

# We Didn't Start the Fire: Using Agent-Directed Thermal Modeler to Keep Servers Cool

Jonathan Hood, Timothy Scott, Jiao Yu, Xiao Qin, Levent Yilmaz

Department of Computer Science and Software Engineering

Auburn University

Auburn, AL 36849

Email: {jwh0011, tas0015, jzy0012, xqin, yilmale}@auburn.edu

Drew Hamilton

Department of Computer Science and Engineering

PO Box 6343

Mississippi State University, MS 39762

Email: hamilton@research.msstate.edu

**Abstract**—As energy use by datacenters has risen over the years, the costs required to run a datacenter have substantially increased. Several algorithms for thermal management and thermal-aware job placement exist, such as [1], [2], and [3]; however, choosing the scheme that will most efficiently cool a datacenter can be challenging. Thermal models offer a great solution to help choose which algorithm will perform best by juxtaposing different thermal-aware algorithms. When the temperature can be observed over all servers through simulated steps, one can decipher the differences and advantages of one thermal-aware algorithm over another. Existing thermal modelers, however, can be slow and may take a while to learn to use. When one wishes to compare several thermal models, waiting for hours for the result of one thermal-aware algorithm may mean that not as many algorithms can be compared.

Agent-Directed Thermal Modeler (ADTM) provides a solution that has a low learning curve and still produces visualizations of datacenters quickly. It is aimed at being easy to configure while still producing very meaningful data through graphs and images. There are very few parameters, making setup much easier - one simply has to configure a few settings such as initial number of jobs, job gain, and which thermal-aware algorithm to use. Once these settings are configured, the simulation can be run in a matter of seconds. A single time-step of the simulation takes milliseconds. The graphical and pictorial output of ADTM can then be used to determine which thermal-aware algorithm works best for a given datacenter in a much shorter time than other thermal-modelers. ADTM is used to compare XInt-GA to random job placement in order to show that a 5% increase in energy savings is expected in an overloaded datacenter. This simulation for both algorithms takes only a few seconds, so many thermal-aware algorithms can be compared quickly in order to determine the most effective and realistic algorithm that can be chosen to cool a datacenter.

**Keywords**—agent-directed; thermal model; thermal-aware

## I. INTRODUCTION

The energy usage of datacenters has been a growing problem as more data are stored. With this increase in data storage, more jobs need to be processed, more nodes are needed to complete these jobs, and, as a result, an increasing amount of energy is required. The energy consumed by IT equipment alone, however, only accounts for part of the increase. The cooling costs for IT equipment can account for 50 percent of energy consumption in a datacenter [4]. How can one save on cooling energy costs without destroying hardware through overheating?

Intelligent thermal-aware algorithms have been developed that help keep servers cool. In addition, some algorithms attempt to minimize energy consumption with little to no impact on performance. This helps in solving the issue of cooling costs, but how can one determine exactly how a specific algorithm will affect a specific datacenter? With so many algorithms available, choosing the proper algorithm for a datacenter can be a difficult process. Simply reading several papers can take a long time, and comparing these algorithms against each other can be nearly impossible without some form of modeling. This is why ADTM was developed: to allow datacenter operators to quickly visualize several thermal-aware algorithms and choose which will best keep a datacenter cool. Using thermal models like those produced by ADTM can visually show how a given scheme will work in a particular datacenter through a step-by-step simulation. These models will help lower cooling costs and prevent "disk-o inferno".

## II. MOTIVATION

As has been mentioned already, several thermal-aware algorithms exist in order to efficiently place jobs on nodes in order to prevent the overheating of specific nodes. This allows a more uniform heating of the datacenter which, in turn, means the HVAC does not need to be turned on until all nodes need to be cooled (to prevent unnecessary cooling of nodes that are already cool). Visualizing these algorithms and comparing them can be a daunting task. Some existing modelers exist, but they can have a high learning curve and take a lot of time to provide a simulation.

ADTM aims to simplify the process. There are only a few parameters to input into ADTM in order to run a simulation. This is much less-detailed and easier to configure than the amount of time that one might use in order to learn how to use a CFD simulator such as Flovent. In addition, each step in the simulation takes milliseconds, meaning the entire simulation for several timesteps will take minutes. These issues with existing thermal modelers provided the motivation in order to create ADTM.

## III. RELATED WORK

The motivation for this report helped determine existing work to use in order to aid in the development of ADTM.

These two algorithms/concepts helped in the design of this thermal modeler: Von Neumann neighborhoods and XInt-GA.

#### A. Von Neumann Neighborhoods

The Von Neumann Neighborhood is a way of representing neighbors around a node. The nodes are represented as a rectangular matrix [5]. Each node is a certain distance away from its neighboring nodes. For example, a node immediately above, below, to the right, or to the left of a node is a distance of one unit away. Two units above, below, to the right, or to the left would be two units away, and so on.

Closer neighbors in the Von Neumann neighborhood have a greater impact on a node's temperature than nodes that are further away. A hot node surrounded by cool nodes will be cooled down, and a cool node surrounded by hot nodes will be heated up. ADTM uses the Von Neumann neighborhood in order to determine the thermal impact of neighboring nodes (a detailed description is in Numerical Description section of this report). For speed considerations, ADTM considers neighbors that are one and two units away. As more neighbors are considered, the thermal impact will be more accurate, but this will slow down the simulation.

#### B. XInt-GA

XInt-GA is a thermal-aware genetic algorithm. It uses genes from a parent population and combines them in order to form a descendant population. A certain percentage is used to determine when a mutation (a change) occurs in the resulting offspring. This process occurs for several iterations, with a fitness function determining the likelihood of an offspring reproducing [6].

In ADTM, XInt-GA uses a three-descendant competition with the fitness function defining the coolest average node at time  $t+1$  as the most fit node. The winning descendant has two offspring plus itself to compete.

### IV. NUMERICAL DESCRIPTION OF ADTM AND EXAMPLE

A more detailed example will help show exactly how Von Neumann neighborhoods are used in ADTM. It takes into account three factors: heat generated from executing jobs, heat recirculated from nearby servers, and heat dissipation. These factors are further explained below.

- Constant job generated heat model: ADTM assumes jobs will consume the same amount of CPU power and will generate the same amount of heat on a server.  $\Delta t_{job}$  is used to denote the temperature increase of a server due to executing a portion of one job in a single time step. Thus, the temperature of server  $s$  will increase by  $\Delta t_{job} * m$  during one time tick, where  $m$  is the number of jobs executing on  $s$  at this tick. The lifetime of jobs in ADTM are random.
- Von Neumann heat recirculation model: The interference of nearby servers to server  $s$  due to heat recirculation is modeled by using a weighted Von Neumann sum method. Nearby servers can affect  $s$ 's temperature to different extents according to their distance from  $s$  and their own

temperature. Equation 1 describes the weighted sum of nearby servers' temperatures as  $t_{recir}$ , which stands for the temperature increase of  $s$  due to heat recirculation from other servers. In this equation,  $w_{s_i}$  decrements linearly as  $s_i$ 's distance to  $s$  increases, and  $\alpha$  is a constant representing how strong the recirculation is. In order to increase the speed at which the simulation runs, only servers within 2 distance units to  $s$  will have a notable effect on  $s$ . Further servers are excluded from the Von Neumann sum.

$$t_{recir} = \alpha * \frac{\sum_{s_i \in \{nearbyServers\}} w_{s_i} * s_i.temp}{\sum_{s_i \in \{nearbyServers\}} w_{s_i}} \quad (1)$$

- Heat dissipation model: Server  $s$ 's heat dissipates at every time tick due to the air conditioning system that constantly blows cold air. The heat dissipation occurs by decrementing each server's temperature by  $t_{dec}$  at every time tick until this server reaches the equilibrium temperature of the datacenter.

The overall temperature change for  $s$  at this time tick is  $t_{totalChange} = \Delta t_{job} * m + t_{recir} - t_{dec}$ . The temperature of  $s$  at the next time tick  $t_{i+1} = t_i + t_{totalChange}$ . The high temp threshold for  $t_{i+1}$  is set to  $65^\circ\text{C}$  [7]. When  $t_{i+1} > t_{threshold}$ , the server is shut down in order to cool down. A server's temperature cannot go below the equilibrium temperature  $t_{equ}$  since the cooling system only works when the datacenter is hotter than  $t_{equ}$ . ADTM sets  $t_{equ}$  to  $25^\circ\text{C}$  [7].

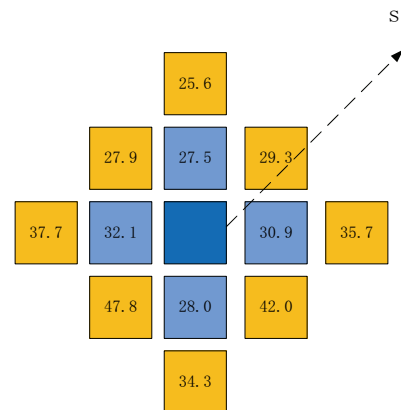


Fig. 1. A numerical example

For example, Figure 1 shows a 2D model of the datacenter, and the center square represents  $s$ , the node of interest. Blue squares represent a distance of 1 server away from server  $s$ , and their weights in the Von Neumann sum are set to 3. Orange squares represent a distance of 2 servers away from  $s$ , and their weights are set to 1 in this example.  $\alpha = 0.1$  is used for equation 1. Solving for  $t_{recir}$  yields  $3.18^\circ\text{C}$ . Suppose  $s$  is executing 10 jobs at the current time tick, and  $t_{job} = 0.1$ . This puts temperature increase due to jobs at 1.  $t_{dec}$  is set to 3 in this example, so the final temperature change can be calculated as  $t_{totalChange} = 1 + 3.18 - 3 = 1.18^\circ\text{C}$ .

## V. JUXTAPOSING THERMAL-AWARE ALGORITHMS THROUGH THERMAL SIMULATIONS

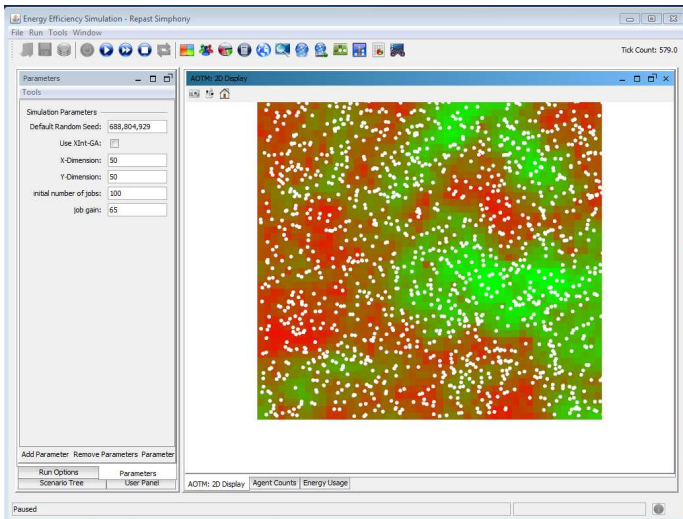


Fig. 2. ADTM Simulation

Figure 2 represents a model generated by ADTM (as well as the configured parameters) that can be used to help cool data centers. Each white circle represents a job. Currently they are either randomly placed (if the XInt-GA parameter is unchecked) or placed according to the XInt-GA algorithm. Each square in the simulation represents a server and its temperature. The lightest green color represents a server that is at its equilibrium temperature (the temperature of a server without interference from job processing, HVAC, or neighboring interference). As a green server turns to red, that server is gaining more heat from processing jobs or from the heat dissipation of nearby servers in its von Neumann neighborhood. At the hottest temperature (completely red), the HVAC unit will be forced to cool the server (expending cooling energy). As smarter algorithms are programmed into ADTM, servers and the HVAC will take on different behaviors. Server temperatures will be cooler on average as jobs are placed in a more efficient manner. The simulation can be run automatically, or time steps can be simulated one at a time. In this way, different thermal-aware algorithms can be modeled with ADTM.

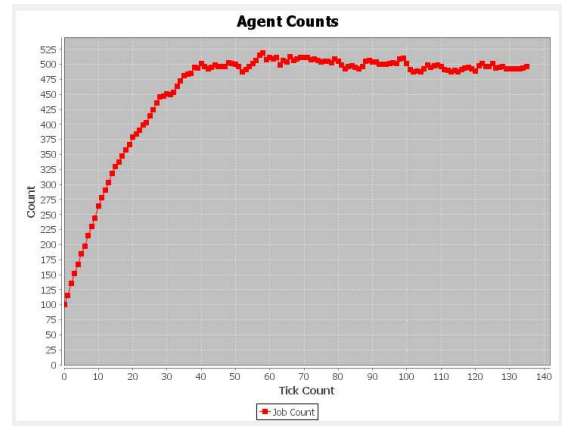


Fig. 3. Agent Counts

In addition to modeling temperature, more fine-grained detail can be seen by adding charts. Figure 3 shows an example chart that can be generated by ADTM of the number of jobs associated with each time step.

## VI. THERMAL MODELING XINT-GA

In addition to random job placement, ADTM implements a version of XInt implemented by a scaled down genetic algorithm (GA). With XInt, ADTM can handle spikes in job input to the simulator. Better job distribution leads to better thermal management in the datacenter. A thermally managed datacenter in turn leads to lower cooling costs.

### A. Implementation

XInt is a process that can be followed using multiple methods. Two methods discussed by its authors are SQP (sequential quadratics) and GA (genetic algorithms). ADTM implements the genetic algorithm version of XInt. The XInt process was simplified in order to make it faster; a full genetic algorithm with simulation lookahead would have taken the simulation too long to run and would have been overkill. Instead, the XInt lookahead was only set to one timestep in the future to make sure jobs weren't clustered into a single section during each time step. Next, the genetic algorithm was simplified. Using a fitness function of the chosen node's future temperature, the parent population chose random, relatively cool spots. The most fit parents were chosen to populate the second generation, which mutated the parents to find the coolest nearby node to the most fit parent. The most fit node in the child generation was chosen as the job placement location.

### B. Observations

The default simulation parameters are intended to load the datacenter in a way that produces enough heat to turn on the HVAC. While the load isn't more than the datacenter can take, it is enough to require a cooling source. The first observation can be drawn from observation of the simulation's graphical datacenter representation interface.

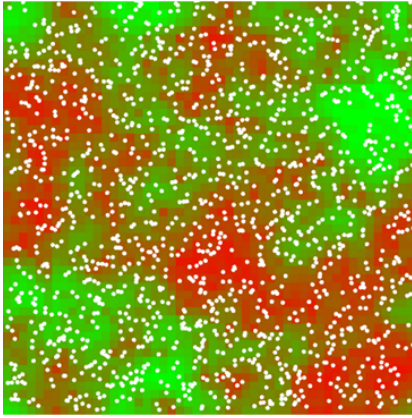


Fig. 4. Random Job Placement at 900 Seconds

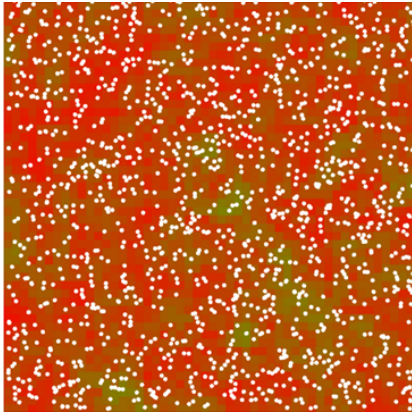


Fig. 5. XInt-GA Job Placement at 900 Seconds

At first glance of figure 4 and figure 5, the inclination is to think that XInt-GA's algorithm causes a hotter datacenter, but this assumption based on the larger area of red would be wrong. In random job placement, the hotspots require the HVAC to turn on, causing a larger difference between the hot and cold spots of the datacenter. An unevenly heated datacenter wastes precious HVAC resources on cooler areas of the datacenter! In contrast, XInt balances out the heat through the entire datacenter. There are fewer hot and cold areas, as the datacenter is more uniform.

A more uniformly managed datacenter yields to the next important observation from the automatically generated data of ADTM.

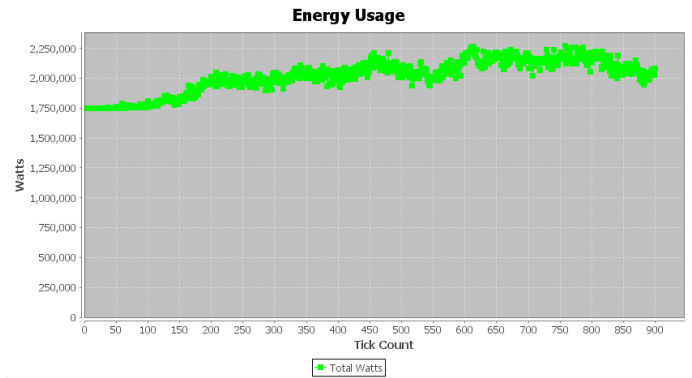


Fig. 6. Random Job Placement Energy Usage

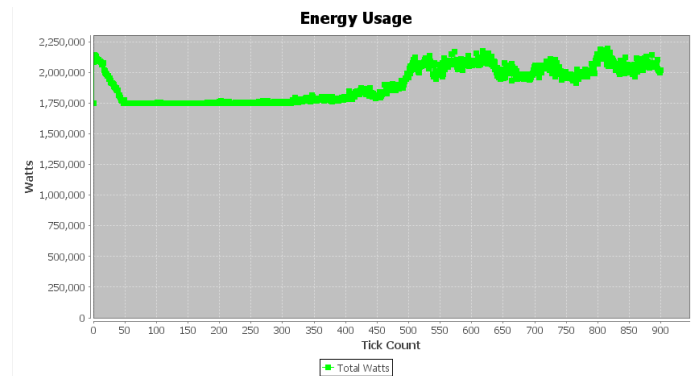


Fig. 7. XInt-GA Job Placement Energy Usage

As shown in figure 6 and figure 7, with random job placement, the HVAC energy usage kicks in after only 150 seconds of simulated job load. Contrast this with XInt, which balances the load and temperature during peak usage, causing it to take longer for the datacenter to heat up (about 400 seconds). This indicates that XInt is better able to balance out high, temporary loads on a datacenter so that the HVAC does not have to be used as often. In addition, the wattage used by XInt never spiked as high as random job placement.

## VII. A LOOK BEHIND THE SCENES OF ADTM: USING REPAST SIMPHONY AS A FRAMEWORK FOR THERMAL MODELING

These models and charts are generated by utilizing the Repast Simphony framework, which provides utilities for time-step-based agent-directed software development. The Repast framework provides tools for visualization and includes a plugin for development in the Eclipse IDE [8]. Within this framework, a simulation of an  $n \times m$  datacenter can be created where the temperature of the nodes in the datacenter are influenced by their neighbors. Instead of simulating the movement of heat through the system, each node's temperature is calculated by a function using its current temperature and the neighboring nodes.

### A. Agent-Based Simulation Development

ADTM's simulation is an imperative-driven agent-directed development. An imperative-driven simulation is one in which the agents (or actors) in the simulation are given rules by which to achieve their goals. In this case, we define the agents and their rules. For more information, please reference [9]. Here we define the agents and their rules:

1) *The World*: The world in which the individual agents reside defines their boundaries. The world uses three important parameters: an X-dimension, a Y-dimension, and a job creation rate. These dimensions define the cluster of nodes that reside next to each other. By default, the world is a 50x50 room of 2500 nodes equally spaced in the cluster. Each time unit, the job creation rate defines how many jobs are created. Therefore, the rules for the world are:

- 1) If the world is not yet created, create an  $n \times m$  world from the X-dimension and Y-dimension parameters, placing a Node agent on each gridpoint.
- 2) Each step of the simulation, create  $x$  jobs defined by the job creation rate.
- 3) Place the jobs using the selected job placement algorithm.

2) *Nodes*: The nodes process jobs, heat up as they process jobs, and cool down when they get too hot by the assumed HVAC unit. The nodes have a homeostasis temperature of 25°C to 65°C. Their behavior is defined as follows:

- 1) Calculate the Von Neumann neighborhood weighted average of temperatures in the surrounding Nodes (Von Neumann Neighborhoods are described in section IV). When the neighbor distance is 1, the weight is 3x that of a neighbor of distance 2.
- 2) Calculate the average of the Node's current temperature with the weighted Von Neumann Neighborhood calculation from (1).
- 3) Change the Node's temperature to the temperature influenced by the Von Neumann Neighborhood from (2).
- 4) Decrement the temperature by 1°C if it is not currently at its homeostasis temperature.
- 5) If the Node's temperature is too high, log that the HVAC implicitly cools the Node by the number of degrees required to get back to a safe temperature range.

3) *Jobs*: Jobs are the workunits of the simulation. They cause the heat intrinsically and actively change the temperature on the node in which they are residing. While the Nodes' instructions have them dissipate heat in a Von Neumann Neighborhood, the Jobs increase heat on the node that is processing them. Operating under the assumption that a node can process as many jobs simultaneously as needed, each job randomly generates its lifespan. A Job's rules are defined as follows:

- 1) If the job has just been created, randomly generate its lifespan. The lifespan is currently defined as being between 0 and 50 seconds.
- 2) Increase the Node temperature by 3°C.
- 3) Decrement the lifespan by 1 second.

- 4) If the lifespan drops to 0, remove the job from the simulation.

### B. Visualization

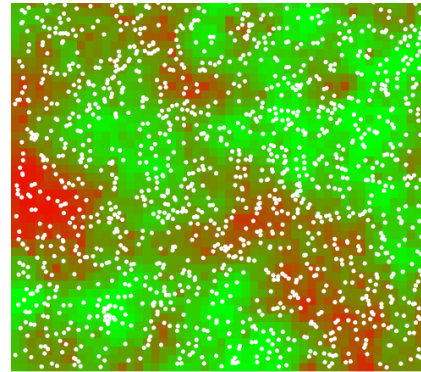


Fig. 8. Simulation Visualization

The world is drawn onto the display as defined by the X and Y dimensions. Nodes are placed onto the grid with their homeostasis temperature of 25°C. Nodes are visualized as an individual square on the grid. When a Node is at its homeostasis temperature, it is bright green. As it warms up, it changes to bright red. Individual Jobs are shown as small white dots close to the Node that is processing them.

### VIII. ISSUES FACED WHILE IMPLEMENTING ADTM

To date, no known issues have been logged against ADTM; however, currently only one major thermal-aware algorithm (XInt-GA) has been implemented.

Six main implementation issues arose as ADTM was designed:

- 1) The Java classpath was difficult to setup for the Eclipse project. As other team members attempted to import the project into Eclipse, path issues arose and had to be resolved before ADTM could be compiled on another machine.
- 2) Several Java dependencies are required that are provided by the JRE built into Repast. These are included in the binary distribution. Although 64-bit Java is operational, it was difficult to get working.
- 3) Agent-directed software development is not very common in our student body, so fewer developers have expertise in it. This means issues that arise from agent-directed development will not have as many searchable answers on the Internet as more traditional object-oriented or function decomposition development. In addition, a learning curve existed for team members not familiar with this approach.
- 4) The parameters that were made available to the user were chosen by how useful they would be to model a data center. As such, size of the data center, initial jobs, and created jobs were made configurable.
- 5) Temperature data can be observed by the color of the node in the model which is not currently configurable. This could pose a problem for color-blind users.

## IX. USING ADTM

### A. Building ADTM

The easiest way to use ADTM is to execute `start_model.bat` included in the ADTM directory of the binary distribution. While this method of execution does not require downloading Repast Symphony, the launcher only supports Windows platforms XP and later. Other operation systems will have to build the project from the source code.

In order to build the project, the Repast Symphony suite can be downloaded from [8]. The “Download” section of this site has install instructions for Windows, Mac, and Linux. Once the download is complete, the ADTM project can be imported into Eclipse from the ADTM directory. One can do this by doing the following within Eclipse Repast:

- Go to File
- Import...
- Existing Projects into Workspace
- Under “Select root directory”, browse for the ADTM project
- Select Finish

Once the project is imported, it can be built by right-clicking the project under Package Explorer and selecting “Build Project”.

### B. Running the Simulation

To run the project, select the arrow next to the Play button and choose “ADTM Model”. The following screen should appear (Figure 9).

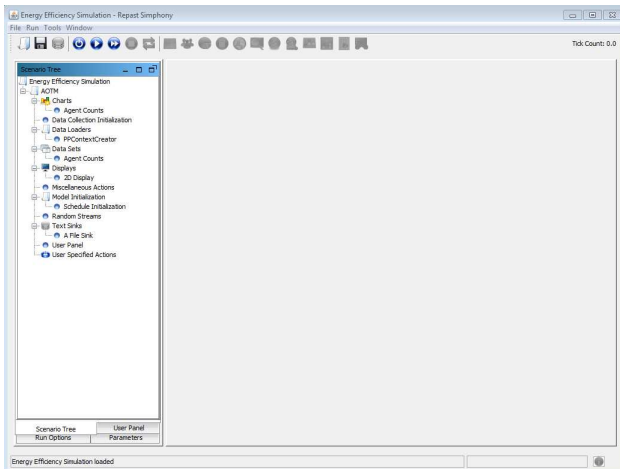


Fig. 9. Starting ADTM

Selecting the play button (Start Run) will start the simulation and run through several time steps automatically until the stop button (Stop Run) is pressed. This will stop the simulation on a time step. In addition, the button between the start and stop button will run one step in the simulation. This can be used run the simulation step-by-step. In order to start the simulation over, the Reset Run button can be selected (just to the right of the stop button). Note that this will change the Default Random Seed and reset the parameters to default

values, which will cause a different random job placement strategy to occur.

The model is shown by default. In order to view a chart, the Scenario Tree needs to be selected in the left panel. Right-click the charts option, and select “Add Time Series Chart”. After the name and data set is input, Finish can be selected. The model view will now have a tab to select that will have the same name as the name of chart that was created. Selecting that tab will display the chart.

### C. Inputting Parameters

By default, ADTM has a set of parameters that will be used in the simulation. The input parameters are as follows (defaults in parenthesis):

- Default Random Seed (random): seeds the Java pseudo-random number generator
- Use XInt-GA (unchecked): selects either the random job placement algorithm or the XInt-GA algorithm
- X-Dimension (50): number of servers on the X-axis
- Y-Dimension (50): number of servers on the Y-axis
- Initial Number of Jobs (100): the number of jobs that will be used at the beginning of the simulation - note that the first timestep will have (initial + job gain) number of jobs
- Job Gain (65): the number of jobs added each timestep

In order to customize these parameters before a simulation, choose the Parameters tab on the left panel. Note that the simulation must be restarted in order to use any modified parameters.

## X. EXPERIMENTAL RESULTS

Having explained the usage of ADTM, this thermal modeler will now be evaluated. The performance of the proposed agent-directed thermal modeler will be given. In addition, a comparison of the modified XInt algorithm and the random job placement strategy will be discussed.

### A. Purpose of ADTM Experiment

Extensive experiments were conducted to test the effectiveness of ADTM. The purpose of these experiments were threefold. First, the performance of the modified XInt thermal-aware algorithm was compared with the thermal-agnostic random job placement strategy. Potential energy savings and simulation run times were inspected. Second, the temperature changes in the data center at different time steps were visualized using different job placement strategies. Third, the simple user interface was demonstrated in order to show how changing parameters affects a specific thermal-aware algorithm.

### B. Experimental Setup

The proposed thermal modeler was implemented in Java on the Repast Symphony Agent-Directed programming framework. The modified XInt genetic algorithm and random job placement were implemented in Java. All the experiments were conducted on a Thinkpad Windows laptop equipped with an Intel Xeon 2.4GHz processor and 4GB memory. ADTM

with default parameters will have a small memory and CPU footprint. It will run on any system that can install a Java runtime environment with the Repast Symphony framework dependencies. Since the parameters for the size of the data-center and number of jobs created are dynamically configured by the user, the hardware utilization footprint will be different for each set of parameters. Using default parameters with the the Java(TM) SE Runtime Environment 1.6.0 build 31 in 32-bit mode, ADTM requires 108MB of memory (73MB of which are used for the VM overhead) with an additional 1MB of memory required for every 30 simulated seconds. Unless otherwise specified, the default 50\*50 world was used, and the duration of experiments were set to 800 seconds.

### C. Energy Usage

The energy usage for the two job placement strategies included with ADTM are now compared using different data center utilizations.

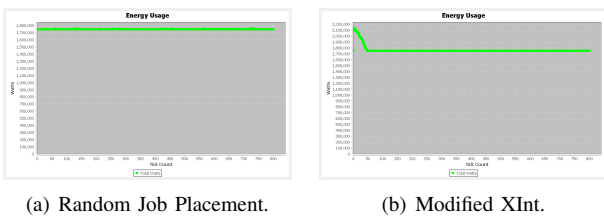


Fig. 10. Underutilization.

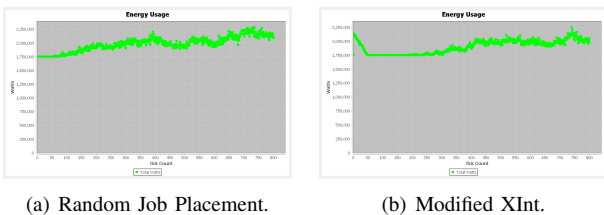


Fig. 11. Overutilization.

Figure 10 shows that when the data center is underutilized, the baseline power consumption for the current 50\*50 data center is 1700 to 1800 KW for both strategies. The baseline power consumption is due to the fact that a server still consumes power when it is idle or in low utilization, which can be as high as 2/3 of its peak power. Also note that in this case, the XInt algorithm barely saves any energy costs over random job placement. The reason is that the HVAC adds cooling costs when a server reaches the max temperature. In the underutilization case, hardly any server reaches  $T_{max}$ , so only the baseline cooling cost is incurred for both strategies. However, the overutilization simulation in Figure 11, shows that both strategies have higher power usage than the baseline. The modified XInt algorithm consumes less power and is more stable. This shows that the modified XInt algorithm provides more energy efficiency than the random job placement strategy in a datacenter that is overutilized.

### D. Simulation Run Time

The simulation run time of the two job placement strategies will now be examined to show that the modified XInt algorithm runs almost as quickly as the random job placement strategy. For a 800 time tick simulation on a 50\*50 world, the random job placement simulation runs for 62 seconds, while the modified XInt algorithm runs for 63 seconds. For a 800 time tick simulation on a 40\*40 world, the random job placement simulation takes 51 seconds, while the modified XInt algorithm runs for 52 seconds. There is barely a noticeable difference in the simulation time between the two job placement strategies, which indicates that ADTM runs quickly even when simulating complicated thermal-aware job placement algorithms.

### E. Analysis and Implications

When the job rate is too high for the number of servers, no thermal-aware algorithm will be able to keep the data center cool enough without excessive cooling from the HVAC. Nevertheless, the modified XInt algorithm uses about 5% less energy than random job placement when the datacenter is overutilized. Extensive experiments confirm the correctness of our agent-directed thermal modeler by successfully showing energy savings for a better thermal-aware algorithm. These experiments also show that getting data from ADTM is quick (several algorithms could potentially be run in just an hour). Finally, the learning curve for ADTM is very low thanks to its well-defined parameters, intuitive user interface, and easy-to-understand graphs and models.

## XI. CONCLUSION

Several thermal-aware algorithms exist that can help data centers minimize the energy needed to cool a data center. Simply reading about different algorithms, however, makes it hard to visualize the impact this strategy will have among the nodes in a data center. Thermal modelers do a great job of visualizing which nodes in a data center will remain hot and which ones will be cooled by simulating an algorithm and the air flowing throughout a data center. However, existing modelers can have a high learning curve and take a long time to simulate in order to produce a model.

ADTM's purpose was to minimize the learning curve while still providing a useful model that does not take several hours to create. The few parameters needed to run an ADTM simulation minimizes the time needed to learn how to use ADTM. It also runs quickly, simulating steps in a matter of milliseconds. This means in just a minute or two, one can have a very meaningful model that can show how a thermal-aware algorithm will affect a data center.

ADTM was then shown to compare two different job placement algorithms. In the first algorithm, random job placement puts jobs randomly on a node in the data center. ADTM's default values are intentionally set to overload the data center. When the data center is overloaded, random job placement still has several nodes that are very cool. This means the HVAC will needlessly turn on to cool down the hot nodes

while several nodes are still at their equilibrium temperature. XInt-GA is more efficient (5% more efficient). When the HVAC turns on when the XInt-GA parameter checkbox is checked, most of the nodes are already heated close to their threshold temperature value of 65°C. The energy charts that can be produced by ADTM shows that it takes longer for the HVAC to turn on using the XInt-GA algorithm. This simple example illustrates the use of ADTM in showing the superiority of XInt-GA over random job placement as well as showing servers' temperatures with each time step.

## XII. FUTURE WORK

This report shows that ADTM can be an effective way of modeling thermal-aware algorithms with a very small learning curve. Future ideas for ADTM may include providing a more accurate thermal model. A heat recirculation matrix could potentially provide more accuracy. In addition, more thermal-aware algorithmic implementations could be included. Some ideas might be algorithms that consider file placement, algorithms that include thermal prediction, and algorithms that minimize heat recirculation. A final idea could be to use real trace data from the industry (perhaps data from Yahoo! or Google, for instance) in order to verify the effectiveness of ADTM.

The von Neumann neighborhood represents a simplistic interference matrix when influencing a neighboring node's change in temperature. CFD models can be used to more accurately model this region of influence. This would also delinearize the HVAC influence in cooling down the nodes. Equations based on CFD and experimentation would remove some of these static assumptions.

## XIII. ACKNOWLEDGMENTS

- Jon Hood
  - Project Manager
  - Programmed ADTM
  - Wrote documentation on detailed look at ADTM
- Tim Scott
  - Divided out report write-ups and wrote sections not covered by Jon and Jiao
  - Wrote README document
- Jiao Yu
  - Wrote documentation for numerical examples and experimental results
  - Researched real-world data to be used in ADTM
- Dr. Xiao Qin
  - Provided assignment to design a thermal modeler
  - Directed students from start to finish throughout the project
  - Provided motivation as ADTM was developed
- Dr. Levent Yilmaz
  - Provided guidance in ADS development
  - Recommended changes for ADS conferences
- Dr. Drew Hamilton

- Provided proofreading and expertise in publishing documents
- Prepared students for presentation at conferences

## REFERENCES

- [1] R. T. Kaushik and K. Nahrstedt, "T\*: a data-centric cooling energy costs reduction approach for big data analytics cloud," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012, pp. 52:1–52:11. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2388996.2389067>
- [2] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers." *Computer Networks*, vol. 53, no. 17, pp. 2888 – 2904, 2009. [Online]. Available: <http://search.ebscohost.com.spot.lib.auburn.edu/login.aspx?direct=true&db=buh&AN=45422060&site=ehost-live>
- [3] K. Joonho, C. Sung Woo, and K. Skadron, "Recent thermal management techniques for microprocessors." *ACM Computing Surveys*, vol. 44, no. 3, pp. 13 – 13:42, 2012. [Online]. Available: <http://search.ebscohost.com.spot.lib.auburn.edu/login.aspx?direct=true&db=buh&AN=77418767&site=ehost-live>
- [4] R. Brown *et al.*, "Report to Congress on server and data center energy efficiency: Public law 109-431," 2008.
- [5] J. Kennedy and R. Mendes, "Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms," in *Soft Computing in Industrial Applications, 2003. SMCia/03. Proceedings of the 2003 IEEE International Workshop on*. IEEE, 2003, pp. 45–50.
- [6] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach." *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 11, pp. 1458–1472, 2008.
- [7] D. Moss. (2009) Data center operating temperature: What does Dell recommend? [Online]. Available: <http://www.dell.com/Learn/us/en/19/business~solutions~whitepapers~en/Documents~dci-Data-Center-Operating-Temperature-Dell-Recommendation.pdf?c=us&l=en&s=dhs>
- [8] R. D. Team. (2012) Repast suite. [Online]. Available: [http://repast.sourceforge.net/repast\\_simphony.html](http://repast.sourceforge.net/repast_simphony.html)
- [9] J. Barjis, I. Rychkova, and L. Yilmaz, "Modeling and simulation driven software development," in *Proceedings of the 2011 Emerging M&S Applications in Industry and Academia Symposium*. Society for Computer Simulation International, 2011, pp. 4–10.