



Improving energy efficiency of supercomputer systems through software-aided liquid cooling management

Radosław Januszewski^{*}, Rafał Różycki[†], Grzegorz Waligóra[‡]

Abstract. Many fields of modern science rely more and more on the immense computing power of supercomputers. Modern, multi-thousand node systems can consume megawatts of electrical energy in highly uneven manner, challenging the data center infrastructure, both power and cooling coils. The traditional way of managing the infrastructure makes each subsystem of a data center (e.g. cooling) independent from all other in the way it relies only on local sensors to manage the infrastructure. The erratic nature of computing in a large data center makes this approach suboptimal. In the paper we show that by challenging the traditional split between the infrastructure and the computing equipment, one can gain significant boost in energy efficiency of the entire ecosystem. A solution that predicts cooling power demand basing on the information from a supercomputer resource manager, and then sets up the parameters of the cooling loop, is presented along with potential benefits in terms of reduction of the power draw.

Keywords: supercomputing, resource management, liquid cooling.

1. Introduction

Over the last decade one could have seen that while the power budget of a standard 1U server has remained relatively stable, devices have started to be packed more densely, resulting in more kW's of heat per square meter to be handled by air conditioning units. Even modern projects, designed with power efficiency in mind, like, e.g., Open Compute Project [9], still stick to the air conditioning principle. The growing power density and resulting complexity of handling the heat removal is addressed by optimizing the air flow using different technologies of separation between hot and cold areas in a data center, as well as increasing the air temperature in order to enable the cooling loop a more efficient operation mode. All these approaches can reduce power overheads on cooling from 40%

^{*} Poznań Supercomputing and Networking Center (PSNC), Poznań, Poland,

[†] Institute of Computing Science, Poznań University of Technology, Poznań, Poland, rafal.rozycki@cs.put.poznan.pl

[‡] Institute of Computing Science, Poznań University of Technology, Poznań, Poland, grzegorz.waligora@cs.put.poznan.pl

[11] up to 25-30%, depending on the solution type and local climate. Those big overheads are a direct result of using air as the heat transfer medium. While in some applications, like offices or food preservation, using air is a necessity, when dealing with servers there are also other options possible. One can use different kinds of liquids for this purpose. Liquids have generally better heat conductivity and specific heat which means that they are more effective. Some studies conducted both in USA [1] and in Europe [5] clearly suggest that this kind of solution should be implemented to improve the efficiency of a data center.

The potential for replacing air with liquids has been investigated for many years [14] using both water and other coolants [7], and in the recent years is being accepted as a factor that should be taken into consideration when designing a data center [4]. Although direct liquid cooling is much more efficient than air as the medium used for removing the waste heat, there are still some issues to be solved, related to both efficiency [6] and management. Algorithms used to manage the cooling loops as well as the way these loops are built and managed are not suited to cope with supercomputer characteristics, especially rapid power jumps that can occur within seconds. Typically, cooling loop management software relies on temperature and pressure sensors that measure the coolant parameters. This means that the cooling loop and the heat source are two separate entities. While this kind of solution serves well in other applications, it is unfortunately not suited for being paired with computers. In the paper we will present some of the problems encountered while working with our system, and show how coupling the cooling management loop with the cluster's resource management system can not only save energy by selecting optimal environmental conditions for a predicted workload, but also allows the cooling loop hardware to work in more steady conditions preventing premature wear of the devices.

On the cluster side, providing a stable environment is important because it affects the economic aspects of the cluster operation. Obviously, cooling the cluster introduces costs. Increasing the temperature of the cooling medium reduces this overhead, however, it leads to warmer working environment for the computers. This, in turn, means that the cluster will consume more energy (fans working faster and leak currents in the chips), wear faster [12], or degrade its performance to keep within the thermal design confines of the CPUs. The last problem is especially important in High-Performance Computing (HPC), as a large application that spans tens or hundreds of servers usually has some sort of synchronization, so degradation of performance resulting from overheating even on a single node can have an impact on the whole computation.

2. Description of the test bed

At present, direct water cooling seems to gain popularity, but at the moment when the prototype system was built, there were only few solutions on the market of x86 servers available. IBM iDataPlex [3] was the only one delivered by one of the big vendors but, because in this product only CPUs and memory were cooled by water, this system was not able to fulfill the primary requirement: all components of the cluster should be directly liquid cooled. Therefore, Poznan Supercomputing and Networking Center (PSNC) [10] decided to choose Iceotope as the system vendor. The Iceotope standard solution was modified for the purpose of the project.

The computing cluster consists of 40 dual 6-core Xeon E5 servers working at 2.0 GHz (2.3 GHz when turbo mode is enabled). In addition, in the cluster we have installed 12 AMD S9000 GPGPU cards installed in 6 modules connected to selected servers with external PCI connections. Each server is equipped with 32 or, in case of GPU serving ones, 64GB of memory, 2x1Gbit Ethernet, QDR InfiniBand interface and local 60 GB SSD drive.

On all nodes Ubuntu 12.04 is used along with SLURM [13] resource manager. The system configuration, from IT configuration perspective, was a typical HPC machine at the moment when it was installed, processor generation, interconnect and memory profiles were similar to what was used in Europe for building HPC clusters. The only aspect that was different was the cooling solution - using immersion in dielectric liquid was at the moment something unique and was a crucial feature from the experiment perspective. The original work plan assumed that the machine and the separate cooling loop were built to check if direct liquid cooling enables a chiller free operation of a HPC cluster, and what are the consequences for the energy efficiency of the solution. Having a separate loop was a key for the second test scenario, namely checking what kind of optimizations may be achieved by pairing the data center infrastructure management with the queue system of the HPC cluster. Because data center the experiment was performed in, provides production services for scientists in Poland, it was not possible to use existing, shared infrastructure for conducting the experiments as it might have impacted other services. The choice of IT components of the cluster was a deliberate one - there were plenty of similar clusters available in different HPC centers in Europe that could be used as a reference for the first scenario.

In addition to the cluster, PSNC has ordered a customized, highly manageable and monitored cooling loop that was designed to provide maximum efficiency while being as simple as possible.

The entire cooling path consisting of three loops is presented in Figure 1.

The first one is an internal server cooling. Each server in the cluster is built as a sealed module filled with nonconductive liquid Novec [8] which acts as a heat conductor that transfers the heat from the chips to the water-Novec heat exchanger installed in the side of the module. All electronic parts (motherboard, CPUs, memory, SSD drives) are put inside the module, and are completely submerged in the Novec. The server modules are installed vertically, so the convection is ensuring proper Novec flow along the heat exchanger and the electronics. Coolant used in this particular installation has a boiling point of 90C which can be reached by heavily loaded processors. The boiling can theoretically be beneficial from the cooling point of view, as phase change can increase the efficiency of heat removal from the CPU. In our installation however this phenomenon must be avoided, boiling of the Novec would increase the pressure inside the module which can cause leaks and possibly damage the module package. From the CPU point of view temperatures in the range of 90C are close to overheating, thus the Turbo Core will be disabled removing the potential performance increase.

The second step, later referenced as the primary loop, is the cooling loop inside the cluster rack that is transferring the heat from the server modules to the heat exchangers installed in the bottom of the server rack. In this loop we use demineralized water as the coolant. The water was supplemented with anti-corrosion and bio-inhibitors. In the rack there are two semi-independent pumps working in parallel. This loop is working on negative pressure ensuring that eventual leaks result in the air being sucked in the loop rather than spilling the water out of the loop.

The third step, a secondary loop, is used to transfer the heat from the server rack inside data center room to a dry cooler unit placed on the roof of the building. The dry cooler is equipped with an adiabatic cooling support that should help keep the coolant within desired

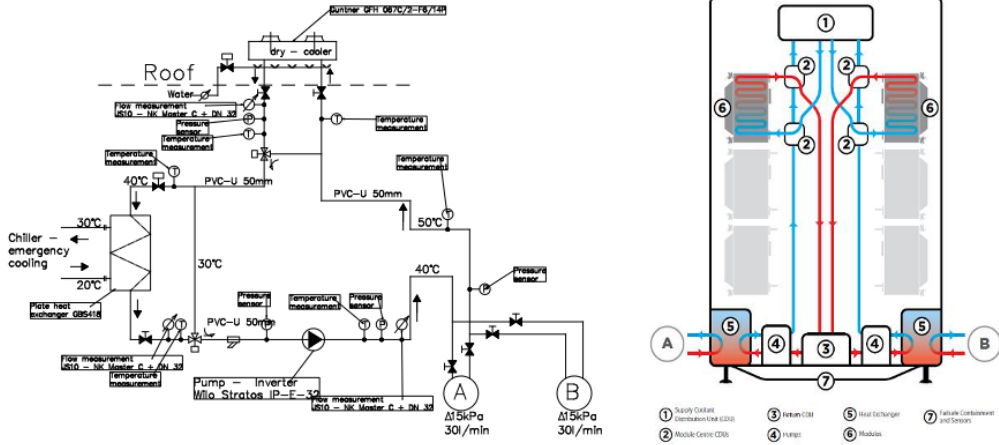


Figure 1. Testbed cooling loop, data center part (left) and cluster internal loop (right)

temperature range even during hot days. Because this loop is designed to operate also in winter conditions, it is using a 30% glycol-water mixture as the coolant. All temperature sensors in this loop are installed in-flow therefore ensuring exact measurements.

The secondary cooling loop control may work in one of three automatic modes. The first one automatically adjusts the setting of the pump, valves and the dry cooler in order to minimize the power consumed by the cooling loop by trying to operate in the high range of the servers specifications (45C inlet and 50C outlet temperatures). In the second mode the controller automatically adjusts the setting of the pump and valves to keep the inlet and outlet temperatures within previously set temperature range. In this mode whenever the temperature of the coolant going to the servers exceeds the set points, the control system tries to react first by opening the three way valve, and thus allowing more coolant to pass through the rooftop heat exchanger. Once the valve is fully opened, the fans on the heat exchanger start to rotate faster. If this does not help then the speed of the main pump is increased gradually until the desired parameters of the coolant are reached. The order of reactions was selected because spinning up the fans on the rooftop heat exchanger is less costly in terms of energy than increasing the speed of the pumps – the maximum power draw of the fans is ~200W while the pump at maximum setting is using (according to the technical data) maximum of 2000W. Obviously this mode cannot be used without additional logic in background that tracks the external temperature and sets reasonable set points for the coolant. One cannot expect to achieve the coolant temperature significantly lower than the ambient air temperature.

Alternatively, there is a possibility of manual control over each of the elements (pumps, valves etc.) to allow using more sophisticated control algorithms. The first mode, that is representing the standard mode of managing cooling loops in the industrial solutions, has

proved to be impractical. The managing algorithm relies only on the reads from the sensors in the cooling loop, without the knowledge on the state of the machine.

As a result, if the servers were idle for a prolonged period, the coolant flow in the secondary loop dropped to a level that was not acceptable, resulting in overheats. Therefore, the second mode was selected as a starting point for the tests on the cluster. The set points in the secondary loop were adjusted every 30 seconds. The inlet temperature value was selected basing on the external air temperature, and the delta was calculated on the basis of power consumed by the cluster.

Servers, when considered as a source of heat, are characterized by significant jumps in cooling power requirements, an increase of 150-200% in a matter of seconds can happen during a normal operation of the cluster. Because the management of the loop relies on the reads from physical sensors placed in the loop, the delay between the change of the water temperature and the measurement caused issues described later in the paper.

Although the cooling loop was designed to be able to handle the servers covering all external temperatures in Poznan, we decided to have a backup connection to the facility chilled water loop. Except from testing this loop was never used during presented tests or normal operations.

3. Testing procedure

During a normal operation of a supercomputer the number of applications that are working at a given moment can change rapidly. Also the energy profiles of applications may vary significantly, often depending on the data set. In order to provide a stable and controlled environment, we have decided to run benchmark applications with stable and known energy profiles.

To simulate a load on the machine we stressed the servers by running a single instance of the High Performance Linpack (HPL) [2] on 34 nodes and a separate HPL instance on 6 nodes and 12 GPU cards. The tests were repeated to produce a constant load.

The entire test procedure consisted of 5 cycles, after each cycle the inlet temperature to the machine was increased by 5 degrees, starting from 15C inlet and ending with 35C inlet temperature. For each inlet temperature we checked different deltas of the inlet-outlet temperatures. An original plan assumed starting with 2K delta and ending with 10C with 2K step, influencing the flow in the secondary loop. It turned out that having a constant 2K delta was not possible with the fully loaded system, therefore the final tests were done with 3,4,6,8 and 10K deltas. The internal server loop was running at a constant speed. Because of a thermal inertia of the cluster, after each temperature change of the server inlet temperature, there was a 30 min period to allow the machine to stabilize the temperature. Both thermal and power sensors internally were read every 5 seconds, and then the values were averaged in 1 minute period and put in the database. The averaged values were used to produce the final results.

The tests are not covering the top end of the spectrum (inlet temperatures of 40C and more), the reason for that is explained in the next section.

Since the primary cooling loop was working at a constant pace, only the components of the secondary loop were controllable and allowed manipulations leading to energy optimizations. In this loop the most power hungry components were the pump and the fans

of the rooftop heat exchanger. Because the fan activity is mostly affected by the external conditions, it is relatively low power (maximum 200W of power consumption), and is in a great degree dependent on the climate conditions, we have focused on optimization of the pump activity.

The pump was running at minimum of 50% because with this setting it was producing the minimum flow required by the cluster design and, at the same time, minimum pressure required to transfer the coolant to the rooftop. Looking at the power consumption profile of the pump (Fig. 2), one can see that the power consumption difference between 50% and 100% is almost an order of magnitude, therefore it is worth keeping the pump activity as low as possible.

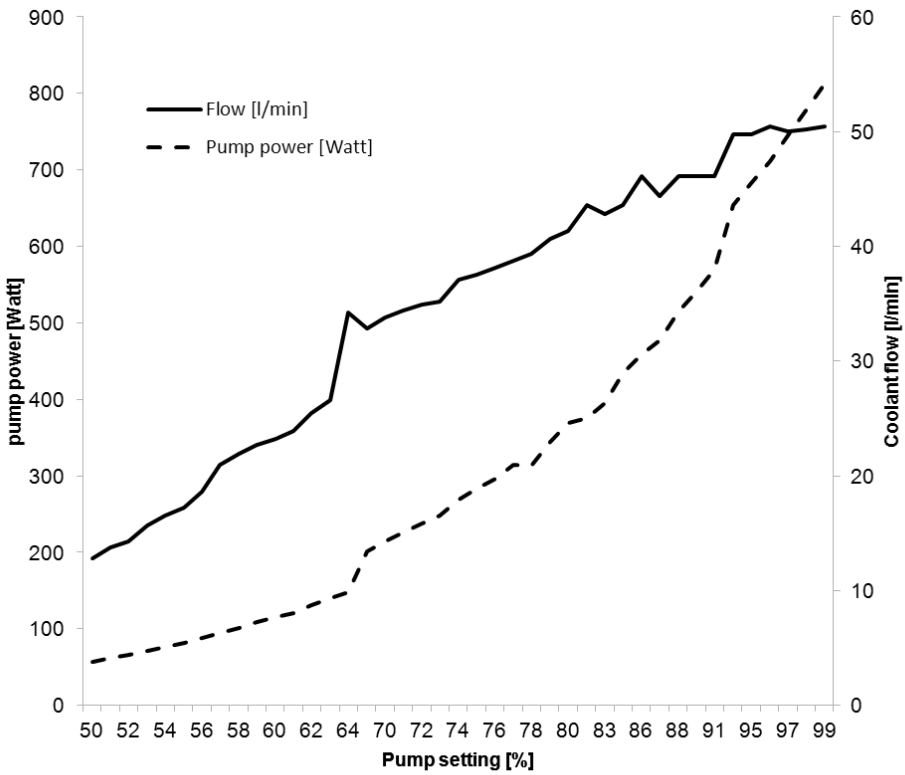


Figure 2. Power consumed by the cooling loop pump for different settings

Because the flow rate of the coolant via both rack and rooftop heat exchanger depends both on pump setting and the setting of the three way valve directing flow to the rooftop heat exchanger, the data for Figure 2 were collected when the valve was fully opened.

4. The results and optimizations

In this section we will present results only from two sets of tests: 20C and 35C. 20C represents temperature range that negates the risk crossing the dew-point in the data center room. Going for lower temperatures (e.g. 15C) might result in condensation on metal parts (e.g. water loop connectors) exposed directly to the data center ambient air.

Increasing temperature of the inlet coolant temperature produced a side effect in form of increased temperature of the servers and rack. Whenever the coolant inlet temperature is greater than the ambient air temperature, in case of PSNC the data center temperature was 25C, the system will dissipate a portion of the heat to the data center air. It is also true in case of our system, even though additional thermal insulation installed in the sides of the rack (see Fig. 3).

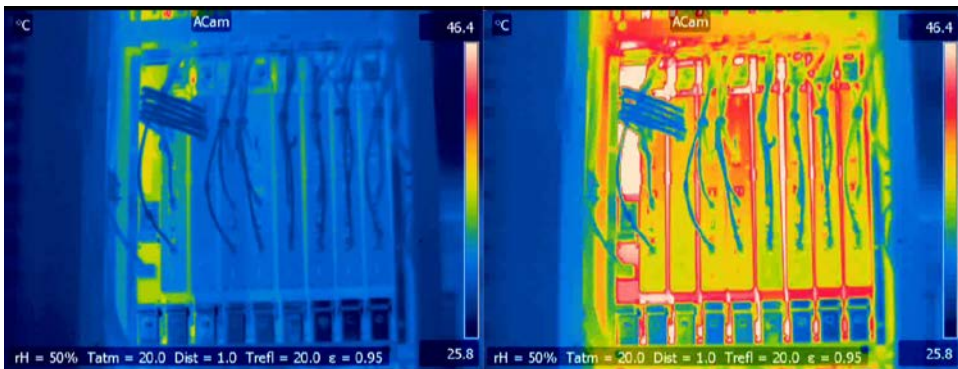


Figure 3. Thermal images of the cluster cooled with 20C and 40C coolant

Within this paper all cooling power references describe the amount of heat that the water cooling loop is able to remove from the cluster, and not the power consumed by the cooling loop. When the cooling power is equal to the power consumed by the cluster, it means that all heat produced by the cluster is removed using the water loop. If the cooling power is less than power consumed by the cluster, then the excess energy is dissipated to the data center room air and has to be removed by less effective, traditional air cooling. Therefore, we want to avoid this situation as much as possible.

One can see that starting with 30C inlet temperature the cooling loop performance delivers less cooling power than the server is generating heat, thus, making it non-neutral from the server room power budget point of view. With 25C (Fig. 4), 30C (Fig. 5) and 35C (Fig. 6) the cooling power might be increased by increasing the flow in the secondary loop, but it comes at an additional cost. It can also be observed that with the inlet temperatures below the ambient air temperature, the cooling loop delivers more cooling power than the server is producing heat. This means that in such conditions the server is cooling down the data center. Whenever the difference between set points on inlet and outlet temperatures increases, one can observe an increasing erratic behavior of the coolant flow. It is caused by the delay between actions issued by the control system and the reactions in form of change of the temperature of the coolant (as we mentioned earlier). Whenever the system detects an

increased temperature of the coolant exiting the servers, it slowly starts opening the valves to the heat exchanger, then speeds up the pumps to keep the coolant temperature within the defined range.

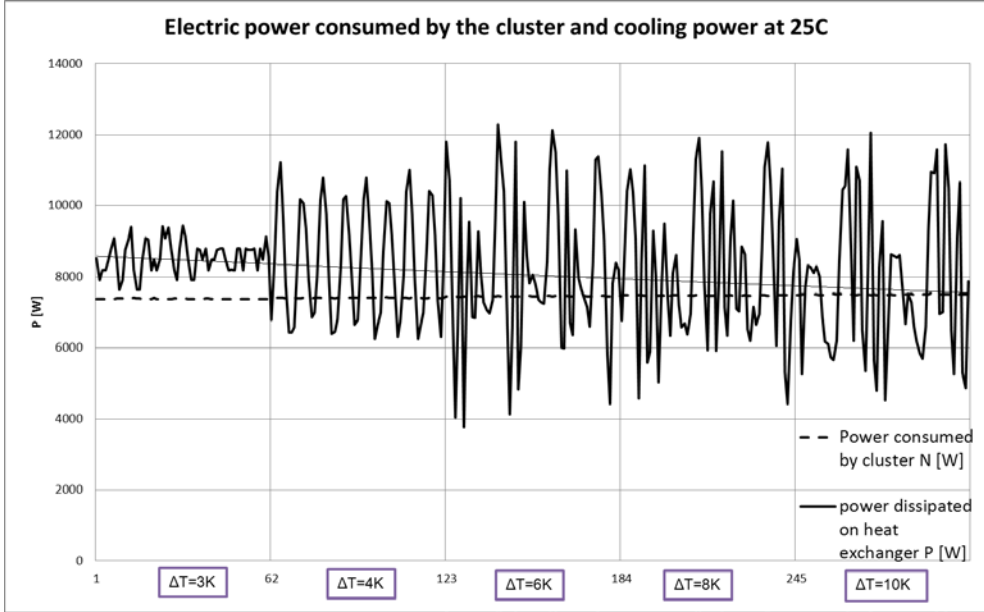


Figure 4. Power consumed by the cluster and cooling loop power at 25C inlet

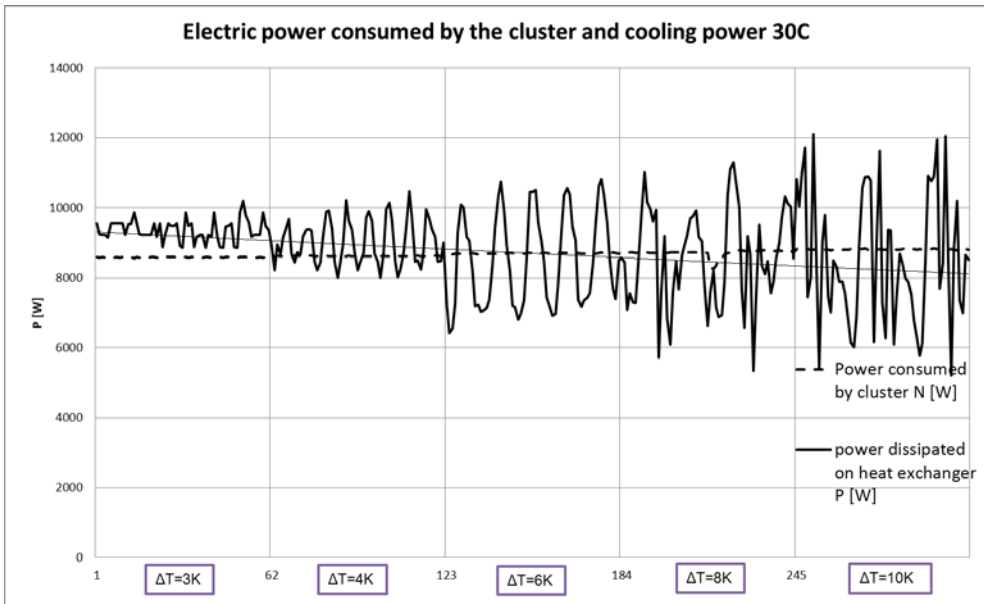


Figure 5. Power consumed by the cluster and cooling loop power at 30C inlet

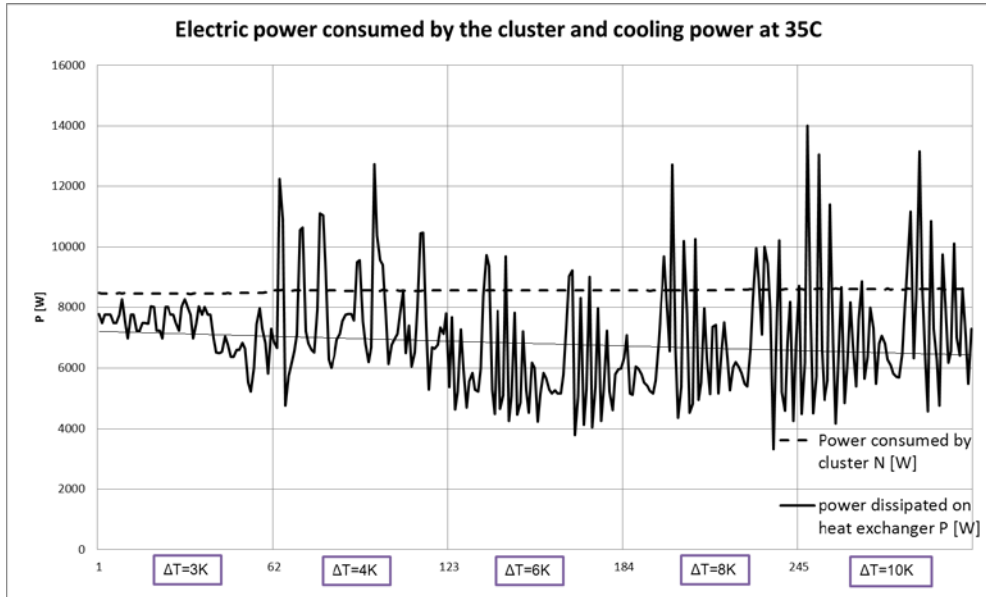


Figure 6. Power consumed by the cluster and cooling loop power at 35C inlet

Because all the sensors are physically installed in the control rack, the coolant has to traverse more than 50m of pipes before the change of the temperature can be detected. Therefore, when the coolant reaches the sensors, it is colder than anticipated so the control system issues a rapid slow-down of the cooling loop and the oscillation occurs. This is affecting the energy efficiency of the cooling loop, as the pump settings are higher than necessary. It can be dealt with using an additional coolant tank in the loop that will act as a buffer, unfortunately it was not installed. We assumed that these issues can be addressed by proper management of the cooling loop. The results of the experiment will be presented in the following sections.

One can see that the temperatures above the 25C are perfectly fine from the CPU point of view, as there was no degradation of the CPU performance. The cooling loops parameters should also be stabilized in order to prevent unnecessary efficiency loss.

In the following section we will focus on the 20C run, as it is the best representation of a case where we can gain energy usage benefits by smart management of the loop.

Looking at Figure 7 one can see that by changing the desired delta add thus altering the flow it is possible to reduce the power consumed by the main pump more than 10 times. With the most efficient pump setting, with delta of 8K and 10K, one can see periodic changes in the amount of power expelled through the heat exchanger, the cooling power of the loop. Whenever the cooling power is lower than the power consumed by the servers, the excess power has to be cooled by the data center air conditioning system, therefore this setting is not optimal from the efficiency point of view. Each cycle of the test (for each

delta) takes 60 min, this means that the servers used 8,36 kWh of electrical power. For 10K delta because of the fluctuation almost 0,4 kWh heat was expelled to the data center air. Basing on the data collected in our data center, the air conditioning used 0,118 kWh to

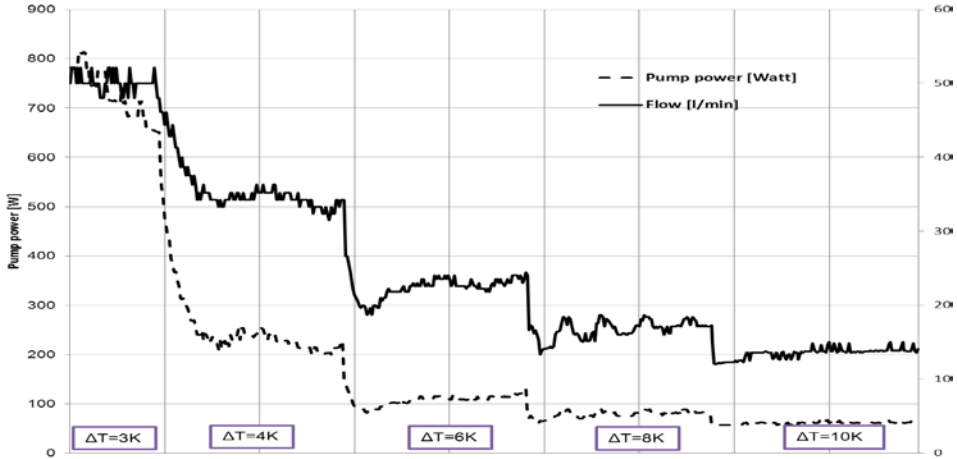


Figure 7. Cooling loop parameters at different temperature deltas for 20C inlet coolant

remove this heat while the liquid cooling loop required only 0,14 kWh to remove the rest of the heat: 7,96kWh. This clearly points out that one should avoid heating up the server room, as cooling it down is way less efficient then using liquid cooling.

Looking at the figure one can observe that the pump setting is not the source of the problem, as the power draw does not change rapidly, especially when the delta was set to 10K. Taking into account the cooling loop management algorithm described in the previous section, the source of the problem is lack of stability of the valve setting. The three-way valve is being closed and opened in a periodic fashion, while cooling loop tries to keep a stable temperature. To prevent the osculation from occurring, we switched from the second management mode of the cooling loop to the third, fully manual one, and prepared a customized configuration algorithm.

Instead of the management algorithm that was using the sensors in the server and cooling loop to adjust valves and pumps, we relied on the state queue system to predict what the load of the machine will be. The power was predicted using the following formula:

$$P_c = p_s * n + p_m * m [W]$$

where:

P_c = power consumed by the cluster

p_s = power consumed by idle processor P

p_m = power consumed by working processor P

$n = \text{number of idle processors}$

$m = \text{number of working processors}$

This very simple formula has proved to be very accurate, the difference between real measurements and the predictions were in 5% tolerance. The measurements were taken every 1 min. but, according to the specifications of the power meter, the value was the averaged value of power draw sampled 100 times per second. This kind of accuracy was possible because in the test scenarios only one type of application was used: High Performance Linpack (HPL). Every instance of the test was executed in a way that it used all resources on a single node and no other applications were running on the cluster. Therefore every node could be either under full load or idle. HPL is an application that is using the computing resources in a consistent fashion, stressing the processor and the memory.

Then, for each possible combination of idle and active processors (different power draws), by searching through the data collected during the prototype usage, we found a configuration of the valves, pump and fan settings where the cooling power of the loop was greater than the consumed power, while the power consumed by the loop was minimal. The selection takes into account also external temperature, because the efficiency of the loop depends on the performance of the roof top heat exchanger. Because the test runs were executed in November the 24h delta was 6K, so the number of configuration profiles was limited to 60 – the resolution of the temperature sensors that is 1/10K limits the number of profiles.

A four dimensional array was created:

$$A[T_e, T_i, \Delta T, P_c]$$

where:

$T_e = \text{external temperature}$

$T_i = \text{coolant cluster inlet temperature}$

$\Delta T = \text{difference between cluster outlet and inlet temperatures}$

$P_c = \text{power consumed by the cluster}$

Each element is a vector $S[P_s, V_s]$,

where:

$P_s = \text{pump setpoint}$

$V_s = \text{valve setpoint}$

Our goal was to find a configuration that generated the least possible overhead for cooling the test cluster. We looked through the data to find (for given sets of parameters) the minimum value of the function:

$$P_{cl} = P_f + P_p + P_w * E$$

where:

$P_{cl} = \text{power consumed cooling loop}$

P_f = power consumed by fans on heat exchaneger

P_p = power consumed by the pump

P_w = power needed to cool the waste heat

$$\begin{cases} P_w = 0, P_c - P_{cp} \leq 0 \\ P_w = (P_{cp} - P_c) * E, P_c - P_{cp} > 0 \end{cases}$$

and:

P_{cp} = cooling power of the cooling loop

E = efficiency of the air cooling

Cooling power of the loop was acquired from the cooling loop controller, and the efficiency of the air cooling solution was an average value for our data center calculated from the power supply records of the data center.

Because the algorithm gives only an approximation of the power consumption and the temperature and flow sensors have limited accuracy, we implemented also the same algorithm as we used previously, but it was allowed to change the parameters only by 10% from the set values in order to allow some flexibility. In addition, a 30s of delay was introduced between job being assigned to a given node and the actual execution of the job. This ensures that the cooling loop has enough time to reach the desired parameters before the actual job is executed.

Repeated test run for all deltas (3K, 4K, 6K, 8K and 10K) presented in Figure 8 show that the approach was successful and the erratic behavior was almost completely removed.

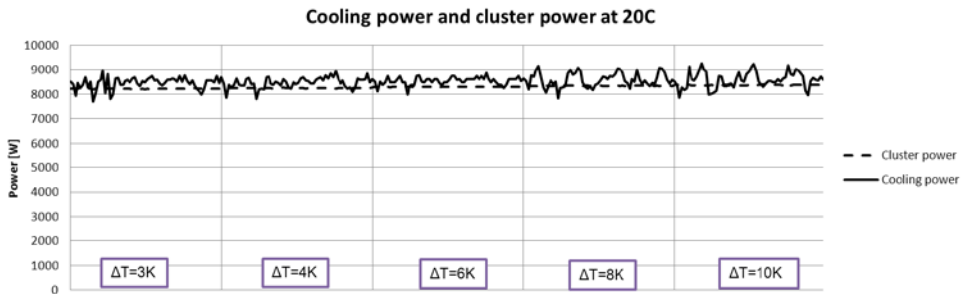


Figure 8. Cluster power and cooling loop cooling power with fixed management algorithm

One can see that, because of a more consistent cooling power profile, the amount of power expelled to the data center room was reduced from 0,4kWh to 0,05kWh, proving that by tight coupling of computing equipment and data center infrastructure one can gain substantial benefits in form of saved energy.

5. Conclusions: benefits and challenges

Using direct liquid cooling is undauntedly more efficient method of cooling, including cooling of supercomputers. It enables a significant reduction of the operation costs of the computing infrastructure. Unfortunately, rapid changes and unpredictable characteristics of the computer load introduce challenges that may impact the efficiency of the solution.

At first glance, working on 35C/40C inlet all the time should be an acceptable solution, and ensures that the servers can operate within right temperatures. Increasing the temperature of the coolant, as advocated by cooling equipment vendors, is undoubtedly beneficial from the cooling-loop energy efficiency point of view, and it simplifies the management. It allows to increase a number of hours during the year when one can bypass the chillers and thus save the energy. However, if one looks at the global power consumption of the entire ecosystem, the problem caused by higher temperatures may overshadow the benefits.

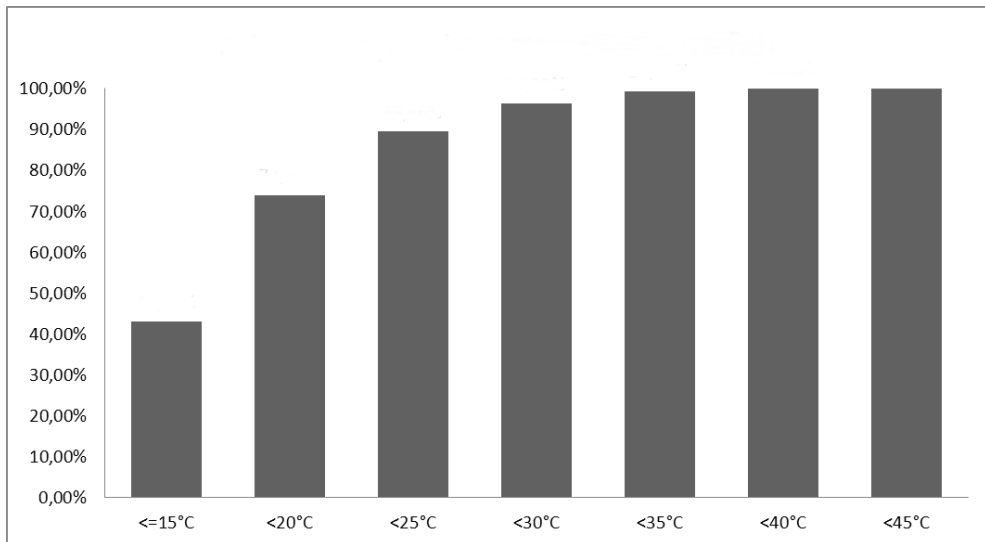


Figure 9. Distribution of the temperature returning from the rooftop heat exchanger

CPUs that work in a warmer environment are characterized by increased power consumption. Increasing both the inlet temperatures and the inlet-outlet delta, and thus reducing the pump activity, results in an increase of the temperature of the chips, and therefore the energy consumed by the system. In case of our prototype, the maximum power draw difference between the system cooled by 15C and 35C, we had been able to measure, was 10%.

Therefore, considering the cooling solution presented in the paper, keeping the CPUs as cold as possible seems to be the right choice. This statement may not always be true for solutions that use chillers to provide cold water, as in some cases the power consumed by the chillers may overshadow the gains from running in lower temperatures. In the solution presented, however, there are chillers in the primary loop, so “as cold as possible” means

“as close as possible to the ambient air temperature that surrounds the water-air heat exchanger”. Usage of the emergency loop that employs chiller is justified only when the coolant exceeds the maximum tolerable temperature. Looking at the distribution of the temperatures (Fig. 9) in the cooling loop we were able to achieve in 12 month time, it should be possible to operate with 25C inlet coolant in almost 90% hours in year.

In this temperature range the erratic behavior of the loop is even more visible than the example presented before. Therefore we can assume that the benefits one can achieve by using the presented optimization will be even more significant.

The experiment showed that while the general idea of using warm water as a coolant may bring great benefits in terms of energy bill savings, it may be improved by changing the way one is managing the cooling loop. Being able to predict accurately the power demand of the cluster allowed us to solve the problems that arose from physical limitations of the cooling loop instrumentation. Regardless of a simplified model, our tests show that by closing the gap between infrastructure and computation environment is beneficial for the entire ecosystem.

Practical implementation outside of the controlled benchmark environment requires a good power prediction algorithm that can accurately assess power demands for applications that are to be executed on the cluster. In addition, better numerical representations of the infrastructure properties are needed for better infrastructure management.

Finally, while we have shown that the management of the infrastructure that relies completely on the sensors is not optimal, a sensor-based feedback about the infrastructure may be provided to the computer management system for more complete awareness about the current and predicted resource state for more efficient scheduling.

Acknowledgment

This research is a part of the project no. 2013/08/A/ST6/00296, funded by the Polish National Science Centre.

References

- [1] Greenberg S., Mills E., Tschudi B., Rumsey P., Myatt B., Best Practices for Data Centers: Lessons Learned from Benchmarking 22 Data Centers, 2006 ACEEE Summer Study on Energy Efficiency in Buildings, 2006.
- [2] HPL: High Performance Linpack, used optimized intel HPL implementation: <https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite>.
- [3] iDataPlex: <https://lenovopress.com/tips0878-idadaplex-dx360-m4>.
- [4] Iyengar M., David M., Parida P., Kamath V., Kochuparambil B., Graybill D., Schultz M., Gaynes M., Simons R., Schmidt R., Chainer T., Server liquid cooling with chillerless data center design to enable significant energy savings, 28th Annual IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), 2012.
- [5] Januszewski R., Gilly L., Yilmaz E., Auweter A., Svensson G., Cooling – making efficient choices, PRACE (Partnership for Advanced Computing in Europe) Whitepaper, 2016.
- [6] Januszewski R., Meyer N., Nowicka J., Evaluation of the impact of direct warm-water cooling of the HPC servers on the data center ecosystem, Proceedings of ISC 2014 Conference, 2014.

-
- [7] Nguyen C.T., Roy G., Gauthier C., Galanis N., Heat transfer enhancement using Al₂O₃-water nanofluid for an electronic liquid cooling system, *Applied Thermal Engineering*, **27**, 2006, 1501-1506.
 - [8] Novec: https://www.3m.com/3M/en_US/novec-us.
 - [9] Open Compute Project: www.opencompute.org.
 - [10] PSNC: Poznan Supercomputing and Networking Center, <http://www.man.poznan.pl/online/en>.
 - [11] Rasmussen N., Implementing Energy Efficient Data Centers, Schneider Electric Whitepaper, 2014.
 - [12] Semenov O., Vassighi A., Sachdev M., Impact of Self-Heating Effect on Long-Term Reliability and Performance Degradation in CMOS Circuits, *IEEE Transactions on Device and Materials Reliability*, **6**, 2006, 17-27.
 - [13] SLURM: <https://slurm.schedmd.com>.
 - [14] Steinberg D.S., *Cooling techniques for electronic equipment*, New York, Wiley-Interscience, 1980.

Received 20.10.2016, Accepted 20.03.2018