DESRES/TR--2008-01

Desmond Performance on a Cluster of Multicore Processors

Edmond Chow,[1] Charles A. Rendleman,[1] Kevin J. Bowers,[1] Ron O. Dror,[1] Douglas H. Hughes,[1] Justin Gullingsrud,[1] Federico D. Sacerdoti,[1] and David E. Shaw[1,2]

28 July 2008

Summary

Desmond is a molecular dynamics (MD) code designed for high performance on large commodity Linux/Unix clusters. This report documents the performance that Desmond achieved on new hardware in April 2008. For the DHFR system (23,558 atoms) with production simulation parameters, Desmond's top simulation rate is 471 ns/day on 1024 cores of an InfiniBand cluster. For the ApoA1 system (92,224 atoms), Desmond's top simulation rate is 289 ns/day on 4096 cores of the same cluster.

Hardware and Operating Environment

The cluster used for the following performance measurements comprises 576 nodes, with each node containing two 2.66 GHz Xeon E5430 processors, for a total of eight cores per node (4608 cores total in the cluster). Each E5430 processor contains two 6 MB L2 caches; each cache is shared by two cores. The network is InfiniBand DDR with ConnectX host adapters and is non-blocking (i.e., has full bisection bandwidth). The operating system is CentOS 4.6 x86_64 Linux (kernel 2.6.22.15 from the Rocks 4.1 cluster distribution). The InfiniBand drivers were Voltaire Gridstack 4.3.5 (based on the OpenFabrics Alliance software stack, version 1.2.5.1) and the MPI implementation was Open MPI, version 1.3a1r17802. A custom communication library based on InfiniBand Verbs was used for point-to-point communication.

Benchmark Systems and Simulation Parameters

Performance results were measured for two chemical systems that are among the most common benchmark systems used for MD codes. These are the ApoA1 (apolipoprotein A1) system with 92,224 atoms (in a global cell of approximately 109×109×78 $Å^3$) [1] and the DHFR

[1] D. E. Shaw Research, New York, NY 10036, USA.
[2] Center for Computational Biology and Bioinformatics, Columbia University, New York, NY 10032, USA. To whom correspondence should be addressed; e-mail: David.Shaw@DEShawResearch.com.

(dihydrofolate reductase) system, also known as the Joint Amber-CHARMM benchmark [2], with 23,558 atoms (approximately 62×62×62 $Å^3$).

For each system, two sets of simulation parameters were used. The first set, called the *benchmark parameters*, are parameters that were previously specified in the literature to be used for benchmarking these two chemical systems. The second set, called the *production parameters*, are parameters that we have chosen to optimize Desmond performance without compromising accuracy [3]. The following tables give pertinent benchmark and production parameters for each chemical system. All simulations were run without temperature or pressure control. For the full set of parameters that were used in these simulations, see Appendix B.

System	Time step	Constraints	PME frequency	Cutoff	PME mesh	PME order
ApoA1	1 fs	No	4 steps	12 Å	128×128×128	4
DHFR	1 fs	No	1 step	9 Å	64×64×64	4

Table 1. Benchmark Parameters. The cutoff is the cutoff radius used for evaluation of pairwise electrostatic and van der Waals interactions, the PME order is the order of the B-splines used in the smooth PME algorithm, and the PME frequency is the interval at which the far term electrostatic forces are evaluated.

System	Time step	Constraints	PME frequency	Cutoff	PME mesh	PME order
ApoA1	2.5 fs	Yes	2 steps	12 Å	64×64×64	6
DHFR	2.5 fs	Yes	2 steps	9 Å	64×64×64	4

Table 2. Production Parameters. The meaning of the parameters is the same as those in Table 1.

Note that for ApoA1 benchmark parameters, Desmond uses a 128×128×128 PME mesh that is finer than that specified by the benchmark (108×108×80), due to a restriction on FFT sizes used in Desmond at the time the simulations were run (this restriction has since been removed).

Results

In the following tables, Desmond simulation rate is reported in units of simulated nanoseconds per wall-clock day and is shown as a function of the number of cores.

Results using under-subscribed nodes

For a given number of cores, the top performance in Desmond is achieved when we do not fully use all the cores in a node, but instead we use more nodes—the nodes are *under-subscribed*. This effectively provides more memory bandwidth to each core and improves the network latency experienced by each core. The additional cores are not used or can be used for non-network and low-memory bandwidth tasks. We note in particular that Desmond's top performance of 471 ns/day on DHFR on the given hardware is achieved this way.

In general, however, running with fewer than the maximum number of cores per node reduces overall throughput of a computing cluster, and is thus not recommended. (There are, on the other hand, some situations where allocating one core per node for operating system functions does improve throughput.) We show the following results for the case of two cores per node. These results are useful for comparison with reported results on clusters with two cores per node (e.g., [3]) or under-subscribed nodes (e.g., [4], [5]) where the effect of network interface contention needs to be factored out.

nodes	**cores**	**DHFR**		**ApoA1**	
		benchmark parameters	**production parameters**	**benchmark parameters**	**production parameters**
4	8	5.09	13.79	0.95	2.15
8	16	9.52	25.47	1.77	4.00
16	32	16.52	46.04	3.30	7.45
32	64	30.25	82.05	6.24	14.36
64	128	51.59	136.52	11.77	26.72
128	256	84.87	227.55	21.38	49.10
256	512	132.41	358.57	39.34	86.07
512	1024	172.07[3]	471.43[3]	66.92	142.24

Table 3. Simulation rate (ns/day) using under-subscribed nodes.

[3] Using 512 processes, 2 threads per process. Threads were run on the same socket on two cores that share L2 cache. For the case of DHFR with benchmark parameters, if the threads are run on separate sockets, the performance is 118.72 ns/day.

Results using fully-subscribed nodes

In the following table, simulation rate (in ns/day) is shown as a function of the number of cores. All 8 cores on each node were used for computation. Each node ran 8 processes and each process ran a single thread. Note that the maximum number of processes for DHFR is 512 and the maximum number of processes for ApoA1 is 2048.

nodes	cores	DHFR		ApoA1	
		benchmark parameters	production parameters	benchmark parameters	production parameters
1	8	4.42	12.35	0.69	1.86
2	16	8.08	22.72	1.45	3.60
4	32	14.02	41.06	2.74	6.79
8	64	26.70	75.55	5.21	13.44
16	128	42.63	122.31	9.59	25.76
32	256	61.28	186.02	16.51	47.39
64	512	90.51	278.69	31.44	81.74
128	1024			52.55	130.55
256	2048			86.87	189.62

Table 4. Simulation rate (ns/day) using fully-subscribed nodes.

Results using two threads per process

Desmond uses a parallelization method which partitions space into boxes. Desmond does not allow box edge lengths to be less than half the cutoff radius, to avoid the need for boxes to communicate with other boxes more than one box-length away. This restriction limits the number of boxes and thus the number of processes that Desmond can use simultaneously.

Each process typically uses a single thread and a single core, but to extend parallelism and use more cores, Desmond can run more than one thread per process. Threads divide the computational load, although only a single thread performs interprocess communication. This is often called a *hybrid communication model*. In the following table, simulation rate is shown when two threads are used per process and each thread is allocated its own core for computation. In particular, the two threads associated with the same process are allocated and fixed to one of the two cores that share L2 cache on the Xeon E5430 processor. Other allocations of the threads to cores resulted in much degraded performance due to the need to copy data between L2 caches.

nodes	cores	DHFR		ApoA1	
		benchmark parameters	production parameters	benchmark parameters	production parameters
2	16	7.22	21.25	1.42	3.60
4	32	12.86	37.93	2.69	6.81
8	64	20.97	64.67	5.03	12.77
16	128	37.93	112.54	9.47	24.89
32	256	57.35	172.92	16.16	45.65
64	512	89.58	268.85	27.42	80.98
128	1024	137.49	394.74	50.86	132.30
256	2048			83.79	204.21
512	4096			127.54	289.04

Table 5. Simulation rate (ns/day) using two threads per process and fully-subscribed nodes.

Desmond using hybrid communication is expected to perform more poorly than when each core has its own communicating process. This is due to the serialization of communication in the hybrid case. On the other hand, there is a potential for improved performance in the hybrid case because there are fewer communicating processes and there can be less contention for the network interface on each node. The results in the tables above show that Desmond's hybrid performance (for two threads per process) is very comparable to when a single thread per process is used. However, for four or eight threads per process (not shown), Desmond's performance degrades, most likely due to increased communication serialization, and possibly also due to additional data copying between L2 caches and between cores.

Using two threads per process allows us to effectively use 4096 cores in the ApoA1 simulations. The top performance that Desmond achieves with ApoA1 is 289 ns/day using this configuration along with production parameters. We note that except to extend parallelism this way, using multiple threads per process is generally not recommended.

Desmond Availability

We are making Desmond available without cost for non-commercial use at universities and other not-for-profit research institutions, with its first free release in June 2008. Desmond is also available commercially from Schrödinger, LLC. These initial releases of the software will run well on current commodity clusters but do not contain all the optimizations for multicore processors used in the performance studies above. These optimizations, which will be added to future releases of Desmond, make little difference at low levels of parallelism or low numbers of cores per node. Even at high levels of parallelism with many cores per node, the difference rarely exceeds 10%.

Appendix A

This appendix reprises results measured in June 2006 and reported earlier [3], using a 1056 node (2112 processor) InfiniBand cluster. Each node of this cluster contained two 2.4 GHz AMD Operton Model 250 (single core) processors. InfiniBand PCI-X host adapters were used. The network fabric was non-blocking.

nodes	cores	DHFR		ApoA1	
		benchmark parameters	production parameters	benchmark parameters	production parameters
4	8	2.09	5.87	0.34	0.85
8	16	4.12	11.26	0.68	1.74
16	32	7.54	20.91	1.34	3.36
32	64	13.78	36.91	2.58	6.40
64	128	23.36	62.04	4.76	11.67
128	256	42.75	115.16	9.23	22.13
256	512	63.56	173.43	16.75	39.62
512	1024			29.14	70.98
1024	2048			42.97	119.84

Table 6. Simulation rate (ns/day) reported in 2006 using an earlier cluster. Simulations used fully-subscribed nodes on a machine with two cores per node.

Appendix B

This appendix reproduces the configuration files used in the performance tests.

DHFR Benchmark Parameters

```
mdsim = {
  title      = "5dhfr benchmark parameters"

  last_time=0.100

  randomize_initial_velocities   = false # Whether to randomize init velocities
  randomize_initial_velocities_seed =1   # RNG seed for randomizing velocities
  randomize_initial_velocities_temp =300 # Temp. for randomizing velocities
}

global_cell={
  reference_time=0.0
  topology     = periodic
  partition    =[  0 0 0  ]
  r_clone =4.9
  est_pdens            =0.1
}

force = {
  type = desmond
  nonbonded = {
    type       = vdw-elec-force-only # optimized

    r_lazy     = 9.8
    r_cut      = 9.0
    r_tap      = 9.0
    n_zone     = 1024
    taper      = none

    far = {
      type     = pme
      order    = [4 4 4]
      n_k      = [64 64 64]
      sigma    = 2.1123
    }
    average_dispersion        =69.5     # Average dispersion coefficient for
                                        # tail correction to E and P,
                                        # in kcal/mol A^6

  }
  sync_random_number_seed  =2006          # Seed for random number generator
}

######### Integrator configuration

integrator = {
  type = V_NVE

  dt=0.001
  temperature=[]

  center_frozen_group       =false        # Center the c.o.m. of the frozen grp
  remove_com_motion         =false        # Whether to remove center of mass
                                          # motion
```

```
  migrate={first   =0.0       # legacy (should be set to zero).
           interval=0.016}      # in ps

  respa={ bonded_interval        =1
          nonbonded_near_interval=1
          nonbonded_far_interval =1
  }

  pressure = {
    isotropy                  =isotropic    # {isotropic|semi_isotropic|
                                            # anisotropic|flexible}
    max_margin_contraction  =0.9     # Maximum pair margin contraction ratios

    p_ref=1.0 # bar
  }
}

######### Output control
mdsim {
  plugins=[status]
  status={
    first         =0
    interval      =.5
  }
  checkpt={}
  eneseq={
    name          =eneseq
    first         =0
    interval      =0.01
  }
}
```

DHFR Production Parameters

```
mdsim = {
  title      = "5dhfr production parameters"

  last_time=0.500

  randomize_initial_velocities   =false  # Whether to randomize init velocities
  randomize_initial_velocities_seed =1   # RNG seed for randomizing velocities
  randomize_initial_velocities_temp =300 # Temp. for randomizing velocities
}

global_cell={
  reference_time=0.0
  topology     = periodic
  partition    =[  0 0 0  ]
  r_clone =5.0
  est_pdens            =0.1
}

constraint = {
  tol          = 1e-08  # constraint tolerance
  maxit        = 5      # maximum number of iterations
}

force = {
  type = desmond
  nonbonded = {
    type       = vdw-elec-force-only

    r_lazy     = 10.0
```

```
    r_cut      = 9.0
    r_tap      = 9.0
    n_zone     = 1024
    taper      = none

    far = {
      type     = pme
      order    = [4 4 4]
      n_k      = [64 64 64]
      sigma    = 2.1123
    }
    average_dispersion        =69.5     # Average dispersion coefficient for
                                        # tail correction to E and P,
                                        # in kcal/mol A^6

  }
  sync_random_number_seed  =2006          # Seed for random number generator
}

######### Integrator configuration

integrator = {
  type = V_NVE # Ber_NPT | Ber_NVT | MTK_NPT | NH_NVT | V_NVE | L_NVT |
minimize

  dt=0.0025
  temperature=[]

  center_frozen_group       =false        # Center the c.o.m. of the frozen grp
  remove_com_motion         =false        # Whether to remove center of mass
                                          # motion
  migrate={first   =0.0     # legacy (should be set to zero).
           interval=0.02}      # in ps

  respa={ bonded_interval        =1
          nonbonded_near_interval=1
          nonbonded_far_interval =2
  }

  pressure = {
    isotropy                  =isotropic    # {isotropic|semi_isotropic|
                                            # anisotropic|flexible}
    max_margin_contraction  =0.9     # Maximum pair margin contraction ratios

    p_ref=1.0 # bar
  }

  #
  # integrator-specific stuff goes in its OWN SECTION
  #

  V_NVE = {}
}

######### Output control
mdsim {
  plugins=[status]
  status={
    first         =0
    interval      =0.5
  }
  checkpt={}
  eneseq={
```

```
    name          =eneseq
    first         =0
    interval      =0.05
  }
}
```

ApoA1 Benchmark Parameters

```
mdsim = {
  title      = "ApoA1 benchmark parameters"

  last_time=0.400

  randomize_initial_velocities   = false # Whether to randomize init velocities
  randomize_initial_velocities_seed =1   # RNG seed for randomizing velocities
  randomize_initial_velocities_temp =300 # Temp. for randomizing velocities
}

global_cell={
  reference_time = 0.0
  topology     = periodic
  partition    =[  0 0 0  ]
  r_clone =6.4
  est_pdens            =0.1
}

force = {
  type = desmond
  nonbonded = {
    type       = vdw-elec-force-only # optimized

    r_lazy     = 12.8
    r_cut      = 12.0
    r_tap      = 10.0
    taper      = force
    n_zone     = 2048

    far = {
      type     = pme
      order    = [4 4 4]
      n_k      = [128 128 128]
      sigma    = 2.3553
    }
    average_dispersion        =69.5     # Average dispersion coefficient for
                                        # tail correction to E and P,
                                        # in kcal/mol A^6

  }
  sync_random_number_seed  =2006          # Seed for random number generator
}

######### Integrator configuration

integrator = {
  type = V_NVE

  dt=0.001
  temperature=[]

  center_frozen_group       =false        # Center the c.o.m. of the frozen grp
  remove_com_motion         =false        # Whether to remove center of mass

  migrate={first   =0.0       # legacy (should be set to 0)
```

```
           interval=0.016}      # in ps

  respa={ bonded_interval        =1
          nonbonded_near_interval=1
          nonbonded_far_interval =4
  }

  pressure = {
    isotropy                  =isotropic

    max_margin_contraction  =0.9     # Maximum pair margin contraction ratios

    p_ref=1.0 # bar
  }

}

######### Output control
mdsim {
  plugins=[status]
  status={
    first         =0
    interval      =0.5
  }
  checkpt={}
  eneseq={
    name          =eneseq
    first         =0
    interval      =0.04
  }
}
```

ApoA1 Production Parameters

```
mdsim = {
  title      = "ApoA1 production parameters"

  last_time=0.500

  randomize_initial_velocities   = false # Whether to randomize init velocities
  randomize_initial_velocities_seed =1   # RNG seed for randomizing velocities
  randomize_initial_velocities_temp =300 # Temp. for randomizing velocities
}

global_cell={
  reference_time = 0.0
  topology     = periodic
  partition    =[  0 0 0  ]
  r_clone =6.5
  est_pdens            =0.1
}

constraint = {
  tol          = 1e-08  # constraint tolerance
  maxit        = 5      # maximum number of iterations
}

force = {
  type = desmond
  nonbonded = {
    type       = vdw-elec-force-only # optimized

    r_lazy     = 13.0
```

```
    r_cut      = 12.0
    r_tap      = 12.0
    n_zone     = 2048
    taper      = force

    far = {
      type     = pme
      order    = [6 6 6]
      n_k      = [64 64 64]
      sigma    = 2.5712
    }
    average_dispersion        =69.5     # Average dispersion coefficient for
                                        # tail correction to E and P,
                                        # in kcal/mol A^6

  }
  sync_random_number_seed  =2006          # Seed for random number generator
}

######### Integrator configuration

integrator = {
  type = V_NVE # Ber_NPT | Ber_NVT | MTK_NPT | NH_NVT | V_NVE | L_NVT |
minimize

  dt=0.0025
  temperature=[]

  center_frozen_group       =false        # Center the c.o.m. of the frozen grp
  remove_com_motion         =false        # Whether to remove center of mass
                                          # motion
  migrate={first   =0.0      # ps       Legacy, should be set to zero.
           interval=0.02}      # ps

  respa={ bonded_interval        =1
          nonbonded_near_interval=1
          nonbonded_far_interval =2
  }

  pressure = {
    isotropy                  =isotropic

    max_margin_contraction  =0.9     # Maximum pair margin contraction ratios

    p_ref=1.0 # bar
  }

}

######### Output control
mdsim {
  plugins=[status]
  status={
    first         =0
    interval      =0.5
  }
  checkpt={}
  eneseq={
    name          =eneseq
    first         =1
    interval      =5
  }
}
```

References

1. James C. Phillips, Gengbin Zheng, Sameer Kumar, and Laxmikant V. Kalé, "NAMD: Biomolecular Simulation on Thousands of Processors," *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing (SC02)*, Balitimore, Maryland, November 16–22, 2002.
2. MD Benchmarks for Amber, CHARMM and NAMD. http://amber.scripps.edu/amber8.bench2.html.
3. Kevin J. Bowers, Edmond Chow, Huafeng Xu, Ron O. Dror, Michael P. Eastwood, Brent A. Gregersen, John L. Klepeis, István Kolossváry, Mark A. Moraes, Federico D. Sacerdoti, John K. Salmon, Yibing Shan, and David E. Shaw, "Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters," *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing (SC06)*, Tampa, Florida, November 11–17, 2006.
4. Berk Hess, Carsten Kutzner, David van der Spoel, and Erik Lindahl, "GROMACS 4: Algorithms for Highly Efficient, Load-Balanced and Scalable Molecular Simulations," *Journal of Chemical Theory and Computation*, vol. 4, no. 3, March 11, 2008, 435–447.
5. Abhinav Bhatelé, Sameer Kumar, Chao Mei, James C. Phillips, Gengbin Zheng, and Laxmikant V. Kalé, "Overcoming Scaling Challenges in Biomolecular Simulations Across Multiple Platforms," *Proceedings of the IEEE International Symposium on Parallel and Distributed Processing (IPDPS 2008)*, Miami, Florida, April 14–18, 2008.