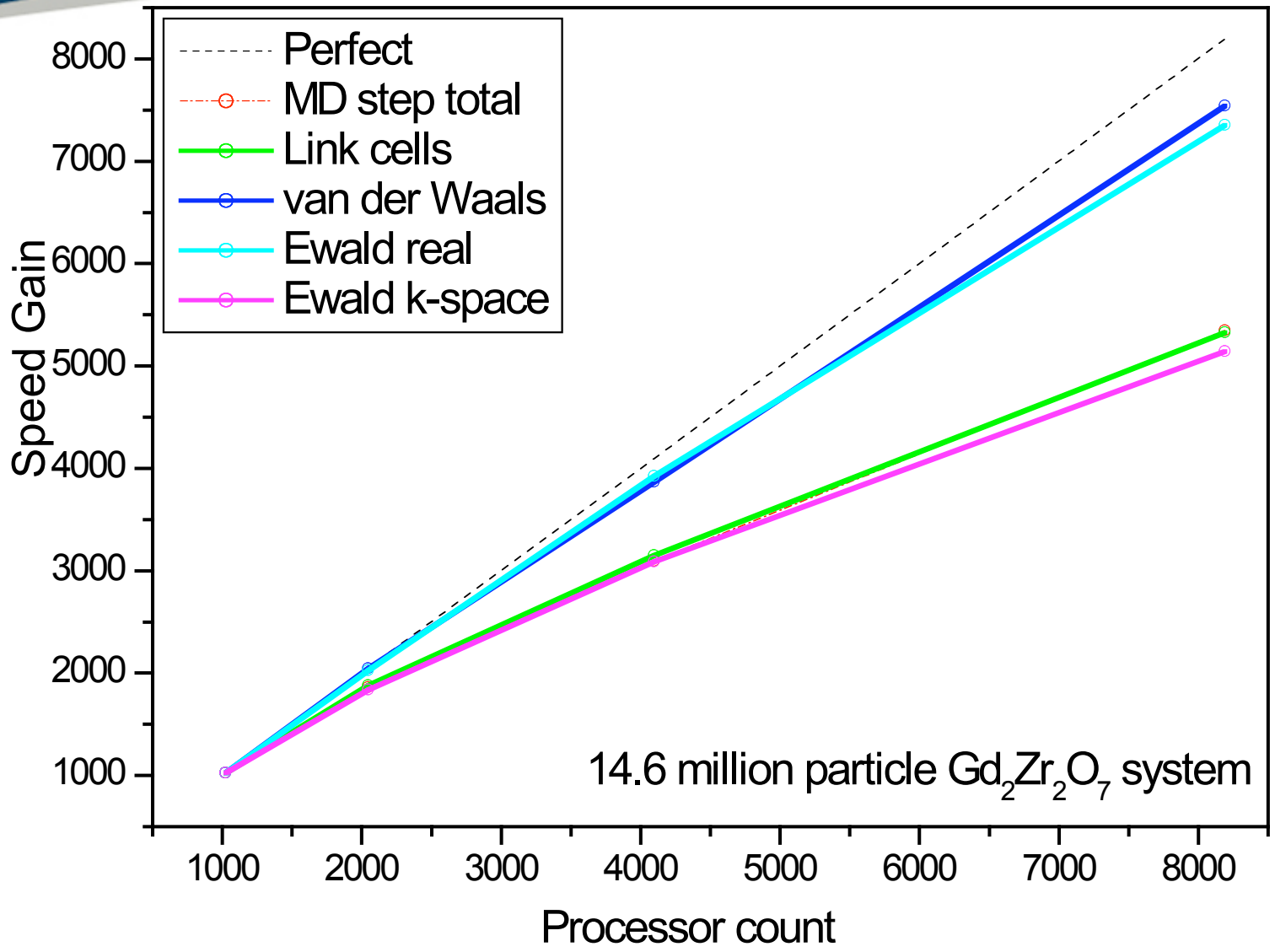
Science & Technology Facilities Council

# DL_POLY_3 I/O Conclusions

- **PDAW** performs markedly superiorly to the **SDAW** or **MPIW** where supported by the platform for this particular size of messages (73 Bytes ASCII per message). *Improvements by an order of magnitude can be obtained, even though the I/O is not scaling especially well itself.*
- **MPIW** optimised and performed consistently well for Cray XT3/4 architectures. **MPIW** and much better than **SDAW** but as seen on Cray XT3 this was still not as fast as **PDAW**. **MPIW** on Cray XT4 can achieve an improvement by a factor of two, similar performance to **PDAW** on the Cray XT3, once the storage methodology (OST) is optimised for the dedicated I/O processing units.
- **MPIW** performs badly on IBM platforms. **PDAW** not accessible on Cray XT4. **SDAW** extremely slow on Cray XT3.
- While on the IBM P5-575 **SNCW** was only 1.5 times faster than **SDAW** on average, on the Cray XT4 it was 10 times and it did not matter whether runs were in single- or dual-core mode. Despite these differences, SNCW performed 2.5 times faster on the IBM P5-575 than on the Cray XT4.

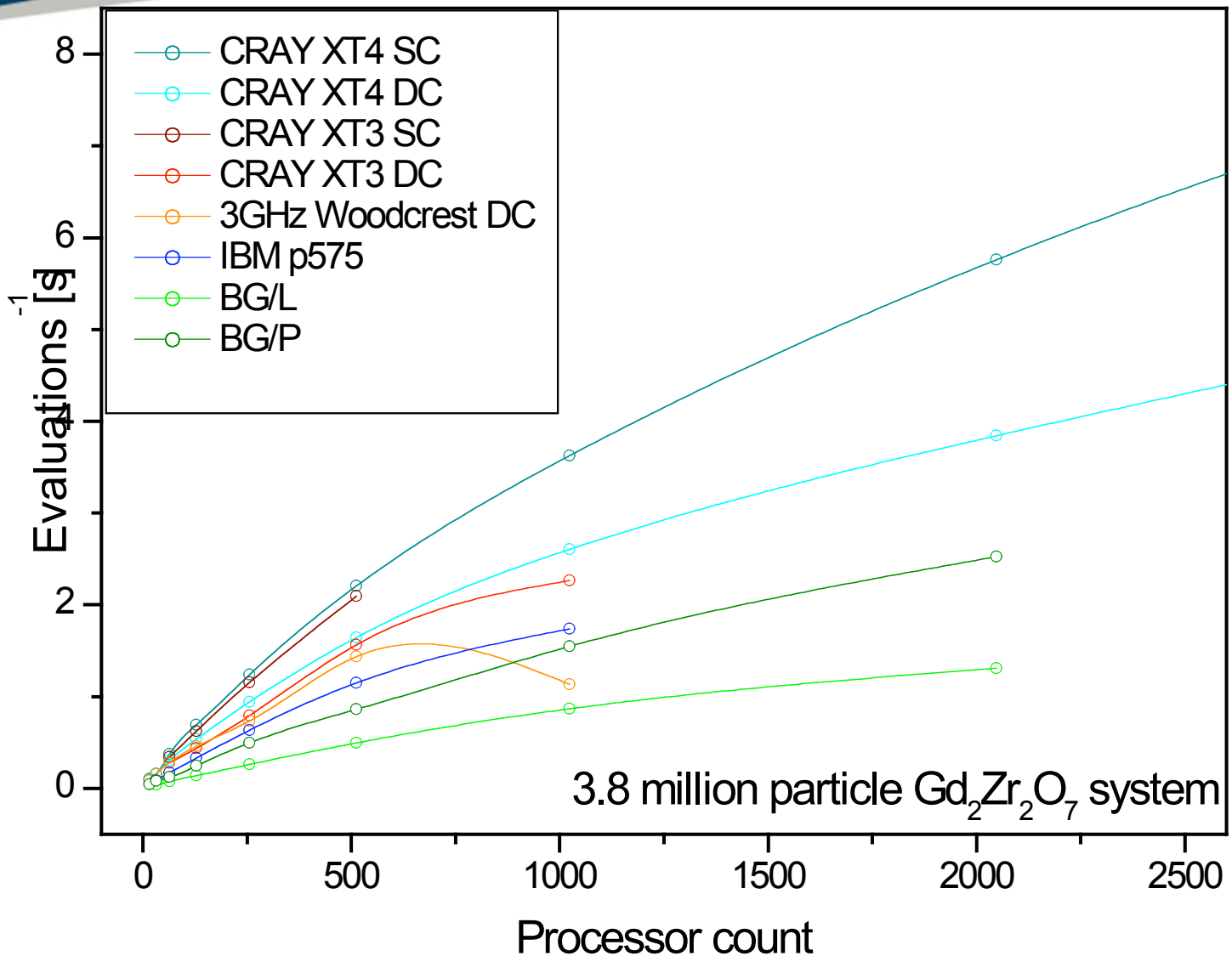
# Benchmarking BG/L Jülich




Perfect
MD step total
Link cells
van der Waals
Ewald real
Ewald k-space
Speed Gain
Processor count
1000
2000
3000
4000
5000
6000
7000
8000
14.6 million particle $Gd_2Zr_2O_7$ system

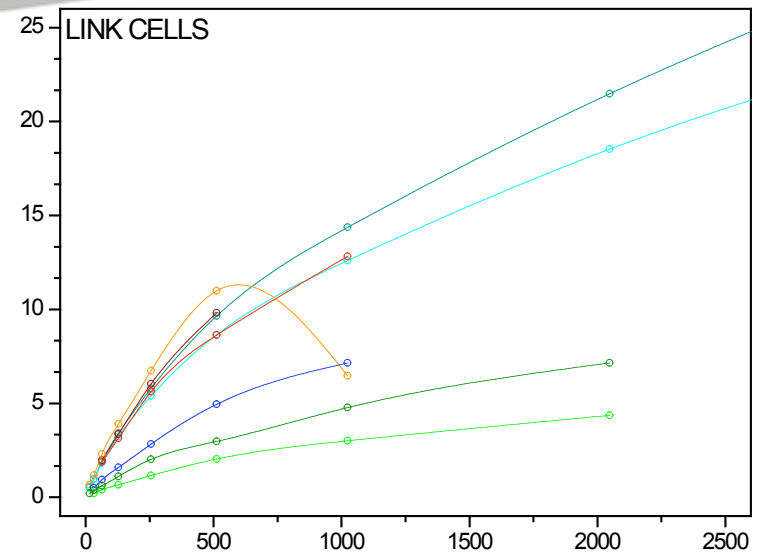
Science & Technology Facilities Council

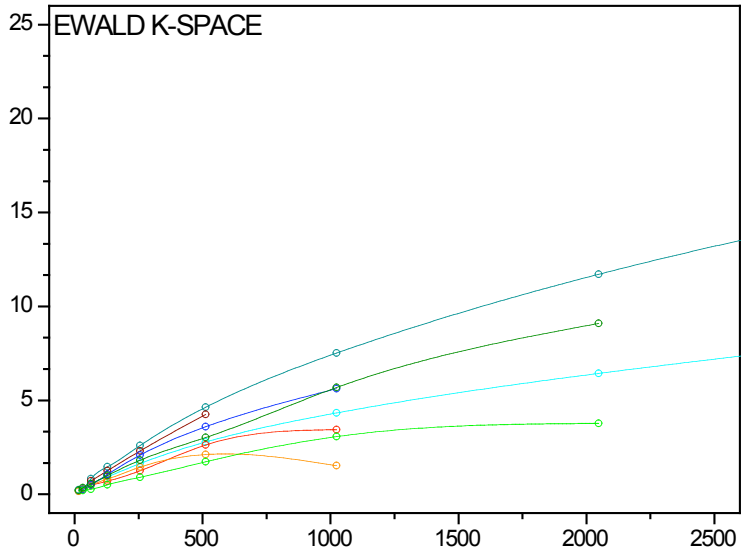
# Benchmarking Main Platforms - I

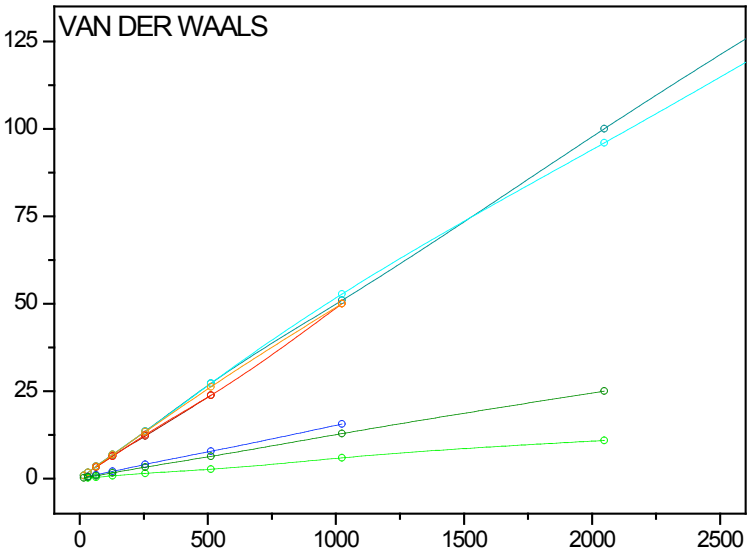


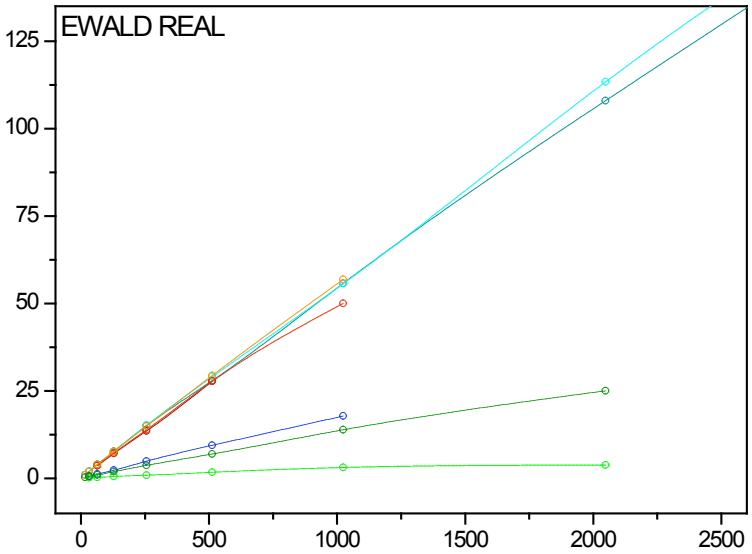
Science & Technology Facilities Council

# Benchmarking Main Platforms - II



Science & Technology Facilities Council

# Acknowledgements

**Thanks to**

- **Ian Bush (DL/NAG) for optimisation support**
- **Andy Porter (DL) for NetCDF work and support**
- **Martyn Foster (NAG) and David Quigley (UoW) for Cray XT4 optimisation**
- **Luican Anton (NAG) for first draft of MPI-I/O writing routine**

**http://www.ccp5.ac.uk/DL_POLY/**