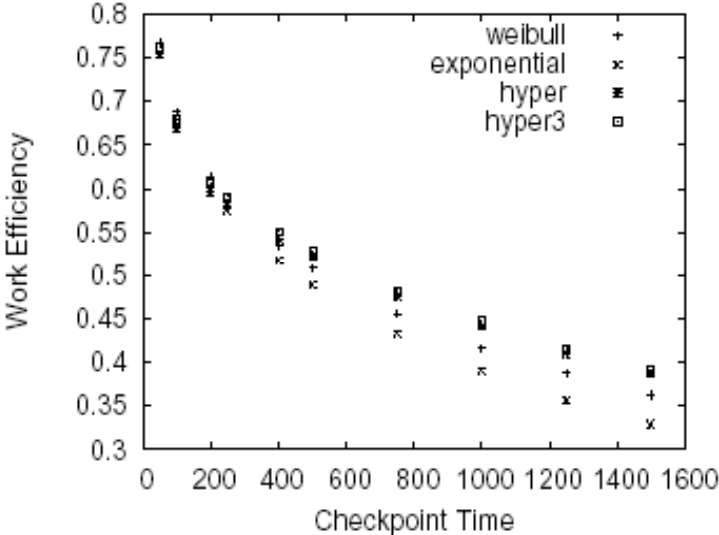
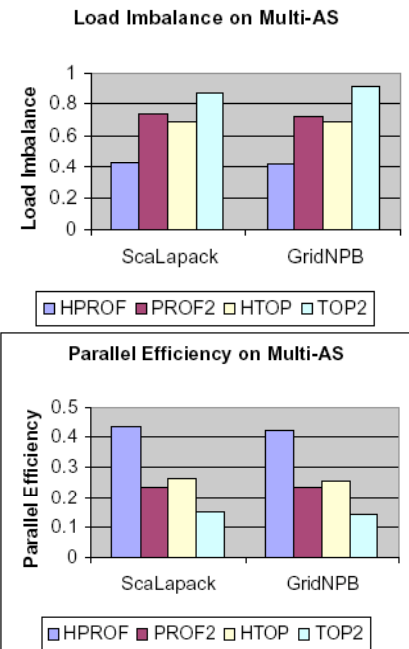


## Nuggets

<p><b>Award No:</b> CCR-0331654</p>	
<p><b>Project Title:</b> Virtual Grid Application Development Software (VGrADS)</p>	
<p><b>Investigators:</b> Ken Kennedy (PI), Fran Berman, Henri Casanova, Andrew Chien, Keith Cooper, Jack Dongarra, S. Lennart Johnsson, Carl Kesselman, Charles Koelbel, Daniel Reed, Richard Tapia, Linda Torczon, Richard Wolski</p>	
<p><b>Institution:</b> Rice University (lead institution), University of California at San Diego, University of California at Santa Barbara, University of Houston, University of North Carolina, University of Southern California / Information Sciences Institute, University of Tennessee</p>	<p><b>Description of Graphic Image:</b> Application run under workflow scheduler. Parallel tasks (in orange) are distributed across one or more clusters, while sequential tasks (in blue) run on a node with good connectivity to all clusters.</p>
<p><b>Project Description and Outcome</b> (Provide content for one or more of the following outcome goals)</p>	
<p><b>Ideas:</b> VGrADS researchers developed new methods for scheduling workflow applications (those with multiple components linked by data and control dependence) on distributed computational grids. This will allow much better performance for many scientific applications, in fields ranging from bioimaging (see graphic above) to climate modeling to genome mapping. Our workflow scheduler differs from others in wide use (e.g. Condor's DAGMAN system) by using information about later tasks to optimize both computation and communication performance simultaneously.</p> <p>Specifically, the system schedules components as a task graph. The computation cost for each component task is estimated by a hybrid performance model, composed of a hand-generated cost of floating-point computations (parameterized by type of processor) and an automated estimate of memory hierarchy costs (parameterized by cache sizes and other system characteristic). The communication cost between component tasks is estimated from latency and bandwidth information derived from NWS. We then apply heuristics (MIN-MIN, MIN-MAX, and Sufferage) to choose the best mapping of components to nodes.</p> <p>Recent experiments have demonstrated the scalability of this scheduler to multiple heterogeneous clusters. Our recent (albeit preliminary) experiments with the EMAN electron microscopy image package show that it gives up to 120% better performance than simple random scheduling, and 50% better than a more sophisticated (but still randomized) scheduling.</p>	

<b>Award No:</b> CCR-0331654	 <table border="1"> <caption>Approximate data points from the Work Efficiency vs Checkpoint Time plot</caption> <thead> <tr> <th>Checkpoint Time</th> <th>weibull (+)</th> <th>exponential (x)</th> <th>hyper (≡)</th> <th>hyper3 (□)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.75</td> <td>0.75</td> <td>0.75</td> <td>0.75</td> </tr> <tr> <td>100</td> <td>0.68</td> <td>0.68</td> <td>0.68</td> <td>0.68</td> </tr> <tr> <td>200</td> <td>0.62</td> <td>0.62</td> <td>0.62</td> <td>0.62</td> </tr> <tr> <td>400</td> <td>0.55</td> <td>0.55</td> <td>0.55</td> <td>0.55</td> </tr> <tr> <td>800</td> <td>0.48</td> <td>0.48</td> <td>0.48</td> <td>0.48</td> </tr> <tr> <td>1000</td> <td>0.45</td> <td>0.45</td> <td>0.45</td> <td>0.45</td> </tr> <tr> <td>1200</td> <td>0.42</td> <td>0.42</td> <td>0.42</td> <td>0.42</td> </tr> <tr> <td>1400</td> <td>0.40</td> <td>0.40</td> <td>0.40</td> <td>0.40</td> </tr> <tr> <td>1600</td> <td>0.38</td> <td>0.38</td> <td>0.38</td> <td>0.38</td> </tr> </tbody> </table>	Checkpoint Time	weibull (+)	exponential (x)	hyper (≡)	hyper3 (□)	0	0.75	0.75	0.75	0.75	100	0.68	0.68	0.68	0.68	200	0.62	0.62	0.62	0.62	400	0.55	0.55	0.55	0.55	800	0.48	0.48	0.48	0.48	1000	0.45	0.45	0.45	0.45	1200	0.42	0.42	0.42	0.42	1400	0.40	0.40	0.40	0.40	1600	0.38	0.38	0.38	0.38
Checkpoint Time		weibull (+)	exponential (x)	hyper (≡)	hyper3 (□)																																														
0		0.75	0.75	0.75	0.75																																														
100	0.68	0.68	0.68	0.68																																															
200	0.62	0.62	0.62	0.62																																															
400	0.55	0.55	0.55	0.55																																															
800	0.48	0.48	0.48	0.48																																															
1000	0.45	0.45	0.45	0.45																																															
1200	0.42	0.42	0.42	0.42																																															
1400	0.40	0.40	0.40	0.40																																															
1600	0.38	0.38	0.38	0.38																																															
<b>Project Title:</b> Virtual Grid Application Development Software (VGrADS)																																																			
<b>Investigators:</b> Ken Kennedy (PI), Fran Berman, Henri Casanova, Andrew Chien, Keith Cooper, Jack Dongarra, S. Lennart Johnsson, Carl Kesselman, Charles Koelbel, Daniel Reed, Richard Tapia, Linda Torczon, Richard Wolski																																																			
<b>Institution:</b> Rice University (lead institution), University of California at San Diego, University of California at Santa Barbara, University of Houston, University of North Carolina, University of Southern California / Information Sciences Institute, University of Tennessee																																																			
<b>Website:</b> <a href="http://hipersoft.cs.rice.edu/vgrads/">http://hipersoft.cs.rice.edu/vgrads/</a>	<b>Description of Graphic Image:</b> Simulation of optimal checkpointing in a Grid application. The simulation used actual traces of machine availability to model failures, and chose optimal checkpoint intervals based on the models shown for various checkpoint costs. Different availability models provide different efficiencies for a given checkpoint cost.																																																		
<b>Project Description and Outcome</b> <i>(Provide content for one or more of the following outcome goals)</i>																																																			
<p><i>Ideas:</i></p> <p>To support work on fault-tolerant applications, we have worked out a method for both modeling and predicting availability times, including the duration that a machine will run until it restarts (availability duration), for Grid resources. We do this by fitting live performance data to standard parametric models (Weibull, 2-phase hyperexponential, and 3-phase hyperexponential) using a maximum likelihood estimate (MLE) approach. Our results indicate that this enables us to predict availability duration with quantifiable confidence bounds and that these bounds can be used as conservative bounds on lifetime predictions.</p> <p>Based on this work, we have also developed a system for automatically determining an "optimal" checkpoint schedule for an application. The word "optimal" is in quotation marks because it relies on the parametric availability model (which is chosen above), making the optimality depend on the goodness of the model fit. In practice, however, the model fits are sufficiently good on a variety of systems. Work continues to validate the quality of models; to empirically evaluate the checkpoint efficiency (difficult because the failure times are so far apart), and extend this work to use non-parametric statistics.</p>																																																			

<p><b>Award No:</b> CCR-0331654</p>	 <p><b>Load Imbalance on Multi-AS</b></p> <table border="1"> <thead> <tr> <th>Benchmark</th> <th>HPROF</th> <th>PROF2</th> <th>HTOP</th> <th>TOP2</th> </tr> </thead> <tbody> <tr> <td>ScaLapack</td> <td>0.42</td> <td>0.72</td> <td>0.68</td> <td>0.85</td> </tr> <tr> <td>GridNPB</td> <td>0.42</td> <td>0.72</td> <td>0.68</td> <td>0.85</td> </tr> </tbody> </table> <p><b>Parallel Efficiency on Multi-AS</b></p> <table border="1"> <thead> <tr> <th>Benchmark</th> <th>HPROF</th> <th>PROF2</th> <th>HTOP</th> <th>TOP2</th> </tr> </thead> <tbody> <tr> <td>ScaLapack</td> <td>0.42</td> <td>0.22</td> <td>0.25</td> <td>0.15</td> </tr> <tr> <td>GridNPB</td> <td>0.42</td> <td>0.22</td> <td>0.25</td> <td>0.15</td> </tr> </tbody> </table>	Benchmark	HPROF	PROF2	HTOP	TOP2	ScaLapack	0.42	0.72	0.68	0.85	GridNPB	0.42	0.72	0.68	0.85	Benchmark	HPROF	PROF2	HTOP	TOP2	ScaLapack	0.42	0.22	0.25	0.15	GridNPB	0.42	0.22	0.25	0.15
Benchmark	HPROF	PROF2	HTOP	TOP2																											
ScaLapack	0.42	0.72	0.68	0.85																											
GridNPB	0.42	0.72	0.68	0.85																											
Benchmark	HPROF	PROF2	HTOP	TOP2																											
ScaLapack	0.42	0.22	0.25	0.15																											
GridNPB	0.42	0.22	0.25	0.15																											
<p><b>Project Title:</b> Virtual Grid Application Development Software (VGrADS)</p>																															
<p><b>Investigators:</b> Ken Kennedy (PI), Fran Berman, Henri Casanova, Andrew Chien, Keith Cooper, Jack Dongarra, S. Lennart Johnsson, Carl Kesselman, Daniel Reed, Richard Tapia, Linda Torczon, Richard Wolski</p>																															
<p><b>Institution:</b> Rice University (lead institution), University of California at San Diego, University of California at Santa Barbara, University of Houston, University of North Carolina, University of Southern California / Information Sciences Institute, University of Tennessee</p>																															
<p><b>Website:</b> <a href="http://hipersoft.cs.rice.edu/vgrads/">http://hipersoft.cs.rice.edu/vgrads/</a></p>	<p><b>Description of Graphic Image:</b> Performance of simulations of 2 benchmarks (ScaLAPACK and the NAS Parallel Benchmarks) on a large Grid (10,000 hosts attached to 100 Autonomous Systems of 200 routers). The top figure shows significant improvements in load balance, while the lower shows similar improvements in efficiency of the simulation.</p>																														
<p><b>Project Description and Outcome</b> <i>(Provide content for one or more of the following outcome goals)</i></p>																															
<p><i>Ideas and Tools:</i> Large-scale network simulation is an important technique for studying the dynamic behavior of Grid applications, as well as other aspects of networks. Simulating Grid applications requires both large scale, which in turn requires advanced strategies for load-balancing the simulation itself, and realism, which in turn requires detailed models within the simulation.</p> <p>To meet the former need, we developed a new hierarchical profile-based load balance technique (HPROF) for our MicroGrid system. The key idea is to cluster network nodes and explicitly control the tradeoff between simulation efficiency and available parallelism. The effect is to produce robust and superior performance for large-scale networks (20,000 simulated routers), improving load balance by 40% and simulation time by 50% on our 128-node cluster.</p> <p>To meet part of the latter need, we developed new methods for generating large network topologies. The basis of our methods is to apply publicly-known BGP policies in the process of creating topologies, thus generating realistic routing of traffic in the topology. This is a significant difference from previous approaches, which typically generated a realistic physical topology but (unrealistically) assumed flat shortest-path routing.</p>																															