Distributed Thermal Aware Load Balancing for Cooling of Modular Data Centres

Joseph Doyle [a], Florian Knorn [b], Donal O'Mahony [a], Robert Shorten [b]

[a] *CTVR, Trinity College Dublin, Ireland*

[b] *Hamilton Institute, National University of Ireland, Maynooth, Ireland*

Abstract

Thermal management in data centres requires a complicated trade-off between cooling costs and thermally induced equipment failure rates. Using ideas from cooperative control and distributed rate limiting, in this paper we describe a distributed architecture that can be used for thermal aware load balancing for a common type of modular data centre; namely an algorithm that, for a given demand D^*, distributes load to individual machines such that the temperatures in the individual modules are equalised. The benefit of shifting load based on thermal considerations is that significant gains in cooling cost can be achieved. We evaluate the performance of the algorithm using computational fluid dynamics (CFD) and MATLAB simulations. Our results show that significant cost savings can be made by applying such algorithms, and that these can be achieved without the need for detailed modelling and tuning of controllers.

Key words: thermal equilibrium; Load balancing algorithms; stability; distributed control; computational methods

1 Introduction

In this study we examine air temperature management in buildings and data centres. A data centre is a warehouse-scale building which may contain up to tens of thousands of high performance servers as well as the associated networking and cooling equipment, thereby producing large amounts of heat as they perform their tasks. In large scale data centre operations, the modular data centre approach is becoming more and more popular, [6]. Such data centres are built from a number of independent, autonomous modules (typically housed in standardised overseas shipping containers), that each contain a large number of servers. Computer room air conditioning (CRAC) units are used to extract heat coming from the servers, cool it, and provide the cooled air back to their inlets, so as to prevent equipment damage. Cooling equipment is clearly crucial to long-term reliability of operation; it has been shown for instance that every 10 °C increase above 21 °C decreases the long-term reliability of electronics by 50% [13]. In addition, a 15 °C rise increases the failure rates of hard disk drives by a factor of two [2]. It is also possible to use water cooling in data centres [20,1] but the majority of data centres use air cooling.

The cost of providing cool air to the server inlets can be considerable. Consider a 30,000 ft^2 data centre with 1000 standard computing racks, each consuming 10 kW. With an average electricity cost of $100/MWh the annual cooling cost would be in the order of $4–$8 million, [17]. It is thus clearly in the interest of the data centre operator to operate the facility as efficiently as possible to minimise such costs. Similar problems also arise elsewhere, for example in thermal management of buildings. Initiatives such as IBM's "Smart Cities" program have been created with this and similar management problems in mind.

Our objective in this paper is to address thermal management in a data centre which utilises modular data centre components. Note that there have been numerous proposals for the thermal management of servers in data centres [21,13,8,16]. A notable deviation to this work [9], considers a combination of the cooling power for CRAC units and the idle and dynamic power of servers in a load balancing scheme to lower overall energy costs. While this appears to be a new approach to the problem, in this paper we consider the traditional approach to achieve

⋆

Email addresses: doylej4@tcd.ie (Joseph Doyle), florian@knorn.org (Florian Knorn), Donal.OMahony@cs.tcd.ie (Donal O'Mahony), robert.shorten@nuim.ie (Robert Shorten).

cost of minimisation. In addition, there is an increased trend toward energy proportional servers [4] which represent advanced servers where the idle power is virtually zero and the idle power of servers is likely to be less important in the future. All of these proposals, however, rely on a central scheduler, which is both a bottleneck for control messages and a single point of failure. Such central solutions typically require lengthy calibrations to minimise operational costs. A distinctive feature of the present work, however, is that we would like to propose a robust distributed algorithm which attempts to reduce cooling costs, prevents equipment damage while satisfying demand requirements, all without the need of detailed calibrations. Such an algorithm would, with its *plug & play* functionality, be eminently applicable in a modular environment. Specifically, the contributions of this paper are as follows:

- We present a distributed algorithm to both regulate total demand serviced to a certain given level, subject to the additional constraint that temperature is equalised amongst all machines.
- In certain situations, when more demanding requirements are placed on the data centre, we show that existing algorithms from communication networks can be applied in this context.
- Finally, data from an industry standard computational fluid dynamic (CFD) Flovent [7] is used to validate the algorithms.

The performance of the proposed algorithms in minimising the total cooling cost is evaluated. It will be shown that the total cooling costs can be reduced significantly while servicing a desired demand.

The application described in this paper uses algorithms first described in [11] and [22]. The specific contribution of this work is to apply and develop the ideas in these papers to problems arising in thermal management of data centres. A detailed mathematical analysis of these algorithms is given in the aforementioned papers. Hence, in the sequel, we give only very brief mathematical details.

2 Preliminaries

We shall study thermal management in the context of a large scale data centre. In the following, we shall use the term "machine" in a rather abstract sense in that we assume that each "machine" actually consists of a housing that contains a large number of individual servers which, jointly, have a cooling facility associated with them. This assumption is justified, for instance, by the increasingly popular modular data centres, where large clusters of servers are housed in autonomous shipping containers that are all connected to a common chilled water supply, [6]. Such a container would then be considered a "machine" in the context of this paper.

2.1 Problem setting

We now consider a data centre that is constructed using n machines. Each machine $i \in \{1,2,\ldots,n\}$ has, at time $k = 0,1,2,\ldots$, an inlet temperature (or just "temperature") of $T_i(k)$, which represents the temperature of the air sucked into the servers for cooling. We assume that this temperature is the same for all the servers inside a given machine; this assumption is justified in particular when suitable cold aisle containment is used inside the machine (Our simulations show that the largest difference in temperature between the inlets of servers is 0.3 °C). In addition, simulation 3 examines how much the demand of neighbouring server racks affects the temperature. Furthermore, even in the presence of this defect the algorithm works well. Note our simulations are based on Flovent [7], an industry standard simulator, and these simulations indicate that even in the presence of this effect, the algorithm is efficacious. We also assume that there is sufficient distance between the machines that heat exchanges can be neglected and that these are cased and isolated from each other. Recall that machines refer to housings that contain a large number of individual servers. Additionally, each machine is servicing a demand (also referred to as "work load") of $D_i(k)$, so that the total demand serviced by the data centre at time k is

$$D(k) := \sum_i D_i(k) \tag{1}$$

An important feature of our work is that the total demand $D(k)$ is regulated to some desired value D^*, which may be time-varying (for notational convenience, we omit the dependence of D^* on k). Hence, the value of the total demand serviced and its deviation from D^* must be known either implicitly or explicitly by at least one machine in the data centre.

Given a constant amount of cooling energy supplied to each machine, the temperatures inside each machine will be a direct function of the demand serviced by that machine (since the heat energy dissipated by the CPUs is roughly proportional to the amount of work done by the CPUs). Borrowing from the terminology established in [11], this interdependency is described by the "utility functions", which relate the "physical state" (demand) D_i of machine i to its "utility value" (temperature):

$$T_i = f_i(D_i) \tag{2}$$

Note that f_i is typically non-linear and is used to model complicated fluid dynamic effects as well as natural cooling within the machines. We assume that the manner in which workload is distributed inside the machines is uniform.

Next, we assume that a limited information exchange between machines is possible. Specifically, at time k, ma-

chine i can provide information about itself (in particular its current temperature) to another machine j if and only if (i,j) is an arc in the directed communication graph $\mathcal{G}(k) = \big(\mathcal{N},\mathcal{E}(k)\big)$, which thus describes the topology of the possible information exchange between machines. This graph is allowed to change over time and is assumed to be jointly strongly connected over a given, fixed time horizon $m \geq 1$. By this, we mean that every union of m consecutive graphs is assumed to yield a strongly connected graph. [1]

We would now like to find an algorithm that will attempt to equalise the temperatures among machines, while ensuring that some desired demand D^* is being serviced by the data centre.

2.2 A cooperative control scheme (Algorithm 1)

The recent publication [11] gives an in-depth discussion of three iterative algorithms and variations thereof that are designed to allow a network to achieve a common goal cooperatively while satisfying certain local constraints. The data centre load balancing problem fits into this framework and we shall now reproduce, for convenience, some of the mathematical statements from this publication (in particular Theorem 4.1).

To apply these results, we must assume that the utility functions are continuous, monotone functions with bounded growth rates (to guarantee feasibility of the solution) and that each machine has knowledge of the total demand being serviced by the data centre. Note it is also assumed that the update law for the algorithm uses a time scale that is larger than that of the dynamics of the settling time for the temperatures; namely that the relationship between D_i and T_i can be adequately modelled using a static map. Given these assumptions, the following theorem (adapted from Theorem 4.1 in [11]) provides a stable update law to iteratively refine the demands being serviced by the individual machines so that the data centre converges to the desired behaviour:

Theorem 1. *For some initial condition $D_i(k=0)$ and any sequence of strongly connected communication graphs, suppose that the machines iteratively update their work load according to*

$$D_i(k{+}1) = D_i(k) + \sum_{(j,i)\in\mathcal{E}(k)} \eta_{ij}(k)\big(T_j(k)-T_i(k)\big) + \mu(k)\sigma(k) \tag{3}$$

where

$$\sigma(k) = \begin{cases} D^* - \sum_{i=1}^{n} D_i(k{+}1{-}M) & \text{if } k{+}1 \text{ is a multiple of } M \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

with $M := n-1$. If the gains satisfy

$$0 < \underline{\mu} \leq \mu(k) \leq \bar{\mu} \quad \text{and} \quad 0 < \underline{\eta_i} \leq \eta_{ij}(k) \leq \bar{\eta}_i \tag{5}$$

then — provided $\bar{\mu} > 0$ and $\bar{\eta}_i > 0$ are sufficiently small — the demands $D_i(k)$ converge asymptotically to values D_i^* for which $f_i\big(D_i^*\big) = T^*$ for all $i = 1,\ldots,n$ and $\sum_i D_i^* = D^*$.

For a proof of the Theorem and a much more detailed description of the mathematical assumptions therein, please refer to [11].

Comment. Explicit bounds on $\bar{\eta}_i$ and $\bar{\mu}$ are also given in [11]. While for the purpose of proving convergence in the theorem small values of these constants are required, in practical situations it is found that they can be significantly larger. Mathematical details are given in [11]. These are quiet involved and are beyond the scope of the present paper. Roughly speaking, the stability conditions are based on connectivity arguments in graphs. As the graphs become large, then these conditions give small controller gains. This approach does not account for structural properties of the graph and consequently may be quite conservative.

The update law (3) from the Theorem, which we will refer to as Algorithm 1, thus provides a rule specifying how to iteratively update the load on the machines to balance the temperatures among the machines in the network. It basically consists of two intuitive-to-understand parts: The first part, summing over the differences in temperatures between neighbours, is aimed at reducing the temperature differences; if, for instance, all neighbouring machines of machine i are operating cooler than machine i, then it should reduce its own demand in order to approach the temperature level of the surrounding machines. The second part in the equation is to ensure that the global demand is satisfied. It is easy to see that if the total demand serviced by all the machines is below the required quantity, then each machine should increase their own work load somewhat so that the network, jointly, increases the demand serviced until it reaches the desired level.

The pseudocode shown in Figure 1 describes an implementation of this algorithm.

2.3 A notable simplification (Algorithm 2)

In situations where the communication graph is undirected, where the total required demand is not subject to

[1] The union of a set of graphs on a common vertex set is defined as the graph consisting of that vertex set and whose edge set is the union of the edge sets of the constituent graphs.

```
UpdateDemand()
   Once every Δ units of time do
     for i = 1 : n
       D_i ← D_i + η Σ_{(i,j)∈E} (T_j − T_i)
       if mod(k + 1, n − 1) == 0
         D_i ← D_i + μσ
       endif
     endfor
     if mod(k + 1, n − 1) == 0
       σ ← (D* − Σ_{i=1}^{n} D_i)
     endif
     k ← k + 1
   enddo

InitialiseDemand()
   k ← 0
   σ ← 0
   for i = 1 : n
       D_i ← D*/n
   endfor
```

Figure 1. Pseudocode for Algorithm 1.

```
UpdateDemand()
   Once every Δ units of time do
     for i = 1 : n
       D_i ← D_i − η Σ_{(i,j)∈E} (T_i − T_j)
     endfor
     k ← k + 1
   enddo

InitialiseDemand()
     k ← 0
     for i = 1 : n
         D_i ← D*/n
     endfor
```

Figure 2. Pseudocode for Algorithm 2.

change, and where the utility functions satisfy stronger assumptions, then considerable simplifications are possible. In particular, the algorithms given in [22] apply. Specifically, suppose that: (a) the communications graph is undirected; (b) the desired demand D^* is constant; (c) the utility functions $f_i : \mathbb{R} \to \mathbb{R}$ are increasing, concave, differentiable functions, and have continuous first derivatives. Then the algorithm described by the pseudocode in Figure 2 may be used to solve the thermal management problem posed above.

The basic idea encapsulated above is as follows. Initially, the demand is divided evenly among the server racks. The demand D_i of each machine i is then iteratively updated with a term that is proportional to the sum over the differences between the own temperature and that of neighbouring racks, that is

$$D_1(0) = D_2(0) = \ldots = D_n(0) \tag{6a}$$

$$D_i(k+1) = D_i(k) - \eta \sum_{(i,j)\in\mathcal{E}} \big(T_i(k) - T_j(k)\big) \tag{6b}$$

The gain parameter η determines the responsiveness and stability of the algorithm and its choice is discussed in [22]. The stability and convergence properties are captured by the following theorem, cf. [22]. Before stating this theorem, we need to establish some further terminology. Denote by $g_i = f_i^{-1}$ the inverse functions of the f_i, which must exist given the convexity and differentiability assumption above. Note also, due to symmetry, the system given by (6) satisfies the demand constraint (1) for all $k = 0,1,2,\ldots$.

Theorem 2. *Let d_i be the degree of node i in the communication graph. If η satisfies:*

$$0 < \eta < \frac{1}{2} \min_{1\le i\le n} \Big[- g_i'\big(T_i(0)\big)\Big] \min_{1\le i\le n} \frac{1}{d_i}, \tag{7}$$

then the system given by (6) *will converge to*

$$\lim_{k\to\infty} D_i(k) = D_i^*$$

with $\sum_i D_i^* = D$, and

$$\lim_{k\to\infty} T_i(k) = T^*$$

for all $i = 1,\ldots,n$.

Proof. See [22]. □

Δ is indeed related to the settling time associated with the $f_i()$ functions. In the original work on this topic [11], these functions are static maps where as in this application this is not entirely true. Our basic assumption is that the dynamics associated with the $f_i()$ functions are fast when compared with the update law. Of course this assumption has to be validated experimentally and the step size determined empirically. For our simulation Δ was always chosen to be much larger than the dynamics associated with the $f_i()$ functions. Recall the objective of this paper was illustrate the use of a new and innovative distributed control algorithm developed in [11] for this novel application. In our CFD simulations each iteration of the Flovent software was run every Δ seconds.

Comment. Before proceeding to evaluate in simulation the algorithms described in this paper, we note here that the decentralised control advocated in this paper is motivated by recent developments in consensus algorithms and in gossiping algorithms. In this context it

may be viewed as a somewhat non-traditional approach to decentralised control systems. The interested reader is referred to the work of Jadbabaie *et al.* [10], Olfati-Saber and Murray [15], Lin *et al.* [12] and Moreau [14] for initial ideas in this direction. More recent developments were presented by Knorn *et al.* [11] as well as Stanojević and Shorten [22]. These later articles provide much of the theoretical framework for the specific work described in this paper.

3 Simulation and validation

To evaluate the performance of the algorithms we used a CFD simulator. Three simulations were conducted. In the first simulation, we consider a data centre with twelve machines of three different types (that is, four machines of each type). Data from the CFD simulator was used to generate the utility functions, and MATLAB based simulations were then used to evaluate the ability of Algorithm 1 to track a given time-varying demand. We then repeat this simulation, using Algorithm 2, in the case where the communication graph is undirected and the demand is fixed. Finally, we then apply Algorithm 2 to a conventional (non-modular) data centre, where "machines" are now represented by server racks inside a single room. In this situation, to capture the extremely complicated fluid dynamic effects, the full simulation was carried out using the CFD simulator. Finally note that while the gains in each situation may be chosen in accordance with the bounds given in Theorems 1 and 2 (and the respective references), larger values were in fact used to speed up convergence for the purpose of exposition.

3.1 Simulation setup

Our simulation setup follows that used in [21,13]. Each machine (data centre) used in this study has dimensions 11.7 m × 8.5 m × 3.1 m with a 0.6 m raised floor plenum that supplies cool air through perforated floor tiles. There are four rows of servers with seven 40 U racks in each case, resulting in a total of 1120 servers. The dimensions of the server racks are depicted in Figure 3(a). Note the front and rear doors were removed to allow the air to flow freely. The servers simulated were based on Hewlett-Packard's *Proliant DL360 G3s* model, which consumes 150 W of power when idle and 285 W at 100% utilization. From this we could determine that the total power consumption of the data centre is 168 kW (40 × 28× 150 W) when idle and 319.2 kW (40 × 28× 285 W) at full utilisation. The flow rate of the server rack was 1,500 $\mathrm{ft}^3/\mathrm{min}$ representing 40 servers with a flow rate of 37.5 $\mathrm{ft}^3/\mathrm{min}$. We also used an ideal *energy proportional* version of said server, [4]. Recall that these represent advanced servers where the idle power is virtually zero.

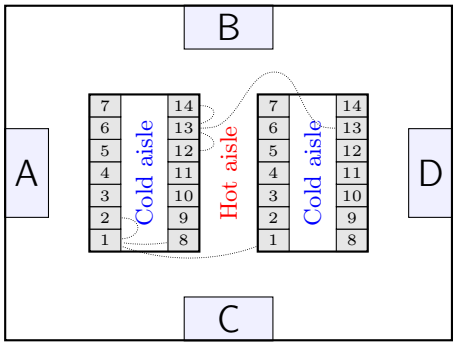
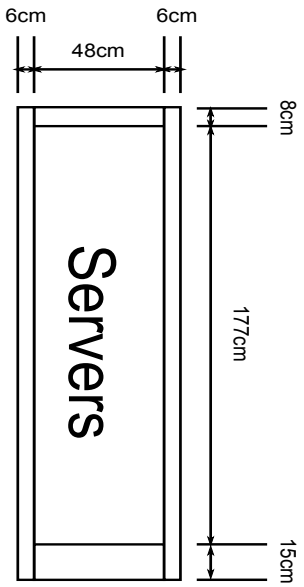


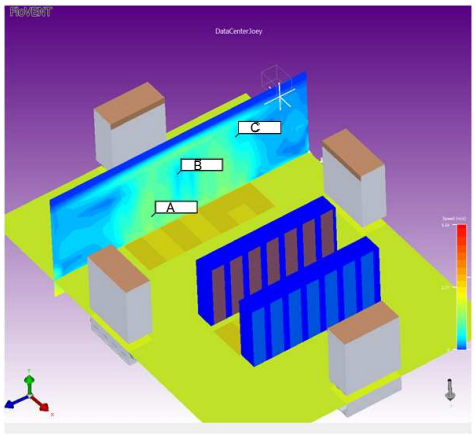
Figure 4. Layout of simulation setup. To illustrate the communication graphs used in simulation 3, the dotted lines indicate which other machines the racks 1 and 13 from the left cold aisle can exchange temperature information with. Server racks in adjacent aisles are connected to lower temperature differences between aisles.

For cooling, the data centre is equipped with four CRAC units "A", "B", "C" and "D" whose locations are also indicated in Figure 4. Each CRAC unit pushes air chilled to 15 °C into the plenum at a rate of 10,000 $\mathrm{ft}^3/\mathrm{min}$. The cooling capacity of the each CRAC unit is limited to 90 kW, and in full operation each CRAC unit itself consumes 10 kW. Flovent basically uses a k-ϵ turbulence model; see documentation describing mathematical modelling given as part of [7]. The air velocity, pressure and turbulence at the front boundary of the server racks are depicted in Figure 3(b), 3(c) and 3(d). CFD simulations are now used routinely in data centre design and have been found be to very useful by practising engineers. There is, however, a need for detailed validation against experimental data. These concerns, however, are not addressed in this paper and CFD simulations are carried out using the popular commercial package Flovent.

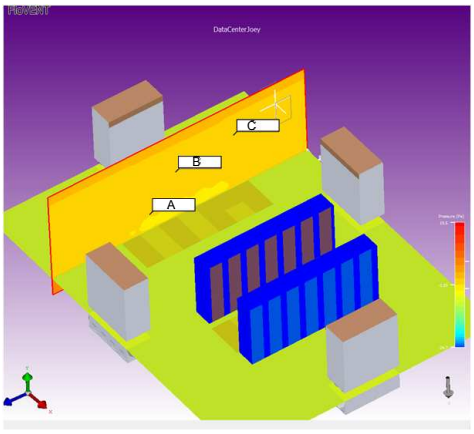
Given this simple setting, we now describe three basic variations that may arise. In the first case, the machine is exactly as described above with the standard HP servers. This type is hereafter referred to as "NC-NF". For the second case, we decided to model the machine using a "cold aisle containment" assumption, [19] with CRAC unit D offline. Recall that cold aisle containment refers to the segregation of the cold aisle from the rest of the data centre using physical barriers such as PVC curtains or Plexiglas [5]. This type of data centre is likely to become more and more popular in the future, [18]. This type is hereafter referred to as "C-F". It is also possible to use "hot aisle containment" which is the segregation of the hot aisle from the rest of the data centre. While there is some evidence that "hot aisle containment" may be a more efficient design, it is recognised [18] that there are difficulties in retrofitting this solution which may make "cold aisle containment" the more popular design, at least in the short-term. The third case is identical to the second case, but servers of the *energy proportional* type were used. This type is hereafter referred to as "C-F-P". Each case has a utility function associated with it



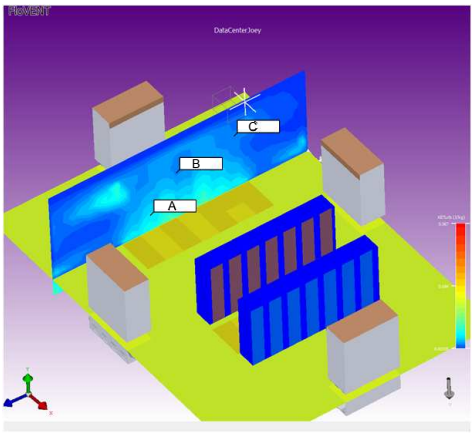
(a) Geometry of the server rack. All components in the server rack are 60cm thick.



(b) Air velocities in front of the server rack. At point A the air velocity is 1.18 m/s. The air velocity at point B is 0.38 m/s and at point C the air velocity is 0.622 m/s.



(c) Pressure conditions in front of the server racks. At point A the pressure is −2.75 Pa. The pressure at point B is −0.61 Pa and at point C the pressure is −1.13 Pa.



(d) Turbulence in front of the server racks. The turbulence at point A is 0.0337 J/kg. At point B the turbulence is 0.0102 J/kg and the turbulence at point C is 0.01 J/kg

Figure 3. Geometry of the server rack as well as the air velocity, pressure and turbulence conditions in front of the server racks

that describes the relationship between load and maximum inlet temperature found inside the machine. These utility functions, which we shall later use in our simulations, are depicted in Figure 5.

In order to evaluate the performance of the algorithms we need to be able to calculate the cooling costs. Let Q be the amount of power the servers consume, T_{sup} the temperature of the air that the CRAC units supply, $T_{\text{safe}} = 25\ ^{\circ}\text{C}$ the maximum permissible temperature at the server inlets in order to prevent equipment damage, T_{max} the maximum temperature of the server inlets in the machine and P_{fan} the power required by

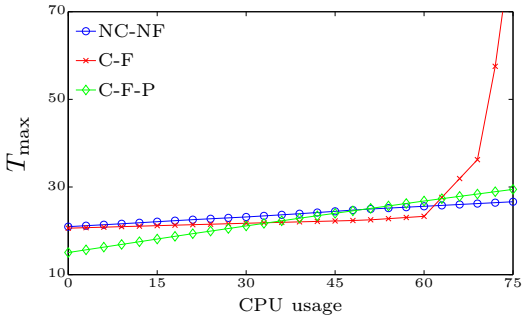


Figure 5. Utility functions.

the fans of the CRAC units. The "coefficient of performance" (COP), that is the ratio of heat removed to work necessary to remove the heat, is a function of the supply temperature T_{sup} being provided by the CRAC. There is considerable debate over the maximum permissible temperature at the inlet in order to prevent equipment damage. While there is a guidelines in this area it has changed recently [3] and we selected the value $T_{\text{safe}} = 25\ ^\circ\text{C}$ as it is a well established value which has been used in other systems [21,13]. The cooling cost C can then be calculated as:

$$C = \frac{Q}{\text{COP}(T_{\text{sup}})} + P_{\text{fan}} \tag{8}$$

If the highest temperature found at any inlet in the data centre is below the "red-line" temperature, then the CRAC is cooling the data centre excessively. In such a situation, the supply temperature can be raised by $T_{\text{safe}} - T_{\text{max}}$ (by reducing the amount of cooling) to reduce costs while still observing T_{safe}.

Comment. While other factors such as complex nonlinear flow effects and the cost of pumping air to difficult-to-reach parts of the data centre affect the cooling cost, the highest machine temperature is a major factor in the cooling cost. This observation is what motivates us to equalise temperatures.

If, in turn, T_{safe}–T_{max} is negative the equipment is in danger of being damaged and the supply temperature must be lowered in order to cool down the machine responsible for T_{max}. In our simulations, to determine this maximum inlet temperature, we used the commercial CFD simulator FLOVENT. Each of the four CRAC fan units consumed 10 kW so that for each machine $P_{\text{fan}} = 40$ kW and $T_{\text{sup}} = 15\ ^\circ\text{C}$. The COP curve used to calculate the cooling costs is a standard curve for a water chilled CRAC and is given in [13].

3.2 Simulation 1

As mentioned above, the setup for this simulation consists of a modular data centre site housing twelve machines (containers), four of each type described above. Our objective in the following is to regulate the aggregate CPU load to three levels: 40%, then 55% and finally 25% in a situation where the resulting communication network is strongly connected, but chosen randomly. The resulting evolution over time of the aggregate demand, individual demands and individual temperatures are given in Figure 6. The simulations are measured in terms of iterations for reasons which are discussed in Section 2.3. As can be seen, the cost savings achieved in the equilibrium for the 40% level are significant when compared to the initial temperature configuration. The maximum rack inlet temperature drops by approximately 1 °C and consequently the cooling cost drops from 1.514 MW to 1.416 MW, yielding a 6.5% reduction in the cooling costs. This represents a saving of tens of thousands of dollars annually.

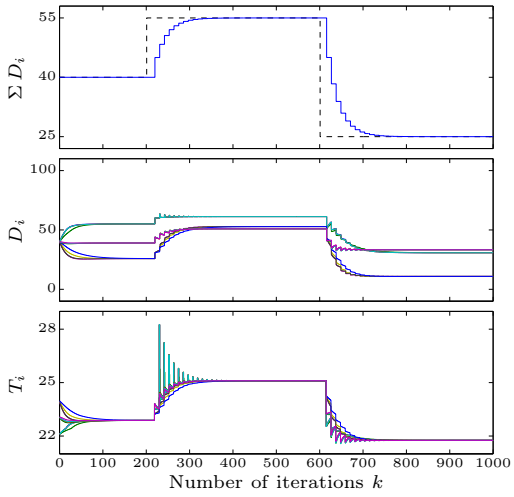


Figure 6. Performance of Algorithm 1 at three demand levels $D^* = \{40\%, 55\%, 25\%\}$, which are indicated by the dashed line in the top plot, using $\eta = 0.1$, $\mu = 1/3$.

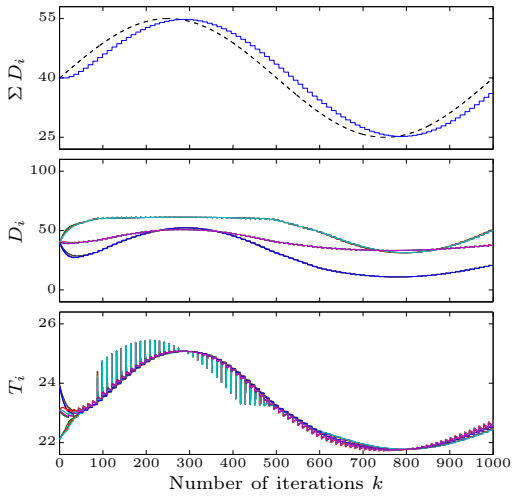


Figure 7. Performance of Algorithm 1 with demand varying in a periodic fashion, using $\eta = 0.1$, $\mu = 1/3$.

Comment. As the equilibrium state is achieved for any randomly connected graph, these cost savings are robust with respect to changes (such as link failures) in the communication topology.

Finally, Figure 7 depicts a scenario where the demand is varying in a periodic fashion; for example daily demand patterns are often assumed to be periodic. As can be seen, satisfactory tracking is achieved even though Algorithm 1 is designed for fixed point regulation only.

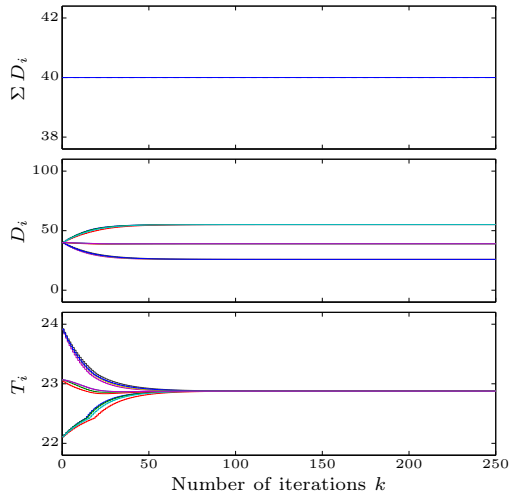


Figure 8. Performance of Algorithm 2 with constant demand of 40%, using $\eta = 0.1$.

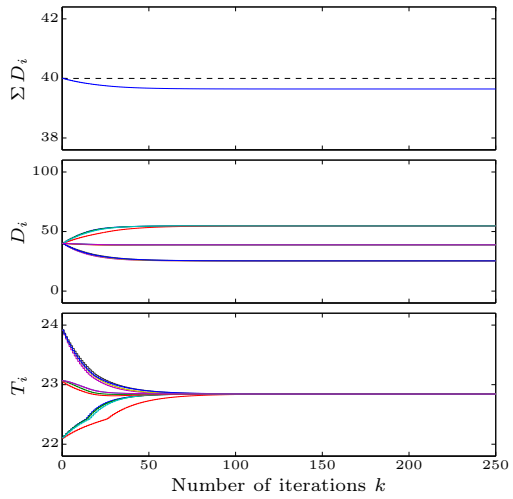


Figure 9. Performance of Algorithm 2 with a single broken link and constant demand of 40%, using $\eta = 0.1$.

3.3 Simulation 2

The setup is exactly as before. However, we now enforce the additional assumptions made in Sec. 2.3; recall in particular the assumption of symmetry in the information exchange. Figure 8 depicts Algorithm 2 equalising temperatures for a constant total load of 40%. Note, however, the effect of breaking a single link. As can be seen in Figure 9, the total demand constraint can no longer be satisfied and the site cannot satisfy the most basic quality of service requirement, namely that all requests are answered. Note that this is not surprising since the algorithm was never designed to operate in such a scenario.

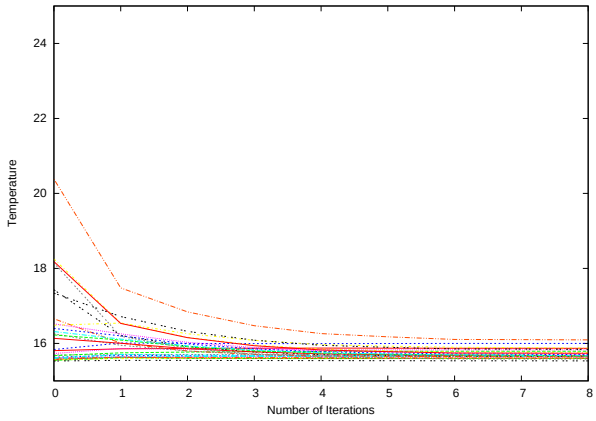


Figure 10. $\eta = 13$. The temperature of inlets of the server racks at each iteration of Algorithm 2 inside a modular data centre.

3.4 Simulation 3

A basic criticism of the discussion thus far may concern the type of data centre site considered. In many situations the assumption of containment is not valid. Namely, that T_i of machine i depends not only on D_i but also potentially on D_j ($j \neq i$) and assumption regarding the lack of heat exchange between machines is removed. Surprisingly, the algorithm works well in this case also. In such situations it is of interest to observe the performance of the algorithms presented above. To conclude this paper, we briefly apply Algorithm 2 to one such situation. Similar results can be expected for Algorithm 1 also.

Let us now consider a single machine as described above, in particular of type 3. Then assume that the server racks inside are labelled in pairs "1" to "14" as shown in Figure 4. While any connected communication graph could be used in our algorithm, we chose the following topology for $\mathcal{G}$: Let each server rack be connected to its immediately adjacent server racks and to the server rack with same label in the other cold aisle. Server racks labelled "1" and "7" are connected to server racks directly opposite them across the cold aisle (that is, "8" and "14" respectively). Clearly, this results in $\mathcal{G}$ being a connected, undirected (3-regular) graph. As can be seen from Figure 10, even in this non-ideal scenario, Algorithm 2 still manages to, more or less, equalise the temperatures across server racks. Note finally that this simulation is not MATLAB based but rather a full scale CFD simulation.

4 Conclusions

Thermal management is an important aspect in the efficient operation of data centres. The relationship between demand and temperature is a complicated one so a robust distributed algorithm is useful in the dynamic

environment of the data centre. In this paper, we have shown that distributed algorithms can be used to reduce cooling costs in certain types of data centres. A 6.5% reduction in the cooling costs which represents savings of tens of thousands of dollars annually can be achieved. Future work will explore learning automata ideas in the context of more general data centres.

Acknowledgements

The authors would like to thank Andreas Simon-Kajda and Dirk Niemeier for their help with the CFD simulations. This work is partially funded by the Irish Higher Education Authority under the HEA PRTLI Network Mathematics Grant and by SFI grant 07/IN.1/I901.

References

[1] Ali Almoli, Adam Thompson, Nikil Kapur, Jonathan Summers, Harvey Thompson, and George Hannah. Computational fluid dynamic investigation of liquid rack cooling in data centres. *Applied Energy*, 89:150–155, 2012.

[2] Dave Anderson, Jim Dykes, and Erik Riedel. More than an interface—SCSI vs. ATA. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, pages 245–257, San Francisco, April 2003.

[3] ASHRAE Technical Committee 9.9. 2011 thermal guidelines for data processing environments - expanded data center classes and usage guidance, 2011. http://tc99.ashraetcs.org/documents/ASHRAE Whitepaper - 2011 Thermal Guidelines for Data Processing Environments.pdf.

[4] L. A. Barroso and Urs Hölze. The case for energy-proportional computing. *IEEE Computer*, 40(12):33–37, 2007.

[5] L. A. Barroso and Urs Hölzle. The datacenter as a computer: An introduction to the design of warehouse-scale machines. *Synthesis Lectures on Computer Architecture*, 2009.

[6] Henry C. Coles and Mark Bramfitt. Modular/container data centers procurement guide: Optimizing for energy efficiency and quick deployment. Lawrence Berkeley National Laboratory, February 2011. Available at `http://goo.gl/ho9cC`.

[7] Mentor Graphics Corporation. Flovent version 9.1, 2010.

[8] Rajarshi Das, Jeffrey O. Kephart, Jonathan Lenchner, and Hendrik Hamann. Utility-function-driven energy-efficient cooling in data centers. In *Proceedings of the 7th International Conference on Autonomic Computing*, pages 61–70, Washington, June 2010.

[9] Ahmad Faraz and T.N. Vijaykumar. Joint optimization of idle and cooling power in data centers while maintaining response time. In *Proceedings of ASPLOS*, pages 243–256, Pittsburgh, 13 - 17 March 2010.

[10] Ali Jadbabaie, Jie Lin, and A. Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbour rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[11] Florian Knorn, Martin J. Corless, and Robert N. Shorten. Results in cooperative control and implicit consensus. *International Journal of Control*, 84(3):476–495, March 2011.

[12] J Lin, A S Morse, and B D O Anderson. The Multi-Agent Rendezvous Problem. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2, pages 1508–1513, Maui, 9 - 12 December 2003.

[13] Justin Moore, Jeffrey S. Chase, Parthasarathy Ranganathan, and Ratnesh Sharma. Making scheduling "Cool": Temperature-aware workload placement in data centers. In *Proceedings of USENIX*, pages 61–75, Anaheim, April 2005.

[14] Luc Moreau. Stability of multiagent systems with time-dependent communciation links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.

[15] Reza Olfati-Saber and Richard M. Murray. Consensus protocols for networks of dynamic agents. In *Proceedings of American Control Conference*, volume 2, pages 951–956, Denver, 4 - 6 June 2003.

[16] Luca Parolini, Niraj Tolia, Bruno Sinopoli, and Bruce H. Krogh. A cyber-physical systems approach to energy management in data centers. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, Stockholm, April 2010.

[17] Chandrakant D. Patel, Cullen E. Bash, and Ratnesh Sharma. Smart cooling of data centers. In *Proceedings of IPACK*, pages 128–137, Maui, 6-11 July 2003.

[18] Michael K. Patterson and Dave Fenwick. The state of datacenter cooling. Available at `http://goo.gl/GUlvF`.

[19] Mikko Pervilä and Jussi Kangasharju. Cold air containment. In *Proceedings of ACM SIGCOMM Workshop on Green Networking*, pages 7–12, Toronto, August 2011.

[20] Peter Rumsey. Overview of liquid cooling systems, 2007. http://hightech.lbl.gov/presentations/Dominguez/5_LiquidCooling_101807.ppt.

[21] Ratnesh K. Sharma, Cullen E. Bash, and C. D. Patel. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9(1):42–49, 2005.

[22] Rade Stanojević and Robert Shorten. Generalized distributed rate limiting. In *Proceedings of IWQoS*, pages 1–9, Charleston, July 2009.