ZettaScaler: Liquid immersion cooling Manycore based Supercomputer

Sunao Torii
ExaScaler Inc.
2-1 Kanda-Ogawamachi, Chiyoda-ku, Tokyo, Japan
torii@exascaler.co.jp

Hitoshi Ishikawa
PEZY Computing K.K.
1-11 Kanda-Ogawamachi Chiyoda-ku, Tokyo, Japan
ishikawa@pezy.co.jp

***Abstract*— We have been developing the power-efficient supercomputer ZettaScaler series. In ZettaScaler-1, the first generation, we adopted three novel technologies, MIMD ultra-manycore processor called "PEZY-SC", high density server board "Brick", and direct liquid immersion cooling system. They enable us to realize high performance, low power consumption, and space miniaturization at a same time. In this article, we explain the novel hardware architectures and programming model of ZettaScaler-1. In addition, we mention the features of the next-generation ZettaScaler-2 which is currently being constructed. In the ZettaScaler-2, we have newly been developing the second generation ultra-manycore processor called "PEZY-SC2", 3-dimensional mounting technology between processor and DRAMs by magnetic coupling TCI (Thru-Chip Interface) and new high-density Brick structure.**

Keywords—supercomputer; manycore processor; liquid immersion cooling system; magnetic coupling inter-chip communication

I. INTRODUCTION

Recently, in terms of advancing the performance of supercomputers, one of the most difficult challenge is reducing power consumption. Supercomputer developers have been making efforts to improve power efficiency and cool the heat generated by huge power consumption.

To solve these issues, we have been developing the original manycore processor and a supercomputer ZettaScaler series using its own direct immersion cooling system. In the supercomputer energy saving ranking Green 500 list, one of the ZettaScaler system "Shoubu" was certificated No. 1 place among 3rd consecutive terms from June 2015 to June 2016.

In this article, we explain the architecture and programming of this ZettaScaler. Furthermore, we will also refer to the innovative technologies introduced in the next-generation ZettaScaler - 2.

II. ZETTASCALER-1 ARCHITECTURE

ZettaScaler-1 is a generic term for supercomputer systems with PEZY-SC manycore processor as a key component. It is also combined with direct liquid immersion cooling system using fluorocarbon as a coolant and high-density server board implementation technique called "Brick". Through these technologies, high performance and high power efficient machines are realized with a small footprint. Each element technology will be explained below.

A. Ultra-manycore Proessor PEZY-SC

PEZY-SC is a manycore calculation specialized processor chip which integrates 1,024 PE (processing element) on a chip. The block diagram is shown in Fig.1 and specification is indicated in TABLE 1.

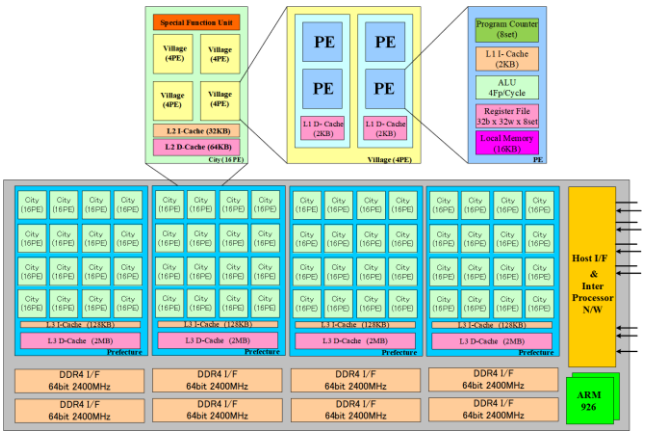

Fig. 1. The block diagram of PEZY-SC processor

TABLE I. PEZY-SC SPEFICICATIONS

Process	TSMC 28HPM
Frequency	733MHz
Cache Memory (Chip Total)	L1:1MB(D), 2MB(I) L2: 4MB(D), 2MB(I) L3: 8MB(D), 512KB(I)
Local Memory	Total 16MB (16KB/PE)
PE	1,024core 2issue/cycle 8way time-sliced fine-grain multi-threading (Total: 8,192threads)
Integrated IP	ARM926 x 2 PCIe Gen3 4ch 8Lane DDR4 64b 1,333MHz 8ch
Peak Performance	3.0TFLOPS(SP), 1.5TFLOPS(DP)
Power	70W(Typical), 100W(Peak)
Package	47.5 x 47.5mm (2,112pin)

PEZY-SC is designed with considering how to realize higher calculating performance, higher power efficiency and rich programmability at a same time.

The design features of PEZY-SC are follows,

- MIMD architecture:
  Improve flexibility, programmability and adaptability
- Fine-grain multi-threading and Hierarchical cache:
  Hiding and reducing memory latency and increasing system scalability
- 2way-superscaler:
  In-order instruction issue, out-of-order completion, Boosting IPC (Instruction per Cycle)
- Special Function Unit (SFU):
  Consolidation complex instruction calculation unit such as Divide and Square-Root calculation, Simplify ALU of each PE
- Original Instruction Set Architecture:
  Refine carefully selected instructions including hierarchical synchronizations

B. *Ultra high-density ZettaScaler-1 Birck server*

We also developed Brick server to utilize liquid immersion cooling capability. The picture of ZettaScaler-1.4 brick server is shown in Fig. 2.

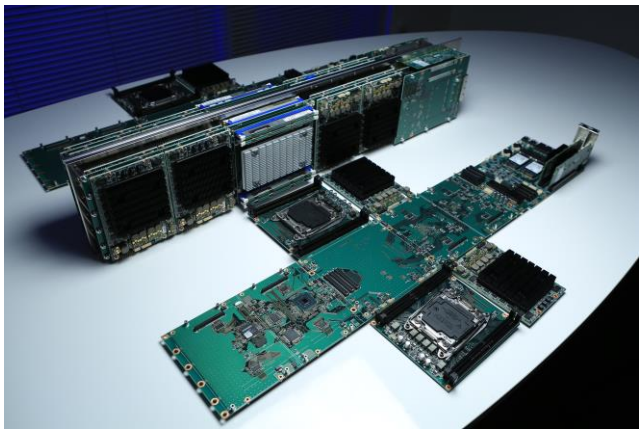

Fig. 2. ZettaScaler-1.4 Brick server

The Brick server has following features:

- High density: around 3-times higher than air-cooling server
- Modular design: easy replacing and versioning up of each part
- Screw power supply: eliminate power cable of among boards
- Simplify: focusing on reducing power consumption

The block diagram of ZettaScaler-1.4 is shown in Fig.3. One Xeon E5 CPU is connected to four PEZY-SC processors as a calculating accelerator, which is similar to GPGPU system.

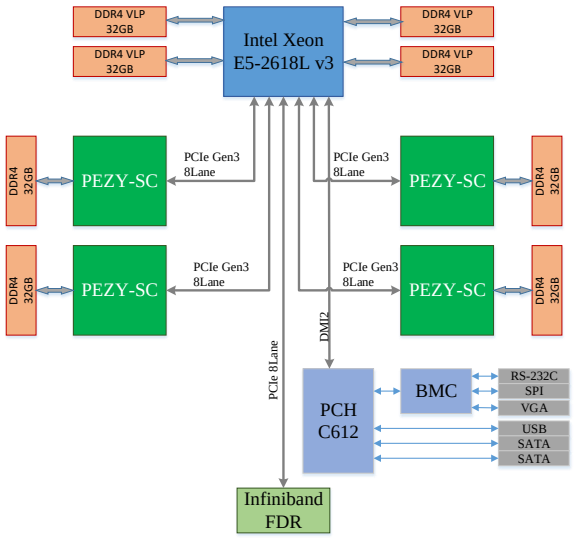

Fig. 3. ZettaScaler-1.4 Block Diagram

C. *Direct Liquid immersion cooling*

We developed direct liquid immersion cooling system to improve cooling capability and density. Whole of the ZettaScaler system boards are soaked in Fluorocarbon-based fluid coolant. Fig.4 shows ZettaScaler cooling methodology. We selected heat-conductor cooling and forced circulation, so heated coolant is cooled down at the heat exchanger, and heat is cooled by heat pump chilling unit located at out-side facilities.

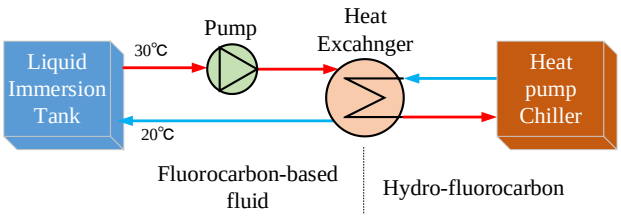

Fig. 4. ZettaScaler liquid immersion cooling methodology

This cooling system has following several advantages:

- Heat conductor cooling: neither sealing nor vacuuming are required, easy maintainability
- Fluorocarbon-based fluid coolant: fewer coolant loss, no side-effect to the system, reducing leakage power and extended semiconductor lifespan due to lower operation temperature

III. PROGRAMMING IN ZETTASCALER-1

A. *PEZY-SC programming model*

An OpenCL-like environment named PZCL is provided, on which users write both the CPU code and the PEZY-SC kernel code. The CPU code sets up the kernel code and buffers, write the initial value to the buffers allocated in the PEZY-SC, and then invoke the kernel code on the PEZY-SC as multiple threads.

Since the PEZY-SC is a MIMD processor, these threads can run independently. PEZY-SC can instantiate up to 8,192 threads all at once, and more than 8,192 threads are dispatched in multiple batches.

After detecting the end of all threads in the PEZY-SC, the CPU code reads the results from PEZY-SC buffers to the CPU buffers. Fig.5 illustrates the typical protocol between the CPU and PE.

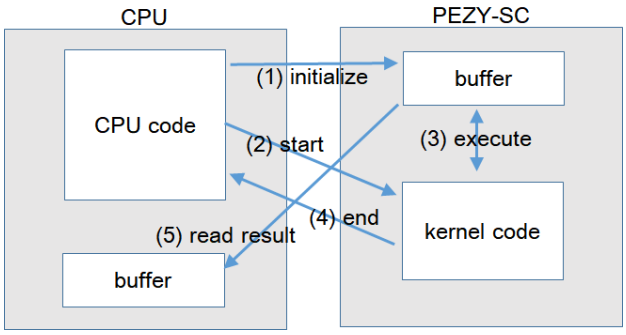

Fig. 5. The typical protocol between CPU and PEZY-SC

It is important, in programming the PEZY-SC code, to be aware of the hardware configurations and adjust the algorithms to them. For the purpose, users can use the following built-in functions in the PEZY-SC kernel code.

- get_pid()/get_tid()
  get_pid() returns the computation core id, and get_tid() returns the thread id in the core. While the physical number of computation cores in PEZY-SC is 1,024, get_pid() returns a logical number which may exceed this limitation.
- get_maxpid()/get_maxtid()
  get_maxpid() returns the number of computation cores, and get_maxtid() returns the number of threads in a core (i.e. 8). As in the case of get_pid(), the return value of get_maxpid() is not limited to 1,024.
- sync()/sync_L1()/sync_L2()/sync_L3()
  All threads in a certain level synchronize when this function is called. L1/L2/L3 specify the level of synchronization which corresponds to the cache level. The threads sharing the Lx cache synchronize at the point sync_Lx() appears. The maximum level synchronization including all the threads in a PEZY-SC processor is done with sync().
- flush()/flush_L1()/flush_L2()/flush_L3()
  All threads in a certain level synchronize and flush the cache data. Same as sync(), L1/L2/L3 means the cache level and flush_Lx() requests the Lx cache to write data to the next level cache. By flush(), all the cached data is requested to be written to the global memory.
- chgthread()
  As shown in the left of Fig.6, chgthread () swaps the front (active) and back (inactive) threads. This function is used to conceal the latency of the global memory access and/or the complicated calculation by the Special Function Unit (SFU).

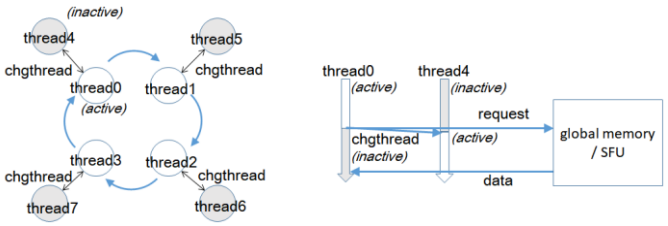

Fig. 6. The typical protocol between CPU and PEZY-SC

B. *Simple example of the kernel code*

Following is a sample kernel code which calculates the element-wise addition from the array-a and array-b to the array-c.

```
1  void pzc_Add(float* a, float* b, float* c, int count)
2  {
3      int tid = get_tid();
4      int pid = get_pid();
5
6      for (int pos = pid*get_maxtid() + tid;
7            pos < count;
8            pos += get_maxtid()*get_maxpid()) {
9          sync_L2();
10         float a_ = a[pos];
11         float b_ = b[pos];
12         chgthread();
13         c[pos] = a_ + b_;
14     }
15     flush();
16 }
```

On line 1, the pointers to the arrays are given to the function with their size. On line 6-8, the target element id is calculated to the variable pos. Line 10-11 issue the global memory read requests. Line 13 receives the two values a_ and b_ and sums up them to the array-c. Line 15 flushes the cache to the global memory.

Although both of line 9 and line 12 are not necessarily required when we only concern with its functionality, they often give the merits in the performance.

The sync_L2() on line 9 is effective for the coalesced global memory access. Since the synchronization function costs a certain amount of overhead, the level should be decided carefully depending on the case.

The chgthread() on line 12 has a role to conceal the global memory access latency issued on line 10 and 11.

C. *Effective use of MIMD architecture*

As we mentioned in previous paragraph, the synchronization functions are frequently effective in the coalesced global memory access. Although the PEZY-SC is a MIMD processor, accessing the global memory randomly by many threads may cause performance degradation.

On the other hand, when there are not so much global memory accesses, each thread in the PEZY-SC can branch independently with few performance penalties.

The following shows a typical kernel code flow using the local memory and MIMD architecture.

1. read the input data from the global memory with coalesced memory access.
2. calculate with local memory in a MIMD fashion.
3. write the result to the global memory with coalesced memory access.

If a job in line 2 can be divided in a task task-parallel way, we can make the most of the PEZY-SC MIMD feature.

IV. Introducing Next-Generation, ZettaScaler-2

Now we are developing the next-generation supercomputer system "ZettaScaler-2" to reflect the findings of ZettaScaler-1. ZettaScaler-2 adopts "PEZY-SC2" processor, 3-dimensional mounting by magnetic coupling TCI (Thru-chip Interface) memory, and higher density new Brick boards. The features of ZettaScaler-2 are described in following paragraphs.

A. PEZY-SC2 manycore processor

PEZY-SC2 is a second generation ultra-manycore processor chip which is fabricated by TSMC 16nm FinFET+ process. TABLE II indicates the specification of PEZY-SC2. It allocates the benefits of microfabrication of a process to increase the number of PEs. In PEZY-SC, two ARM926 processors are implemented, but those specification and capability are not enough to operate 64bit memory address space and host processing. In contrast, PEZY-SC2 integrates MIPS64 P6600 6-core processors as host-CPUs which provide unified address space with PEs. They enable us to reduce data transfer overhead.

TABLE II. PEZY-SC2 Specifications

Process	TSMC 16FinFET+
Frequency	1GHz
Cache Memory (Chip Total)	L1:4MB(D), 8MB(I) L2: 8MB(D), 4MB(I) LLC: 40MB
Local Memory	Total 40MB (20KB/PE)
PE	2,048core 2issue/cycle 8way time-sliced fine-grain multi-threading (Total: 16,384threads)
Integrated IP	MIPS64R6(P6600) 6core PCIe Gen4 8Lane x 4Port Custom TCI Stacked DRAM 4Port 2TB/s DDR4 3,200MHz 4Port 100GB/s
Peak Performance	8.2TFLOPS(SP), 4.1TFLOPS(DP), 16.4TFLOPS(HP)
Power	180W (Peak Estimated)
Package	55x55mm 2,338pin (Non-TCI) 95x55mm 3,040pin (TCI)

Furthermore, PEZY-SC2 has 16 x 40MB Last Level Caches. Because they are located beside each memory controller and allocated unique physical memory address space, which shared between MIPS cores and PEs. Therefore, it is not necessary for programmer to pay attention for cache consistencies among LLCs level. PEZY-SC2 also supports half-precision floating calculation to boost performance of Deep Learning applications.

B. PEZY-SC2 New Features from programming side

In the PEZY-SC2, new distinctive features, which bring a change on the programming style, are also provided.

Automatic chgthread mode: PEZY-SC users need to write chgthread() at the point they want to reduce the latency. Yet PEZY-SC2 users allow to write it by themselves, but with this new mode they need not do it. In this mode, the front and back threads change repeatedly without chgthread(). So PEZY-SC2 users can select whether to validate or invalidate the mode in the kernel code.

Intra-core synchronization mode: This mode forces the pipeline stages in a core to keep pace. This feature looks like the SIMD architecture but is different in that each thread can execute its own instruction. Like the case of the automatic chgthread mode, whether to validate or invalidate this function can be selected in the kernel code. This feature gives an advantage in the power efficiency in a calculation. In the programming aspect of view, we have another characteristic. E.g., we can write a kernel code to exchange the data between the threads using the local memory without any synchronization function.

Atomic operations between CPU and PEs: Using these operations, the CPU code can communicate asynchronously with the kernel code. This is also useful for debugging since the CPU can look into the kernel states without terminating it.

C. Expanding memory bandwidth using magnetic coupling TCI technologies

To enlarge memory-bandwidth and to match the various random-access demands from the huge number of MIMD PEs, we have been developing novel 3-dimmension magnetic coupling TCI technology. Fig. 7 shows the picture of PEZY-SC2 TCI package.

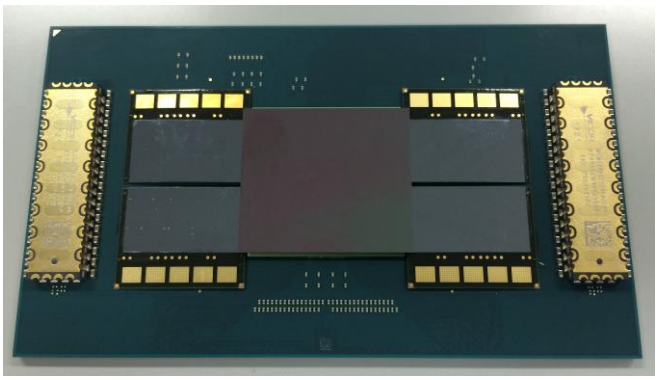

Fig. 7. PEZY-SC2 TCI package

PEZY-SC2 is connected between four TCI-DRAMs using TCI technology and each TCI-DRAM has 0.5TB/sec memory bandwidth and 32-channel 4-bank structure. Since the peak floating-point performance of PEZY-SC2 reaches 4.1TFLOPS, then the memory bandwidth ratio achieves around 0.5B/FLOP.

It means PEZY-SC2 is one of the most memory intensive accelerator than any other processors. This enables us to extend application coverage area of PEZY-SC2.

D. High Density Brick Server

The Brick structure of ZettaScaler-2 is drastically re-designed with the aim of improving the degree of integration flexibility and maintainability, and further reducing the power consumption.

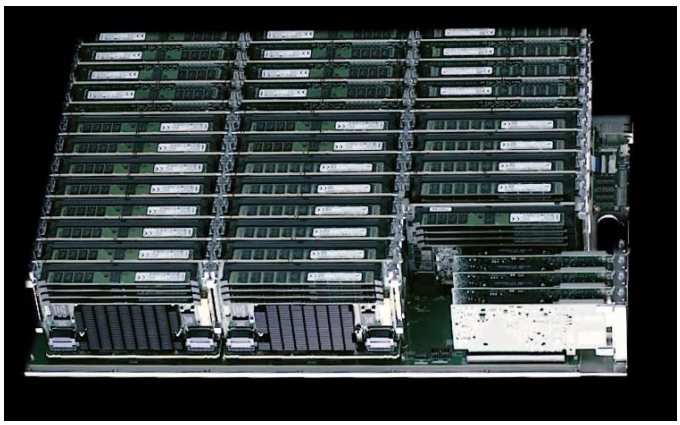

Fig. 8. ZettaScaler-2 PEZY-SC2 Brick

Fig. 8 shows the picture of the ZettaScaler-2 PEZY-SC2 brick. The brick consists of 32 PEZY-SC2 module cards, one Dual-XeonD module card, four Infiniband network interface cards and the base/sub carrier-boards. Each module card is connected via programable PCIe fabric switches which realize flexible programmable network structure. The density of ZettaScaler-2 brick achieves about twice from previous one. Moreover, we also develop not only PEZY-SC2 module card but also XeonD card, Xeon E5 card, FPGA card, SSD card and GPGPU card of ZettaScaler-2 to fit various computation demands.

ZettaScaler-2 also adopts DC48V higher voltage power supply line to reduce power transmission loss through the board. The Embedded Power Units (EPU) are located the bottom of liquid cooling tank and they convert from 3-phase AC200V to DC48V directly. It reduces power conversion overhead and downsizes of the power supply unit.

V. Future Prospective toward to ZettaScaler-3

We are planning to continuously develop new processor, which increases the number of cores per chip as semiconductor processes evolve. At present, we are starting to develop PEZY-SC3, which plans to advance the process up to 7 nm generation and to implement 4,096 to 8,192 cores per chip. In addition, although improvement was attempted with respect to the main storage band at PEZY-SC2 regarding the network bandwidth, it remained as before. PEZY-SC3 is exploring the possibility of improving by introducing a new optical interface. As early as 2018, we would like to follow the deployment of the ZettaScaler-3 series using PEZY-SC3.

VI. CONCLUSION

In this article, we introduced PEZY-SC series ultra-manycore processor architecture, ZettaScaler server features, implementation techniques, and software development environment, as well as liquid immersion cooling methodologies. Since the developing resources are limited, there are many insufficient completions, but if there is an opportunity to use the original supercomputer we created, it would be greatly appreciated.

Acknowledgment

The development of ZettaScaler supercomputer series is supported by the dedication efforts of many employees and cooperating companies. We'd like to express our deep appreciation to you here.

References

[1] S.Torii, ”ExaScaler-1: The Power-Efficient Submersion Manycore Processor Based Supercomputer”, COOL chips -XVIII, 2015.

[2] N.Nakasato, “Performance evaluation of scientific applications on Suiren System (in Japanese)", JSICR HPC-152 (11) , pp.1 - 7 ,Dec, 2015.

[3] T.Mitsuishi, T.Kaneda, H.Amano and S.Torii, “Breadth-first Search on Suiren: a compact supercomputer”, Proc. of International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies (HEART), 2016.

[4] T.Aoyama, K.Ishikawa, Y.Kimura, H.Matsufuru, A.Sato, T.Suzuki, and S.Torii, “First application of lattice QCD to Pezy-SC processor”, Proc. of The International Conference on Computational Science (ICCS) , 2016.

[5] T.Yamazaki, J.Igarashi, J.Makino, T.Ebisuzaki,”Realtime simulation of a cat-scale artificial cerebellum on PEZY-SC processors.”, International Journal of High Performance Computing Applications, 6 June 2017, https://doi.org/10.1177/1094342017710705

[6] Y.Haribara, H.Ishikawa, S.Utsunomiya, K.Aihara, Y.Yamamoto, “Performance evaluation of coherent Ising machines against classical neural networks”, 14 Aug. 2017, Quantum Sci. Technol. 2 044002, https://doi.org/10.1088/2058-9565/aa8190.

[7] A.Tabuchi, Y.Kimura, S.Torii, H.Matsufuru, T.Ishikawa, T.Boku, M.Sato: “Design and Preliminary Evaluation of Omni OpenACC Compiler for Massive MIMD Processor PEZY-SC”, OpenMP: Memory, Devices, and Tasks: 12th International Workshop on OpenMP, IWOMP 2016, pp. 293-305, Oct. 2016.

[8] "Evolving semiconductor process, cooling system and interconnection toward an Exa-Scale high-performance computation : Part1 (in Japanese)", Nikkei Electronics, July 2015, pp.99-105, 2015.

[9] "Evolving semiconductor process, cooling system and interconnection toward an Exa-Scale high-performance computation : Part2 (in Japanese)", Nikkei Electronics, Sept. 2015, pp.69-75,2015.

[10] P.Kongetira, K.Aingaran, K.Olukotun, “Niagara: a 32-way multithreaded Sparc processor”, IEEE MICRO Vol.25 Issue2, 2005.

[11] “Oil Submersion Cooling for Today's Data Centers” , Green Revolution Cooling white paper, http://www.grcooling.com/wp-content/uploads/2015/06/GRC_WP-CLICK-Oil_Sub_DCc.pdf

[12] “Two-Phase Immersion Cooling -A revolution in data center efficiency”, 3M technical report, http://multimedia.3m.com/mws/media/1127920O/2-phase-immersion-coolinga-revolution-in-data-center-efficiency.pdf

[13] Oil Submersion Cooling for Today's Data Centers”, Green Revolution Cooling white paper, http://www.grcooling.com/wp-content/uploads/2015/06/GRC_WP-CLICK-Oil_Sub_DCc.pdf