

# Performance Analysis of MPI and Hybrid MrBayes Computational Biology Code on Multicore Systems

Holly K. Williams  
hollywil@uab.edu

**Abstract**—This research focuses on the performance analysis of the MPI-only and hybrid versions of the MrBayes computational biology program on multicore systems. We conducted experiments using two different multicore systems—one from the Virginia Polytechnic Institute and State University, Dori, and the other from the Texas A&M University Supercomputing Facility, Hydra. We used strong scaling and weak scaling to collect performance data for seven different datasets. We used three performance metrics to analyze performance: execution time, efficiency, and speedup. Also we collected energy consumption data for MrBayes for both versions. The results of this research show that the hybrid version has a higher execution than the MPI-only version. The results also show that the hybrid version experiences more speedup than the MPI version and that the hybrid version scales better than the MPI version. The energy consumption data collected shows that the hybrid version has higher energy consumption than the MPI version.

## I. INTRODUCTION

The current trend in parallel computing has shifted towards hierarchically composed multicore systems. Programs that take a very long time to execute have been a driving force in parallel computing. These types of programs can be made parallel using a parallel programming paradigm such as MPI or OpenMP, and execution time will be reduced. Another method that suggests lower execution time is the combining of two different paradigms in one application. Combining two different techniques such as OpenMP and MPI in one program to make a hybrid application allows users to exploit multiple levels of parallelism on a multicore system.

OpenMP [2] is used for shared memory parallelism. It follows a fork and join model where initially there is one master thread. When the master thread reaches a parallel region, the thread will fork off many other threads to execute that particular area in parallel.

OpenMP is applied to a program using directives. It is very popular for fine-grained parallelism, particularly large for loops. These loops can be divided amongst other threads so that the problem can be worked on at the same time. With OpenMP and shared memory, we must watch out for data dependencies and the scope of variables to make sure that in the end, we obtain the correct results.

MPI [1] is the interface used for message passing parallelism and distributed memory. MPI uses processes to execute parallel regions. Since MPI is dealing with distributed memory, each process originally only knows its own data. A process does not have access to another process's data. To communicate this data, processes must pass their data using messages. MPI is useful for coarse-grained parallelism.

Hybridization with OpenMP and MPI in one application allows for multiple levels of parallelism when using multicore clusters. OpenMP can be used per node for shared memory fine-grained parallelism amongst the cores on the node. MPI can be used for distributed memory coarse-grained parallelism between the nodes themselves.

This research focuses on the performance analysis of the MPI-only and hybrid versions of the MrBayes computational biology program on multicore systems. We conducted experiments using two different multicore systems. Two different methods of testing were conducted. Weak scaling was used on one system. Strong scaling was used on two systems. Execution time, speedup, and efficiency were used to compare the performance of both versions. Also we collected energy consumption data for MrBayes for both versions.

The remainder of this paper is organized as follows. Section II discusses the MrBayes application. Section III describes the experimental setup and the multicore systems used. Section IV shows the results obtained from the experiment. Section V is conclusions.

## II. MRBAYES COMPUTATIONAL BIOLOGY PROGRAM

MrBayes [5] is a computational biology program used for Bayesian inference of phylogeny. The application takes as input a file containing a number of taxa and each taxon's data. The program then based on the taxa's data will construct a tree depicting the relationship of the organisms. There are many possible phylogenies to choose from. Bayesian inference is used to find the best tree to show this relationship.

$$Pr[Tree|Data] = \frac{Pr[Data|Tree] \times Pr[Tree]}{Pr[Data]}$$

$Pr[Data]$  involves a summation over all possible trees and, for each tree, integration over all possible combinations of branch lengths and parameter values [4].

The posterior probability

$$\frac{Pr[Data|Tree] \times Pr[Tree]}{Pr[Data]}$$

is impossible to calculate analytically. MrBayes uses Markov chain Monte Carlo simulation to approximate the posterior probability of trees [4]. MrBayes is a worthy target for parallel processing because the program does many computations and takes a very long time to execute. Evaluating real world data could potentially be very time consuming without parallelization thus rendering the application useless. Making the program parallel produces a lower execution time and produces results in an amount of time that is reasonable.

Both versions of MrBayes apply MPI to the Markov chains. The total number of chains is divided amongst the number of processes. The hybrid version is based on the MPI version. OpenMP is applied to parallelize large for loops in only one source file, mcmc, where the Monte Carlo chaining is handled, so OpenMP is not used as much as MPI in the application. The flow of the program is:

- Read dataset and initialize
  - The program reads in a dataset. The dataset specifies the number of taxa in the file. The dataset also contains the names of each taxon and their data
- Set parameters
  - set number of generations and number of chains
- Distribute number of chains amongst MPI processes
- Run analysis
  - run analysis for the set number of chains for the set number of generations

Seven datasets tested

| Dataset   | Number of Taxa |
|-----------|----------------|
| Adh       | 54             |
| Avian     | 89             |
| Bglobin   | 17             |
| Cynmix    | 32             |
| Anolis    | 30             |
| Primates  | 12             |
| Replicase | 9              |

Table 1

Table 1 shows the 7 datasets and their corresponding number of taxa

### III. EXPERIMENTAL SETUP

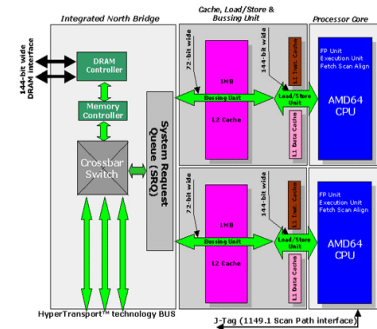
The two multicore systems used were Dori, a machine from the Virginia Polytechnic Institute and State University, and Hydra [3] from the Texas A&M Supercomputing Facility. Dori uses an AMD Opteron processor chip. The hierarchical organization of Dori is 8 nodes with 2 dual core chips each so there are 32 cores. The makeup of Hydra is 52 nodes with 8 dual core Power5+ chips each for a total of 832 cores. The specifications and comparison of the two systems can be seen in Table 2.

|                 | Dori                      | Hydra                        |
|-----------------|---------------------------|------------------------------|
| Number of Nodes | 8                         | 52                           |
| CPU's per node  | 4                         | 16                           |
| Cores per chip  | 2                         | 2                            |
| Total Cores     | 32                        | 832                          |
| CPU type        | 1.8 GHz<br>AMD<br>Opteron | 1.9GHz IBM<br>Power5+        |
| Memory per node | 6GB                       | 32GB/49 nodes<br>64GB/3nodes |

Table 2 Dori and Hydra Specifications

Figure 1 illustrates the architecture of a dual core chip on Dori. Figure 2 shows the architecture of a dual core chip on the Hydra cluster.

Data Flow view of the AMD Opteron™ Processor  
Dual core Model 100



The dual-core Opteron shares a set of three HyperTransport links and a dual-memory controller. The memory system is part of a NUMA architecture when utilized in a multiple-processor environment.

Figure 1 Dori AMD Opteron dual core chip

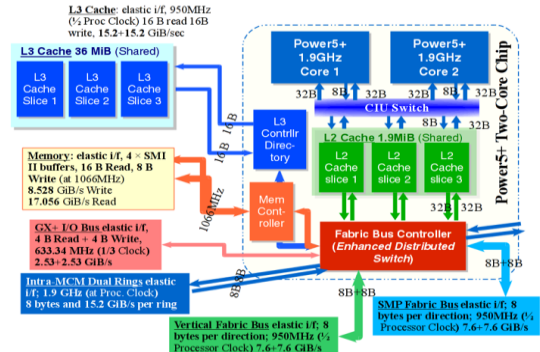


Figure 2 Power 5+ Dual Core Chip on Hydra

When executing hybrid MrBayes on Dori, shared memory parallelism is used on the 4 cores on each node, and MPI is used between the 8 nodes. With Hydra, shared memory parallelism is used on the 16 cores per node, and MPI is used between the nodes.

MrBayes was executed on both multicore systems using both versions. Each data set was run on a different number of cores for 10,000 generations using a different number of chains.

On Dori, we did strong scaling with MrBayes. For each dataset, 32 chains total were used. This number of chains was

chosen because Dori has 32 cores and with this number of chains, each chain could be distributed to one core.

We did strong and weak scaling on Hydra. For the weak scaling, 16 chains per node were used. For strong scaling, 512 chains were used.

Dori was used to collect energy consumption data for MrBayes. Energy consumption on Dori is collected using the PowerPack software. The system has an external meter connected to the power supply and cables connected to each component to detect energy consumption and power data per node.

For the collection of energy data, we ran one dataset, primates, using both versions and collected energy consumption data.

#### IV. PERFORMANCE ANALYSIS AND COMPARISON

To compare the performance of hybrid MrBayes to that of the MPI-only version, we use three performance metrics. Execution time is compared on each number of cores for each dataset. The speedup compares the speedup that both versions experienced with increasing cores. The efficiency is also used to compare the efficiency of both versions when running on multiple cores. Seven datasets were run on Dori. Only six were run on Hydra because of memory issues.

##### A. Results on Dori

The following charts show results after executing on the Dori multicore system using 2, 4, 8, 16, and 32 cores with 32 chains each.

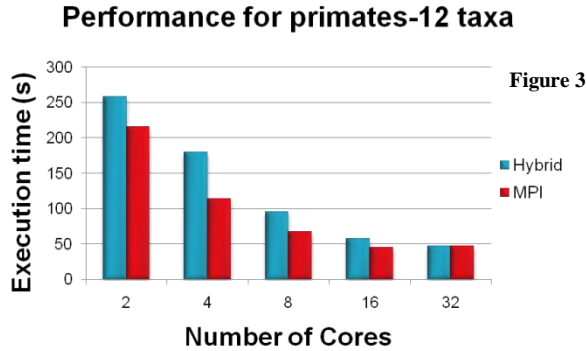


Figure 3 shows the execution times for primates on Dori

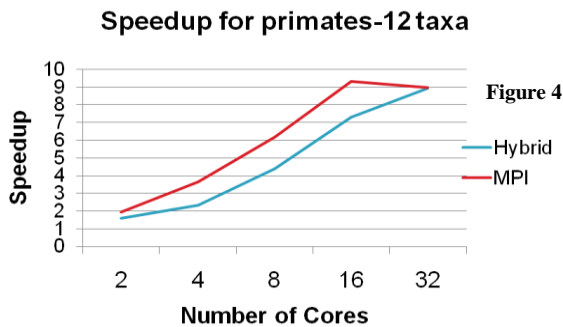


Figure 4 shows the speedup comparison for the primates dataset on Dori

Figure 3 and Figure 4 show performance data after executing one of the smaller datasets, primates, on Dori. Figure 5 shows the execution time on 5 different number of cores for the hybrid and MPI-only versions. The chart shows that the MPI-only version outperforms the hybrid version, but at 32 cores, the execution times are very close.

Figure 4 shows the speedup of both versions when executing the primates dataset. The results show that both versions experience speedup, however at 32 cores, the MPI-only version actually has a decrease in speedup while the hybrid version keeps experiencing increased speedup. The results show that 16 cores for the MPI version is the optimal number of cores to use when executing the primates dataset. The communication time when using 32 cores increases the execution time making it larger than that of 16 cores.

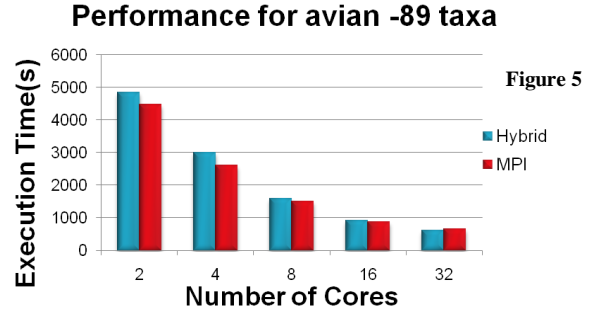


Figure 5 shows the execution times for avian on Dori

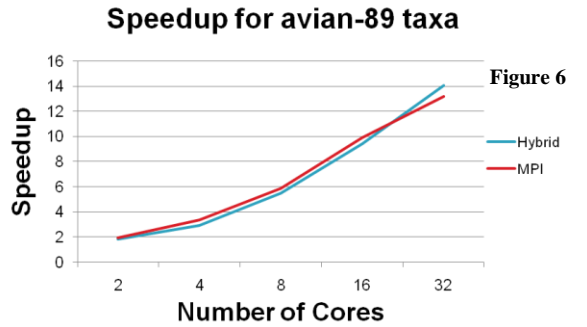


Figure 6 shows the speedup for avian on Dori

Figure 5 and Figure 6 show the performance data on Dori after running the largest dataset avian. These results show that MPI MrBayes has lower execution time than the hybrid version for up to 16 cores. At 32 cores, the hybrid version has a lower execution time. The speedup data shows that both versions experience increased speedup. At 32 cores, the speedup of the hybrid version is larger.

**Performance for anolis-30 taxa**

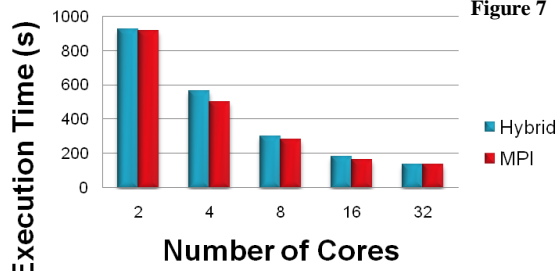


Figure 7 shows the execution times for anolis on Dori

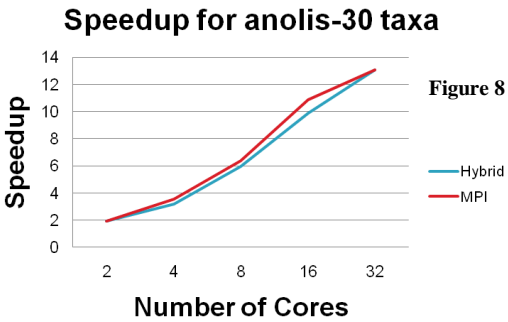


Figure 8 shows the speedup for anolis on Dori on Dori

Figures 7 and 8 show the results after executing the anolis dataset. The hybrid version is slower than the MPI version. Figure 8 presenting the speedup shows that both versions experience increased speedup. At 32 cores, the two versions have close speedup values.

## B. Results on Hydra

On Hydra there were memory issues with all but two of the datasets. We could only run the two smallest datasets successfully. The amount of memory needed is proportional to the number of taxa in the dataset and also the number of chains needed. On Hydra, the number of chains was increased to 512 total, and this caused a problem with larger datasets.

To be able to execute on Hydra, we performed weak scaling. The work load, number of chains, increased as the number of cores increased. To keep the work load constant, we used sixteen chains per node. Sixteen was chosen because there are sixteen cores per node on Hydra.

The following data shows results on Hydra executing on up to 256 cores.

**Performance for primates-12 taxa Weak Scaling**

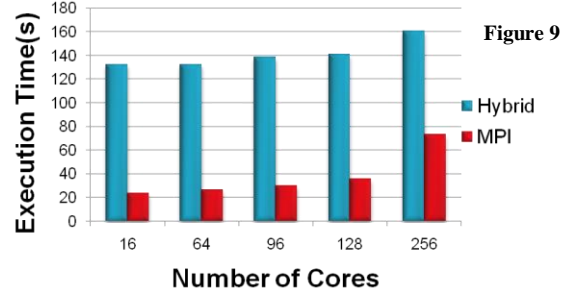


Figure 9 Execution time for primates on Hydra with weak scaling

**Performance for adh-54 taxa Weak Scaling**

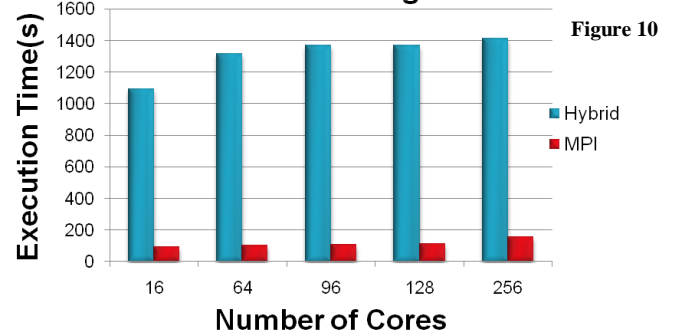


Figure 10 Execution time for adh on Hydra with weak scaling

Figure 9 and Figure 10 show results after doing weak scaling with the primates dataset and the adh dataset on Hydra. These results show that hybrid MrBayes has a higher execution time than the MPI-only version.

Figure 11 shows the results of executing the primates dataset on Hydra using the strong scaling technique. We were able to do strong scaling on this dataset since it is smaller with only 12 taxa. The results show that the hybrid version is much slower than the MPI version, but at 256 the execution times begin to get closer. Figure 12 shows the speedup of the primates dataset during the same run. This chart shows that hybrid experiences substantial speedup, whereas, MPI levels off. This is reflected in the chart of the execution times. The MPI version's execution time did not change substantially, but the hybrid version's execution time steadily decreased. The MPI version's execution time sees no substantial decrease in execution time suggesting that the additional cores are not necessary and actually introduce communication overhead.

### Performance for primates-12 taxa Strong Scaling

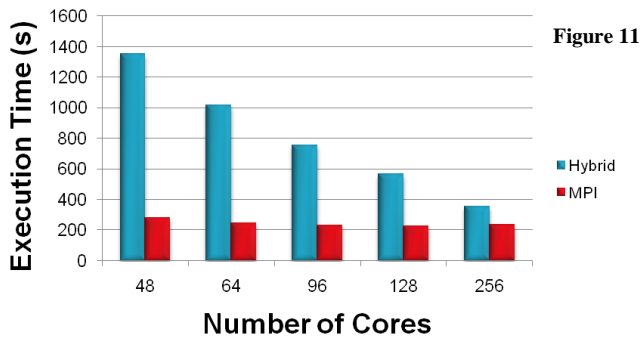


Figure 11 Execution times for primates on Hydra with strong scaling

### Speedup for primates-12 taxa Strong Scaling

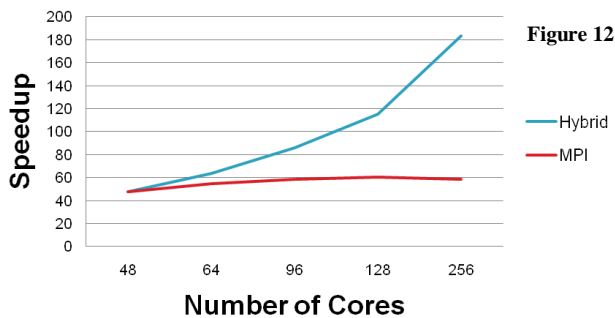


Figure 12 Speedup for primates with strong scaling on Hydra

#### C. Energy Consumption on Dori

Table 3 shows energy consumption data collected with the primates dataset using Dori with 32 chains. The results in Table 3 show that the hybrid version has higher energy consumption than the MPI-only version for all number of cores except 32. With 32 cores, memory, hard drive, and motherboard all show lower energy values for hybrid MrBayes than that of MPI MrBayes. Also at 32 cores, hybrid MrBayes has a lower execution time.

The energy consumption data also show that 4 cores is the optimal number of cores for low energy consumption for both versions even though the time is higher than for 32 cores.

Tables 3-17 show the results for all datasets used.

#### V. CONCLUSIONS

The results of this research show that the MPI version of this application has a lower execution time than the hybrid version. The program benefits more from additional MPI processes than OpenMP threads. This is because OpenMP is not used as extensively as MPI. The different Markov chains

are distributed amongst the MPI processes. For example on Hydra—which has 16 cores per node—using strong scaling with the MPI version and 128 cores, the 512 chains were divided amongst the cores so that each core was given 4 chains. With the hybrid version in the same case, there are only 8 MPI processes to use. In the hybrid version each of the 8 MPI processes receives 64 Markov chains.

The results show that the hybrid version experiences a larger speedup than the MPI version. This suggests that the hybrid version would scale better on a larger multicore system. This is because the number of nodes increases as the number of cores increases, and the hybrid version uses nodes for MPI processes in contrast with the MPI-version which uses each core for MPI processes. So as the number of cores becomes larger the MPI-only version will begin to experience communication overhead which will increase the execution time, but the hybrid version will have more MPI processes to distribute the Markov chains amongst and will experience decreased execution time. This can be seen from the primates dataset using strong scaling on Hydra. The speedup and execution times show that with added cores, the hybrid version experiences increased speedup, whereas the MPI version stays almost the same. The hybrid version would experience lower execution time as well if more OpenMP directives were applied to other parts of the program rather than in one source file.

#### ACKNOWLEDGEMENT

I would like to thank my mentors, Dr. Valerie Taylor, Dr. Xingfu Wu, and Charles Lively for all their help and guidance. I would also like to thank the CRA-W DREU program, the sponsors of this research.

#### REFERENCES

- [1] Message Passing Interface. <http://www.mcs.anl.gov/research/projects/mpi/>. 3 Aug 2010
- [2] OpenMP. <http://openmp.org/wp/>. 3 Aug 2010
- [3] Texas A&M Supercomputing Facility <http://sc.tamu.edu/systems/hydra/hardware.php> 3 Aug 2010
- [4] Huelsenbeck, J.P and F. Ronquist. 2004. Bayesian Analysis of Molecular Evolution using MrBayes
- [5] Huelsenbeck, J. P. and F. Ronquist. 2001. MRBAYES: Bayesian inference of phylogeny. *Bioinformatics* 17:754-755.

**Energy Consumption Primates-12 taxa on Dori**

| No. of Cores | Time (s) | System (kJ) | CPU (kJ)  | Memory (kJ) | Hard Drive (kJ) | Motherboard (kJ) |
|--------------|----------|-------------|-----------|-------------|-----------------|------------------|
| 2-hybrid     | 254.154  | 45.963065   | 27.30582  | 5.256297    | 2.103262        | 3.481821         |
| 2-mpi        | 214.611  | 39.263947   | 23.852611 | 3.997544    | 1.809655        | 2.966753         |
| 4-hybrid     | 178.418  | 37.648057   | 22.565793 | 3.875879    | 1.518767        | 2.486724         |
| 4-mpi        | 113.77   | 24.91126    | 14.459266 | 2.647337    | 0.972663        | 1.587298         |
| 8-hybrid     | 94.565   | 39.304558   | 22.842574 | 3.83381     | 1.545488        | 2.580222         |
| 8-mpi        | 67.301   | 28.043112   | 15.750636 | 2.833562    | 1.154268        | 1.855912         |
| 16-hybrid    | 56.887   | 45.89634    | 25.943368 | 4.370968    | 1.896912        | 3.085648         |
| 16-mpi       | 43.887   | 34.622904   | 18.566292 | 3.34392     | 1.378972        | 2.38708          |
| 32-hybrid    | 46.099   | 72.668664   | 40.61384  | 6.896368    | 3.023776        | 5.036632         |
| 32-mpi       | 46.651   | 71.06504    | 39.91648  | 7.145648    | 3.3108          | 5.181408         |

**Table 3 Energy Consumption Hybrid MrBayes vs. MPI MrBayes from Dori**

| Primates on Dori |                   |           |            | Anolis on Dori  |                   |            |            |
|------------------|-------------------|-----------|------------|-----------------|-------------------|------------|------------|
| Number of Cores  | Execution Time(s) | Speedup   | Efficiency | Number of Cores | Execution Time(s) | Speedup    | Efficiency |
| 2-hybrid         | 258.38            | 1.607129  | 0.8036     | 2-hybrid        | 925.85            | 1.93260247 | 0.96630    |
| 2-mpi            | 215.17            | 1.9299    | 0.9649     | 2-mpi           | 917.03            | 1.95119    | 0.97560    |
| 4-hybrid         | 179.27            | 2.3163385 | 0.5791     | 4-hybrid        | 562.71            | 3.17979    | 0.79495    |
| 4-mpi            | 113.76            | 3.6502    | 0.9126     | 4-mpi           | 501.72            | 3.56633    | 0.89158    |
| 8-hybrid         | 94.99             | 4.3715128 | 0.5464     | 8-hybrid        | 300.37            | 5.95699    | 0.74462    |
| 8-mpi            | 67.44             | 6.1573    | 0.7697     | 8-mpi           | 279.71            | 6.39698    | 0.79962    |
| 16-hybrid        | 57.04             | 7.279979  | 0.4550     | 16-hybrid       | 180.98            | 9.88673    | 0.61792    |
| 16-mpi           | 44.57             | 9.3168    | 0.5823     | 16-mpi          | 164.31            | 10.88978   | 0.68061    |
| 32-hybrid        | 46.51             | 8.9281875 | 0.2790     | 32-hybrid       | 136.7             | 13.08925   | 0.40904    |
| 32-mpi           | 46.28             | 8.9725583 | 0.2804     | 32-mpi          | 136.99            | 13.06154   | 0.40817    |

**Table 4 Performance data for primates on Dori****Table 5 Performance data for anolis on Dori**

| Replicase on Dori |                   |          |            | Adh on Dori     |                   |          |            |
|-------------------|-------------------|----------|------------|-----------------|-------------------|----------|------------|
| Number of Cores   | Execution Time(s) | Speedup  | Efficiency | Number of Cores | Execution Time(s) | Speedup  | Efficiency |
| 2-hybrid          | 307.85            | 1.306903 | 0.65345    | 2-hybrid        | 1146.35           | 1.242718 | 0.621359   |
| 2-mpi             | 207.10            | 1.94268  | 0.97134    | 2-mpi           | 728.28            | 1.956102 | 0.978051   |
| 4-hybrid          | 263.39            | 1.52751  | 0.38188    | 4-hybrid        | 1000.65           | 1.423665 | 0.355916   |
| 4-mpi             | 110.19            | 3.65124  | 0.91281    | 4-mpi           | 416.06            | 3.424001 | 0.856000   |
| 8-hybrid          | 132.97            | 3.02572  | 0.37822    | 8-hybrid        | 544.65            | 2.615606 | 0.326951   |
| 8-mpi             | 63.73             | 6.31304  | 0.78913    | 8-mpi           | 237.47            | 5.999031 | 0.749879   |
| 16-hybrid         | 77.46             | 5.19404  | 0.32463    | 16-hybrid       | 332.97            | 4.278433 | 0.267402   |
| 16-mpi            | 42.69             | 9.42446  | 0.58903    | 16-mpi          | 147.36            | 9.667413 | 0.604213   |
| 32-hybrid         | 56.38             | 7.13604  | 0.22300    | 32-hybrid       | 223.3             | 6.379713 | 0.199366   |
| 32-mpi            | 45.59             | 8.82496  | 0.27578    | 32-mpi          | 114.3             | 12.4636  | 0.389488   |

**Table 6 Performance data for replicase on Dori****Table 7 Performance data for adh on Dori**

| Cynmix on Dori  |                   |          |            |
|-----------------|-------------------|----------|------------|
| Number of Cores | Execution Time(s) | Speedup  | Efficiency |
| 2-hybrid        | 1223.81           | 1.631479 | 0.815739   |
| 2-mpi           | 904.27            | 2.207991 | 1.103995   |
| 4-hybrid        | 994.69            | 2.007279 | 0.50182    |
| 4-mpi           | 490.31            | 4.072158 | 1.01804    |
| 8-hybrid        | 517.37            | 3.859172 | 0.482397   |
| 8-mpi           | 312.62            | 6.386731 | 0.798341   |
| 16-hybrid       | 303.51            | 6.578432 | 0.411152   |
| 16-mpi          | 186.23            | 10.72126 | 0.670079   |
| 32-hybrid       | 193.21            | 10.33394 | 0.322936   |
| 32-mpi          | 155.66            | 12.8268  | 0.400838   |

**Table 8** Performance data for cynmix on Dori

| Avian on Dori   |                   |          |            |
|-----------------|-------------------|----------|------------|
| Number of Cores | Execution Time(s) | Speedup  | Efficiency |
| 2-hybrid        | 4834.53           | 1.799341 | 0.899671   |
| 2-mpi           | 4477.48           | 1.942827 | 0.971414   |
| 4-hybrid        | 3000.02           | 2.899637 | 0.724909   |
| 4-mpi           | 2608.69           | 3.334612 | 0.833653   |
| 8-hybrid        | 1579.85           | 5.5062   | 0.688275   |
| 8-mpi           | 1488.91           | 5.842509 | 0.730314   |
| 16-hybrid       | 922.3             | 9.431823 | 0.589489   |
| 16-mpi          | 877.38            | 9.914712 | 0.619669   |
| 32-hybrid       | 618.76            | 14.05871 | 0.439335   |
| 32-mpi          | 659.04            | 13.19946 | 0.412483   |

**Table 9** Performance data for avian on Dori

| Bglobin on Dori |                   |            |            |
|-----------------|-------------------|------------|------------|
| Number of Cores | Execution Time(s) | Speedup    | Efficiency |
| 2-hybrid        | 4259.39           | 2.23601267 | 1.11800633 |
| 2-mpi           | 4669.53           | 2.03961641 | 1.0198082  |
| 4-hybrid        | 2771.49           | 3.43643672 | 0.85910918 |
| 4-mpi           | 2722.62           | 3.49811946 | 0.874530   |
| 8-hybrid        | 1456.01           | 6.54119821 | 0.81764978 |
| 8-mpi           | 1539.55           | 6.186256   | 0.773282   |
| 16-hybrid       | 864.83            | 11.0126268 | 0.68828917 |
| 16-mpi          | 983.56            | 9.683243   | 0.605203   |
| 32-hybrid       | 626.83            | 15.193992  | 0.47481225 |
| 32-mpi          | 690.18            | 13.7993712 | 0.43123035 |

**Table 10** Performance data for bglobin on Dori

| Primates on Hydra Weak Scaling<br>Execution Time(s) |        |       |
|---|--------|-------|
| Number of Cores                                     | Hybrid | MPI   |
| 16  | 132.05 | 23.22 |
| 64  | 132.05 | 26.08 |
| 96  | 138.61 | 29.58 |
| 128   | 140.59 | 35.43 |
| 256   | 160.55 | 73.38 |

**Table 11** Performance data for primates on Hydra

| Adh on Hydra Weak Scaling<br>Execution Time(s) |         |        |
|--|---------|--------|
| Number of Cores                                | Hybrid  | MPI    |
| 16   | 1093.74 | 92.42  |
| 64   | 1316.73 | 100.02 |
| 96   | 1368.99 | 105.04 |
| 128  | 1367.41 | 111.96 |
| 256  | 1414.11 | 155.23 |

**Table 12** Performance data for adh on Hydra

| Replicase on Hydra Weak Scaling<br>Execution Time |        |       |
|---|--------|-------|
| Number of Cores                                   | Hybrid | MPI   |
| 16  | 261.16 | 21    |
| 64  | 285.47 | 25.52 |
| 96  | 291.27 | 27.96 |
| 128   | 295.06 | 33.78 |
| 256   | 319.34 | 72.84 |

**Table 13** Performance data for replicase on Hydra

| Cynmix on Hydra Weak Scaling<br>Execution Time(s) |        |        |
|---|--------|--------|
| Number of Cores                                   | Hybrid | MPI    |
| 16  | 778.87 | 104.35 |
| 64  | 858.44 | 114.75 |
| 96  | 884.95 | 121.75 |
| 128   | 888.95 | 131.37 |
| 256   | 934.48 | 177.39 |

**Table 14** Performance data for cynmix on Hydra

| Anolis on Hydra Weak Scaling<br>Execution Time(s) |        |        |
|---|--------|--------|
| Number of Cores                                   | Hybrid | MPI    |
| 16  | 273.92 | 94.26  |
| 64  | 315.33 | 101.22 |
| 96  | 327.94 | 104.75 |
| 128   | 330.45 | 112.97 |
| 256   | 354.74 | 154.29 |

**Table 15** Performance data for anolis on Hydra

| Bglobin on Hydra Weak Scaling<br>Execution Time(s) |        |        |
|--|--------|--------|
| Number of Cores                                    | Hybrid | MPI    |
| 16   | 596.18 | 255.29 |
| 64   | 711.13 | 271.96 |
| 96   | 736.16 | 283.32 |
| 128  | 742.76 | 293.92 |
| 256  | 775.13 | 345.34 |

**Table 16** Performance data for bglobin on Hydra

| Primates on Hydra Strong Scaling<br>Execution Time(s) |         |        |
|---|---------|--------|
| Number of Cores                                       | Hybrid  | MPI    |
| 48  | 1353.51 | 279.7  |
| 64  | 1014.44 | 244.64 |
| 96  | 755.75  | 230.18 |
| 128   | 564.64  | 221.99 |
| 256   | 353.48  | 232.23 |

**Table 17** Performance data for anolis on Hydra  
using strong scaling