The Application of Distributed Computing to the Investigation of Protein Conformational Change

C. J. Woods, J. G. Frey, J. W. Essex

School of Chemistry, University of Southampton, SO17 1BJ, UK

Abstract

Distributed computing is a potentially very powerful approach for accessing large amounts of computational power. Under the umbrella of the comb-e-chem project we have examined distributed computing software and applied it to the problem of investigating protein conformational change. These investigations required the development of protein simulations that were suited to distributed computing. Each simulation was split into many coupled, parallel parts. These proved challenging to schedule on the flexible and unreliable distributed computing resource. Scheduling algorithms were thus written that identified which parts of the simulation were likely to impact the overall efficiency. These parts were then rescheduled to be 'caught-up' via a fast and dedicated cluster.

1. Background

Distributed computing is a potentially powerful approach for accessing large amounts of computational power. Cycle stealers, which allow a PC user to donate the spare power of their computer, are now used in a wide range of scientific projects, e.g. the SETI@home study,[1] the CAN-DDO cancer screening project[2] and folding@home.[3] While the use of cycle stealers can provide supercomputer-like resources, their use is limited to calculations that may be split into many independently parallel parts (i.e. coarsely parallel simulations). The distributed and unreliable nature of this resource makes it unsuitable for closely coupled parallel calculations. For these calculations, the speed and latency of inter-processor communication are a bottleneck that cannot be overcome simply through the addition of more nodes. Unfortunately, a large number of chemical simulations require closely coupled parallel calculations, and are thus not suitable for deployment over a distributed computing cluster. An example of such a simulation is the investigation of protein conformational change. These simulations are typically performed using molecular dynamics (MD),[4] where the motions of the atoms are integrated over time using Newton's laws. These simulations cannot be broken up into multiple independent parts, as each nanosecond of MD must be run in series and in sequence.

The investigation of protein conformational change is important as it lies at the heart of many biological processes, e.g. cell signaling. Some bacteria regulate nitrogen metabolism using one such signalling pathway. Nitrogen Regulatory Protein C (NTRC)[5] plays a key role in this pathway. Changes in nitrogen concentration activate the kinase NTRB. This phosphorylates an aspartate residue in NTRC, causing it to change conformation (figure 1).[5] This change in conformation allows the NTRC to join together to form oligomers, which then help promote the transcription of genes. These genes are used to produce proteins that are used in nitrogen metabolism.[5] A key stage of this pathway is the change in conformation that occurs in NTRC when it is phosphorylated. It is difficult to study this conformational change experimentally as the phosphorylated form of NTRC has a very short lifetime.[6] It is thus desirable to simulate the NTRC protein and

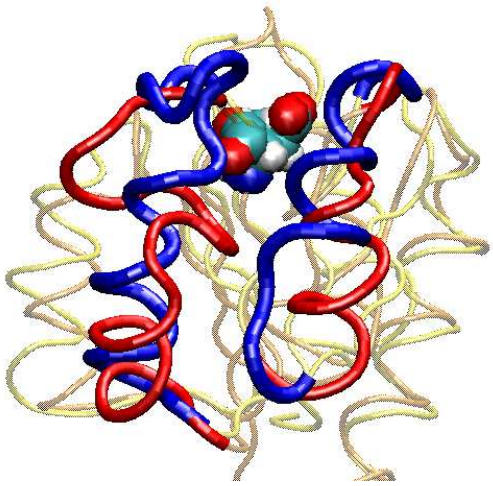

Figure 1. The native (blue) and phosphorylated (red) conformations of NTRC. The site of phosphorylation (Asp54) is shown as spheres.

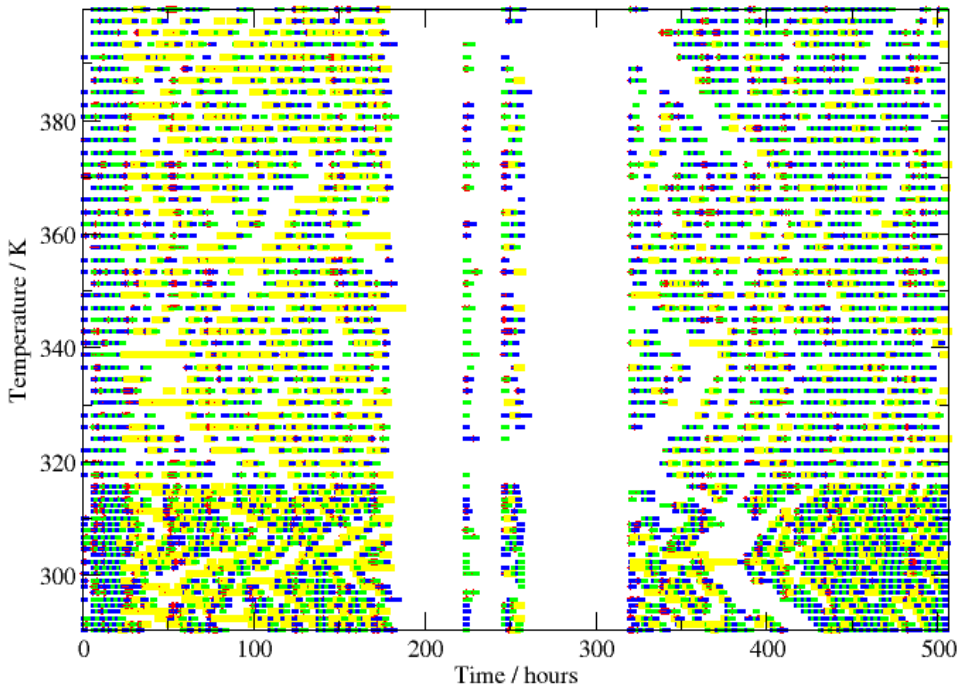

Figure 2. Progress of the simulation at each temperature as a function of simulation time. Iterations are run at each temperature; odd iterations are shown in blue and even iterations are shown in green. Our scheduler has to cope with extreme events, e.g. the complete failure of the distributed cluster after about 200 and 270 hours of simulation. The distributed cluster contains both fast and slow nodes. Some iterations can thus take a lot longer than others (visible here as longer bars). In addition, the owners of the PCs will also wish to use them (shown here as red dots). This will interrupt the calculation that temperature, again slowing it down relative to the other temperatures. Because of this, neighbouring temperatures will be ready to test at different times. This can lead to a loss of efficiency as completed temperatures wait for their neighbours. In the worst case this waiting can propagate, as occurs for temperatures around 310 K after 360 hours of simulation. To help prevent this, a catch-up cluster is used that identifies and reschedules slow temperatures (use of the catch-up cluster is shown in yellow).

encourage the conformational change on computer.

1.1 The Replica Exchange Method

We can use a distributed computing cluster to investigate protein conformational change via Replica Exchange simulations.[7,8] Multiple simulations of the protein are run in parallel, each running under a different condition, e.g. temperature. Periodically the simulations running at neighbouring temperatures are tested and swapped. This enables simulations at high temperatures, where there is rapid conformational change, to rain down to biologically relevant temperatures where conformational change occurs more slowly. The testing of neighbouring temperatures introduces a light coupling to the simulation, meaning that it no longer fits the archetypal coarsely parallel distributed computing model. This light coupling introduces inefficiencies to the scheduling of the simulation, as any delay in the calculation of one temperature can propagate out to delay the calculation of all temperatures. To help overcome this, a catch-up cluster has been developed that monitors the simulation for temperatures that are taking too long to complete, and that are likely to negatively impact the overall efficiency of the simulation. Once identified, the calculation of these temperatures is rescheduled onto a small, yet fast and dedicated, computational resource so that they can 'catch-up' with the other temperatures (figure 2).

2. Experimental Details

NMR structures of the phosphorylated (1DC8) and unphosphorylated (1DC7) conformations of the NTRC protein were obtained from the protein databank.[9] Polar hydrogen atoms were added via WhatIf[10]. The proteins were solvated in 60^3 Å^3 boxes of TIP3P water and sodium ions were added via the XLEAP module of AMBER 7.0[11] to neutralise the system. The CHARMM27 forcefield[12] was used, and the systems minimised, then annealed from 100 K to 300 K.

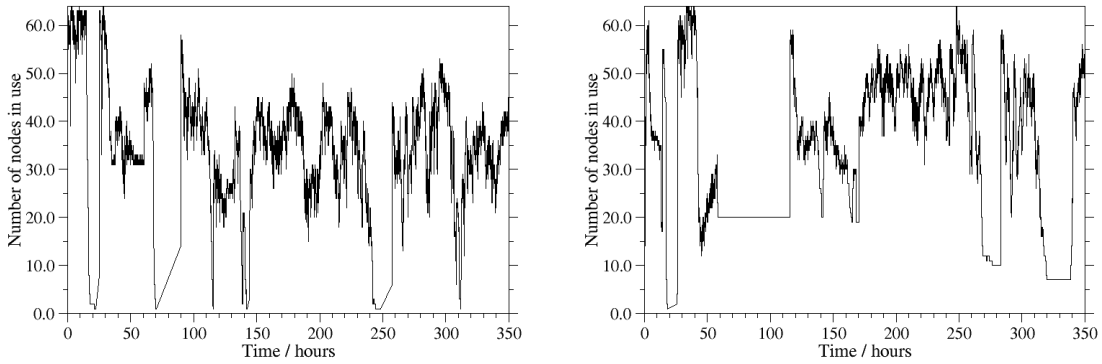

Figure 3. The number of nodes in use for the replica exchange simulation over just the distributed cluster (left), and over the distributed and catchup clusters (right).

The systems were finally equilibrated for 100 ps at constant temperature (300 K) and pressure (1 atm). The final structures from equilibration were used as the starting structures for all of the replicas.

The temperatures for each replica were chosen using a custom program that optimised the temperature distribution such that a replica exchange move was accepted with a probability of 20%. The temperatures chosen ranged from several replicas below the target temperature of 300 K (290.1 K) to a maximum of 400 K. In total 64 replicas were used for each of the two proteins.

The simulations were conducted using NAMD 2.5.[13] A replica exchange move was attempted between neighbouring temperatures every 2 ps, after the initial 20 ps of sampling that was used to equilibrate each replica to its initial temperature. A Langevin thermostat[14] and a Nose-Hoover Langevin piston barostat[15] were used to sample at constant temperature and pressure, while SHAKE[16] was used to constrain hydrogen bond lengths to equilibrium values. A 1 fs time step was used for the molecular dynamics integrator, and the non-bonded interactions were evaluated using a 12 Å cutoff and the particle mesh ewald sum.[17]

2.1 Details of the Distributed Cluster

The simulations were run over the condor[18] cluster provided by the University of Southampton. This cluster uses condor[18] to make available the spare cycles of approximately 450 desktop computers running Microsoft Windows NT 5.1. The two replica exchange simulations were run simultaneously on this cluster. There was little competition between the two simulations for nodes as they each required a maximum of 64 nodes out of the available 450.

2.2 Implementation of the Catchup Cluster

The catchup cluster was implemented via dedicated dual Xeon 2.8 GHz nodes running Linux. Each Xeon processor was able to provide 2 virtual processors, allowing NAMD to run in parallel over 4 virtual processors per node. This meant that each dual Xeon could complete 2 ps of MD approximately 3.5 times quicker than a typical node on the condor cluster. The catchup cluster consisted of two dual Xeon nodes, thus allowing it to catch up two replicas simultaneously.

To test the utility of the catchup cluster, it was only made available to the replica exchange simulation on the phosphorylated conformation of the protein (1DC8). As both replica exchange simulations were running simultaneously, any differences in efficiency should thus be wholly attributable to use of the catchup cluster.

3. Results

Figure 3 shows the number of nodes in use during the replica exchange simulations on the phosphorylated and unphosphorylated conformations of NTRC. The initial phase of the simulation involved the 20 ps of equilibration of each replica to its initial temperature. This was broken down into 10 iterations of 2 ps. As there were no replica exchange moves during these first 10 iterations the replicas were all independent and thus the maximum number of 64 nodes were in use. However, there were efficiency problems during this phase of the simulation, as the unreliable nature of the distributed cluster caused several short periods of downtime that stopped both simulations. Frequent periods of downtime were common throughout the rest of replica exchange simulations.

The second stage of the simulations occurred when the replicas begun to complete their 10th iteration. At this point each replica had to wait for its partner to complete 10 iterations such that the pair of replicas could be tested and potentially swapped. Owing to the range of processors available in the distributed cluster and the different impacts of downtime on each of the replicas, there was a large spread of times over which each replica completed 10 iterations. This meant that a large number of replicas were left waiting for a significant time for their partner to complete, and thus the number of nodes in use for each simulation dropped from the maximum of 64 down to approximately 30. If the efficiency is defined as the ratio of the number of nodes in use compared to the theoretical maximum, then the efficiency dropped from 100% down to about 47%.

After this dip in efficiency, the simulations then moved towards the final stage, which was a steady state, where the number of replicas running and the number of replicas waiting for their partner to complete reached a consistent range of values. This steady state was periodically disrupted by failure of the condor cluster, but was always quickly recovered once the disruption was over. The steady state for the simulation that used the catchup cluster had significantly more replicas running, and significantly fewer replicas waiting compared to the simulation that did not use the catchup cluster. The catchup cluster clearly improved the steady state number of nodes in use to approximately 50, compared to approximately 40 for the simulation that did not use the catchup cluster. This is an improvement in efficiency from 63% to 78%.

3.1 The Heterogenous Distributed Cluster

The distributed condor cluster consisted of a range of desktop computers with varying processor speeds. To investigate the effect of running the simulation on this heterogeneous cluster, the total simulation time for each iteration was histogrammed. The histogram of replica completion times for the phosphorylated form of the protein is shown in figure 4. This figure shows that while the majority of iterations completed in under ten thousand seconds (2.8 hours), there was a significant spread of replica completion times up to twenty thousand seconds (5.6 hours). This spread of completion times means that fewer iterations have been completed than would have been expected based on the speed of the processors used.

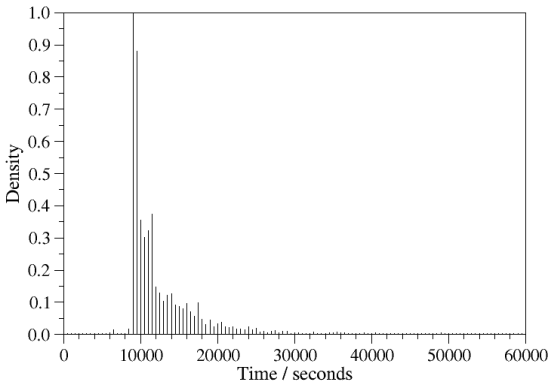

Figure 4. Histogram of the times to complete each iteration of the replica exchange simulation on the phosphorylated form of NTRC.

3.2 Comparison to Normal MD

The use of dual Xeons in the catchup cluster aiding the efficiency of a replica exchange simulation was compared to their use as dedicated nodes running a normal Molecular Dynamics simulation. The phosphorylated conformation of NTRC was simulated at 300 K using a standard MD simulation with an identical starting structure as the replica exchange simulation and identical simulation conditions. Over the same period of time as the replica exchange simulations were running, this MD simulation on a single dual Xeon node completed 1.9 ns of dynamics. This compares to a total of 10.5 ns of dynamics generated by the corresponding replica exchange simulation. However, the 1.9 ns of dynamics generated via the MD simulation forms a single, self-consistent trajectory. In comparison, the 10.5 ns of sampling from the replica exchange simulations was formed over 64 individual trajectories of only 0.16 ns in length. The dedicated node has produced a single trajectory over ten times the length of those produced via the distributed condor cluster with catchup cluster. This is despite the dedicated node only running the MD approximately 3.5 times the speed of a typical node in the distributed cluster. There are two reasons for this discrepancy; firstly, as demonstrated in figure 4, the heterogeneous nature of the distributed cluster meant that there was a large spread in the amount of time needed to complete each iteration of the replica exchange simulation. This could be mitigated against by running each iteration twice at the same time on the distributed cluster and using the results from the first node that completed the calculation. The second reason for the discrepancy is that the

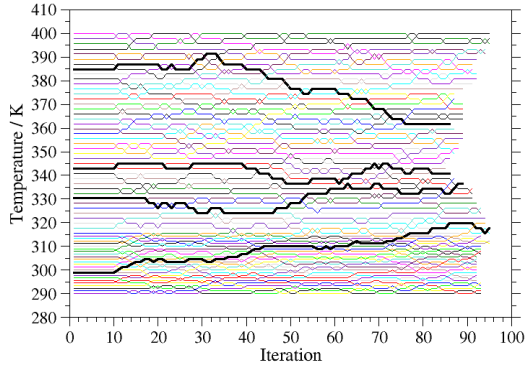

Figure 5. Temperature of each replica as a function of iteration from the replica exchange simulation on the unphosphorylated conformation of NTRC. Four randomly chosen replicas are highlighted.

distributed cluster was very unreliable, leading to large periods of time when the simulation was not running. The replica exchange simulations presented here were run at a time when the condor cluster was experiencing a higher than normal amount of downtime. It is anticipated that during normal operation the condor cluster would be more reliable, and that the steady state efficiency of the replica exchange simulations would be maintained throughout the majority of the simulation. However, the experience of running these simulations demonstrate that applications that use distributed clusters need to include estimates of downtime and the range of available resources when predicting how long a particular simulation will take to run. These results also demonstrate that a distributed computing resource is, unsurprisingly, not efficient compared to a dedicated computing resource. However, distributed computing typically provides resources that would otherwise not be available.

3.3 Effectiveness of Replica Exchange

The primary aim of running these simulations over the distributed computing resource was to sample the conformational change induced by phosphorylating NTRC. The aim was to use replica exchange to swap simulations running at high temperature, where the conformational change occurs more rapidly, down to room temperature, where the simulation statistics are collected. Figure 5 shows the temperature for each replica of the unphosphorylated simulation as a function of iteration. Four of the replicas are highlighted. This figure shows that while the replica exchange moves were accepted with the desired frequency, the replicas themselves did not travel far in temperature. Instead each replica tended to oscillate around its initial temperature. No replicas from high temperature swapped down to room temperature. This shows that the replica exchange simulations need to be continued for many more iterations before the improved sampling of high temperature is able to be of use in enhancing the rate of sampling at room temperature.

4. Conclusion

Distributed computing provides a resource that is not ideally suited to a wide range of chemistry problems. The investigation of protein conformational change is one such problem. The replica exchange algorithm was used in an attempt to fit this chemistry problem to the distributed computing resource. The coupled nature of replica exchange simulations caused problems for the scheduling of the computation that were partially solved through the development of a dedicated catchup cluster. This cluster improved the efficiency of the replica exchange simulation from 63% to 78%.

Acknowledgements

We thank R. Gledhill, A. Wiley and L. Fenu for discussions and the EPSRC for funding comb-e-chem.

References

1. http://setiathome.ssl.berkeley.edu
2. http://www.chem.ox.ac.uk/curecancer.html
3. http://www.foldingathome.org
4. Leach, A.R., *Molecular Modelling, Principals and Applications*, Longman, Harlow, UK, 1996
5. Pelton, J.G., Kustu, S., Wemmer, D.E., *J. Mol. Biol.*, 292, 1095, 1999
6. Kern, D., Volkman, B.F., Luginbuhl, P., Nohaile, M.J., Kustu, S., Wemmer, D.E., *Nature*, 402, 894, 1999
7. Sugita, Y., Kitao, A., Okamoto, Y., *J. Chem. Phys.*, 113, 6042-6051, 2000
8. Hansmann, U.H.E., *Chem. Phys. Lett.*, 281, 140, 1997
9. http://www.rcsb.org/pdb/
10. Vriend, G., Hooft, R.W.W., Van Aalten, D., WhatIf, 1997
11. Pearlman, D.A., Case, D.A., Caldwell, J.W., Ross, W.S., Cheatham, T.E., Debolt, S., Ferguson, D., Seibel, G., Kollman, P., *Comput. Phys. Commun.*, 91, 1, 1995
12. Mackerrell, A.D., Bashford, D., Bellot, M., Dunbrack, L., Evanseck, M., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-Mccarthy, D., Kuchnir, L., Kuczera, K., Lau, F.T.K., Mattos, C., Michnick, S., Ngo, T., Nguyen,

D.T., Prodhom, B., Reiher, W.E., Roux, B., Sclenkrich, M., Smith, J.C., Stote, R., Straub, J., Watanabe, M., Wiorkiewicz-Kuczera, J., Yin, D., Karplus, M., *J. Phys. Chem. B.*, 102, 3586, 1998

13. Kale, L., Skeel, R., Bhandarker, M., Brunner, R., Gursoy, A., Krawetz, N., Phillips, J., Shinozaki, A., Varadarajan, K., Schulten, K., *J. Comp. Phys.*, 151, 283, 1999. NAMD was developed by the Theoretical Biophysics Group in the Beckman Institute at Urbana-Champaign.
14. Paterlini, M.G., Ferguson, D.M., *Chem. Phys.*, 236, 243, 1998
15. Feller, S.E., Zhang, Y.H., Pastor, R.W., Brooks, B.R., *J. Chem. Phys.*, 103, 4613, 1995
16. Ryckaert, J.P., Ciccotti, G., Berendsen, J.C., 23, 327, 1977
17. Darden, T., York, D., Pedersen, L., *J. Chem. Phys.*, 98, 10089, 1993
18. http://www.cs.wisc.edu/condor