Data centers are a software development challenge

Luca Bortot†
IT Architectures Project Leader
ENI
Italy
luca.bortot@eni.com

Walter Nardelli
Technical Services Manager
ENI
Italy
walter.nardelli@eni.com

Peter Seto
EE HPC WG
USA
jordanruthseto@aol.com

ABSTRACT

ENI's Green Data Center (GDC) was commissioned in 2008 in Pavia, IT to contain all of the energy company's IT hardware from servers up to several HPC machines. Given ENI's core competence in industrial controls, the design team, led by computer scientists, attacked the design challenge as a controls software development problem. Their software included a hardware debugger, data from extensive instrumentation stored in perpetuity, and an adaptive, dynamic, hardware control stack with experimentally-confirmed optimizing algorithms. The results are a high-availability, highly-automated, low PUE [1] and relatively low cost system, and are of interest for designers and operators of HPC facilities and energy-dense Data centers.

KEYWORDS

Resource Management

Evaluation of Hardware and Algorithms

Power-aware Energy-efficient Machine learning

1. Introduction

The poster describes a case study showing the application of Operational Data Analytics (ODA) [2] at ENI Green Data Center [3] for:

- fault detection/correction in the control software and facility hardware,
- adaptive modification of control software, and real-time optimization,
- fine tuning hardware and optimization of the control software for cost.

Cases of failures and/or near-misses, their detection by the debugger software, and active prevention methods developed and implemented at GDC are described.

The poster also provides a typology for categorizing and understanding fault detection and correction created by the authors.

2. Fault masking and detection

Because the control software acts autonomously to maintain setpoint values by controlling many separate and redundant devices, the malfunction of any one device may be difficult to detect or predict. This fault-hiding artifact, called masking, prompted the development of two detecting strategies: assigning a detectable fault co-variant, and routine stress testing of components. Examples of both are presented.

In the proposed typology hardware faults are divided into four classifications: non- trivial detected or undetected, and trivial detected or undetected. Non-trivial faults can affect the system's nominal function. Trivial faults cannot affect the nominal function of the system under any circumstances, but can affect efficiency or cost.

The ODA design should detect all non-trivial faults -despite masking effects of the control software- and detect the trivial faults based on a rank ordering of cost-effectiveness.

3. System optimization

The Design Team created an adaptive control software stack, including dynamic adjustment of setpoints and a variable menu of components working in various configurations to optimize energy and facility use. ODA allowed for historical and real-time observation of results during testing. ODA was also applied to improve PUE, as faults which wasted power or generated added cooling load were detected and eliminated.

3.1. Dynamic setpoint adjustment

The control system starting parameters assumed a static, worst-case scenario. This produced large trapped and stranded capacity.

To detect, quantify, and reduce this burden, setpoint ranges were broadened experimentally to test system response; for example, air plenum temperature was varied from 16C to 28C, and code was written to allow this autonomously in response to environmental and load conditions. This also allows a predictive function that anticipates ambient conditions based on weather forecasts and workload history, in order to change setpoints to take advantage of weather changes before they happen.

3.2. Dynamic hardware selection

Cooling components (eg: air dampers, chillers, evaporative towers) are combined dynamically by the control software. The process involves evaluation of many parameters, including, for instance: net cost, environmental conditions, maximum rate of change of a variable, or reducing wear-and-tear on an aging component. Every device is also pushed to its experimentally-determined limit- often beyond data-plate values.

4. Reducing PUE and cost

The control software allowed rank-ordering alternative methods of setpoint attainment for cost, and the experimental testing of manufacturer's data-plate ratings, both resulting in significant cost and energy savings.

4.1. Data plate testing

With fine-grained data, component data-plate values and setpoints supplied by manufactures were tested. Allowing chiller condenser intake temperature to float 5C above ambient, rather than using the manufacturer's setpoint, the COP of the chillers rose from 11 to 22 in the conditions described below.

4.2. Rank ordering for cost

To operate the facility as cost-effectively as possible, the ODA system identified, and selected among, alternative means to attain setpoints, ranked by costs. To attain nominal inlet air temperature and humidity during low WB temperature winter conditions, the software can select chillers to cool recirculating air of the correct humidity, rather than humidifying a large volume of dry ambient air. Three variables are involved in the choice matrix: net costs of chiller cooling, net cost of humidification, and ambient air conditions. (ODA historical data showed that adding an additional humidification system would not be cost-effective.)

5. Conclusions

The work at ENI GDC can be viewed as both an ODA application that is practical and cost effective for mid-to-small HPC centers, and one with value for Data Centers that are growing into the power densities of 50Kw/rack of the GDC design. ODA has proven critical to achieve control automation, automated fault detection, and energy optimization in a high-availability data center, and has been scaled to the HPC level. Cost considerations, both variable and fixed are directly addressed by this ODA application. Use of the archived data, and lessons learned by computer scientists and power engineers will advance the knowledge base applied to future facility and software design. This work is also the necessary precursor to an AI application.

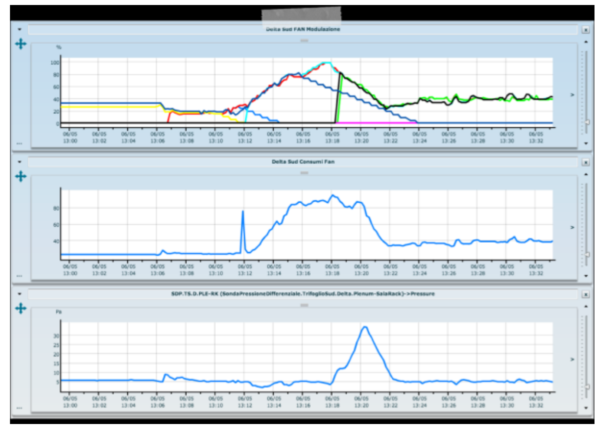

Figure 1: Example of non-trivial, masked fault and detectable co-variant

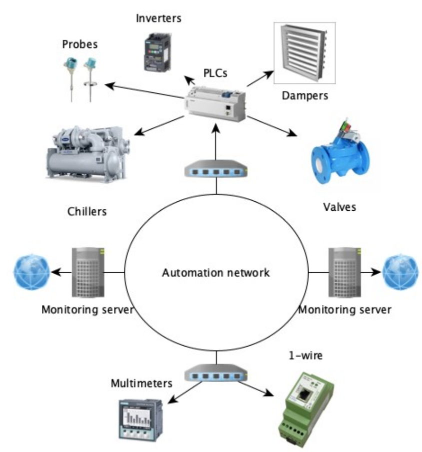

Figure 2: The data collection is performed in-band on the automation network (Ethernet). Most of field devices are accessed through the PLCs. Communication protocols include BACnet, SNMP, Modbus, HTTP, ICMP. Average polling is 10s.

REFERENCES

[1] White Paper #49: Avelar,V et al, PUE: A Comprehensive Examination of the Metric, 2012 https://datacenters.lbl.gov/sites/all/files/WP49-PUE%20A%20Comprehensive%20Examination%20of%20the%20Metric_v6.pdf

[2] Bourrassa, N. et al, Operational Data Analytics: Optimizing the National Energy Research Scientific Computing Center Cooling Systems. ICPP Proceedings EE HPC SOP Workshop, 2019

[3] Bortot, L.., ENI Green Data Centre. Unpublished Manuscript,9th European Infrastructure Worlshop , 2018