PicoServer: Using 3D Stacking Technology To Enable A Compact Energy Efficient Chip Multiprocessor

Taeho Kgil Shaun D'Souza Ali Saidi Nathan Binkert [*] Ronald Dreslinski Steven Reinhardt [†]
Krisztian Flautner[‡] Trevor Mudge

Advanced Computer Architecture Laboratory
Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor
{tkgil,sdsouza,saidi,binkertn,rdreslin,stever,tnm}@eecs.umich.edu

[‡]ARM Ltd
110 Fullbourn Road,
Cambridge, United Kingdom
krisztian.flautner@arm.com

Abstract

In this paper, we show how 3D stacking technology can be used to implement a simple, low-power, high-performance chip multiprocessor suitable for throughput processing. Our proposed architecture, PicoServer, employs 3D technology to bond one die containing several simple slow processing cores to multiple DRAM dies sufficient for a primary memory. The 3D technology also enables wide low-latency buses between processors and memory. These remove the need for an L2 cache allowing its area to be re-allocated to additional simple cores. The additional cores allow the clock frequency to be lowered without impairing throughput. Lower clock frequency in turn reduces power and means that thermal constraints, a concern with 3D stacking, are easily satisfied.

The PicoServer architecture specifically targets Tier 1 server applications, which exhibit a high degree of thread level parallelism. An architecture targeted to efficient throughput is ideal for this application domain. We find for a similar logic die area, a 12 CPU system with 3D stacking and no L2 cache outperforms an 8 CPU system with a large on-chip L2 cache by about 14% while consuming 55% less power. In addition, we show that a PicoServer performs comparably to a Pentium 4-like class machine while consuming only about 1/10 of the power, even when conservative assumptions are made about the power consumption of the PicoServer.

Categories and Subject Descriptors C.1 [*Processor Architectures*]: Parallel Architectures; C.5 [*Computer System Implementation*]: Servers

General Terms Performance, Design, Experimentation

Keywords Low power, Tier 1 server, Web/File/Streaming server, 3D stacking technology, chip multiprocessor, full-system simulation

1. Introduction

3D stacking technology enables new chip multiprocessor (CMP) architectures that significantly improve energy efficiency. Our proposed architecture, PicoServer, employs 3D technology to bond one die containing several simple slow processor cores to multiple DRAM dies that form the primary memory. In addition, 3D stacking enables a memory processor interconnect that is both very high bandwidth and low latency. As a result the need for complex cache hierarchies is reduced. We show that the die area normally spent on a L2 is better spent on additional processor cores. For example, in our experiments we show that an L2 cache can be replaced by 4 extra cores. The additional cores means that they can be run slower without affecting throughput. Slower cores also allow us to reduce power dissipation and with it thermal constraints, a potential roadblock to 3D stacking. The resulting system is ideally suited to throughput applications such as Tier 1 web servers.

Internet service providers like *AOL*, *Yahoo* and *Google* require large numbers of web servers to satisfy customer needs. Server farms based on off-the-shelf general purpose processors are unnecessarily power hungry, require expensive cooling systems and occupy a large space. It has been shown that 25% of the operating costs for these "server farms" can be directly or indirectly attributed to power consumption [36]. This figure has the potential to grow rapidly along with the continuing growth in web services. Figure 1 shows a 3 Tier server farm and how it might handle a request for service. The first tier handles the bulk of the requests. Employing PicoServers at this level can significantly lower power consumption and space requirements.

Tier 1 server applications handle events on a per-client basis, which are independent and display high levels of thread level parallelism. This high level of parallelism makes them ill-suited for traditional monolithic processors. CMPs built from multiple simple cores can take advantage of this thread level parallelism to run at a much lower frequency while maintaining a similar level of throughput and thus dissipating less power. By combining them with 3D stacking we will show that it is possible to cut power requirements further. 3D stacking enables the following key improvements:

- **High bandwidth buses between main memory and L1 caches that support multiple cores—1000's of low latency connections with marginal area overhead between dies are possible.** Since the interconnect buses are on chip, we are able to implement wide buses with a relatively lower power budget compared to inter-chip implementations.
- **Modification in the memory hierarchy due to the integra-**

[*] Currently at Hewlett-Packard Labs.

[†] Also with Reservoir Labs, Inc.

ASPLOS'06 October 21–25, 2006, San Jose, California, USA.
Copyright © 2006 ACM 1-59593-451-0/06/0010...$5.00.

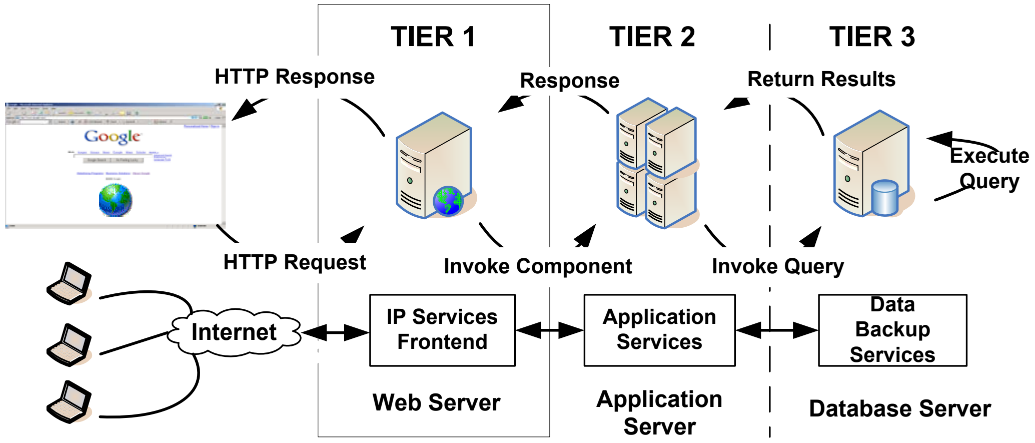


Figure 1. A Typical 3 Tier Server Architecture. Tier 1—Web Server, Tier 2—Application Server, Tier 3—Database Server. PicoServer is targeted at Tier 1. An example of an internet transaction is shown. When a client request comes in for a Java Servlet Page, it is first received by the front end server—Tier 1. Tier 1 recognizes a Java Servlet Page that must be handled and initiates a request to Tier 2 typically using Remote Message Interfaces (RMI). Tier 2 initiates a database query on the Tier 3 servers, which in turn generate the results and send the relevant information up the chain all the way to Tier 1. Finally, Tier 1 sends the generated content to the client.

tion of large capacity on-chip DRAM. It is possible to remove the L2 cache and replace it with more processing cores. The access latency for the on-chip DRAM is also reduced because address multiplexing and off-chip IO pad drivers [35] are not required.

- **Overall reduction in system power primarily due to the reduction in core clock frequency.** The benefits of 3D stacking stated in items 1 and 2 allow us to integrate more cores clocked at a modest frequency—in our work 500-1000MHz—on chip while providing high throughput. Reduced core clock frequency allows their architecture to be simplified; for example, by using shorter pipelines with reduced forwarding logic.

The potential drawback of 3D stacking, now that the technology has been shown to be feasible, is thermal containment. However, this is not a limitation for the type of simple, low power, cores that we are proposing for the PicoServer, as we show in section 4.4. In fact the ITRS projections of Table 1 predicts that systems consuming just a few watts do not even require a heat sink.

The general architecture of a PicoServer is shown in Figure 2. We have been conservative in what we assume about stacking depth. For the purposes of this study we assume a stack of 5 dies. The ITRS roadmap in Table 1 predicts much deeper stacks being practical in the near future. The connections are by vias that run perpendicular to the dies. The dimensions for a 3D interconnect via varies from $1 \sim 3\mu m$ with a separation of $1 \sim 6\mu m$. Current commercial offerings can support 1,000,000 vias per cm^2 [23]. This is far more than we need for PicoServer. These function as interconnect and thermal pipes. For our studies, we assume that the logic-based components—the microprocessor cores, the network interface controllers (NICs), and peripherals—are on the bottom layer and conventional capacity-oriented DRAMs occupy the remaining 4 layers. To understand the design space and potential benefits of this new technology, we explored the trade-offs of different bus widths, number of cores, frequencies, and the memory hierarchy in our simulations. We found bus widths of 1024 bits with a latency of 2 clock cycles at 250 MHz to be reasonable in our architecture. In addition, we aim for a reasonable area budget constraining the die size area to be below 80mm^2 at 90nm process technology. Our 12 core PicoServer configuration which occupies the largest die area is conservatively estimated to be approximately 80mm^2.The die areas for our 4, 8 core PicoServer configurations are respectively 40mm^2 and 60mm^2.

The paper is organized as follows. In the next section we provide background for this work by describing a brief overview of 3D stacking technology, the characteristics of the target workload, and previous work. In section 3, we outline our methodology for the design space exploration. In section 4, we provide more details for the PicoServer architecture and evaluate various PicoServer configurations. In section 5, we present our results in the PicoServer architecture for Tier 1 server benchmarks and compare our results to conventional architectures that do not employ 3D stacking. These architectures are CMPs without 3D stacking and conventional high performance desktop architectures which we call OO4 with Pentium 4-like characteristics. A summary and concluding remarks are given in section 6.

2. Background

2.1 3D stacking technology

In the past there have been numerous efforts in academia and industry to implement 3D stacking technology[16][30][27][32][42]. They have met with mix success. This is due to the many potential challenges that need to be addressed in 3D stacking technology: 1) Achieving high yield in bonding die stacks, 2) delivering power to each stack and 3) managing thermal hotspots due to the silicon dioxide 3D interface. However, the large scale and competition typical of mobile untethered systems have re-initiated a demand for small form factors with very low power. In response, several commercial enterprizes have begun offering reliable low-cost die-to-die 3D stacking technologies.

In 3D stacking technology, dies are typically bonded as face to face or face to back. Face to face bonds provide higher die to die via density and lower area overhead than face to back bonds. The

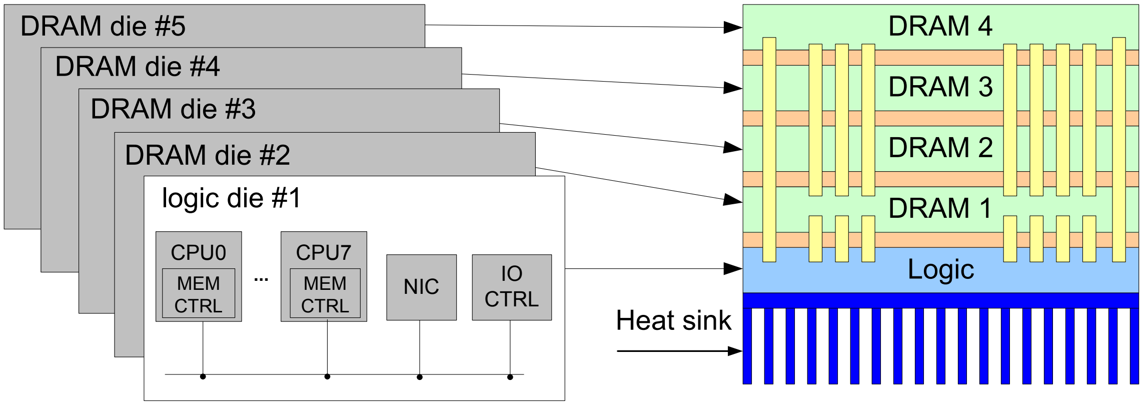


Figure 2. A diagram depicting the PicoServer: a CMP architecture connected to a conventional DRAM using 3D stacking technology with an on-chip NIC to provide low-latency high-bandwidth networking.

	2005	2007	2009	2011	2013
Low-cost/handheld #die/stack	6	7	9	11	13
SRAM density Mbits/cm^2	84	138	225	365	589
DRAM density Mbits/cm^2 at production	1,220	1,940	3,660	5,820	9,230
Max. Power Budget for cost-performance systems(W)	91	104	116	119	137
Max. Power Budget for low-cost/handheld systems with battery(W)	2.8	3.0	3.0	3.0	3.0

Table 1. ITRS projection [11] for 3D stacking technology, memory array cells and maximum power budget for power aware platforms. Based on this information, we assume a 3D stack of 5 layers can be reasonably implemented. ITRS projections suggest DRAM density exceeds SRAM density by $15 \sim 18\times$ entailing large capacity of DRAM can be integrated on-chip using 3D stacking technology as compared to SRAM.

	Face-to-Back	Face-to-Face	RPI	MIT 3D FPGA
Size	$1.2\mu \times 1.2\mu$	$1.7\mu \times 1.7\mu$	$2\mu \times 2\mu$	$1\mu \times 1\mu$
Minimum Pitch	$< 4\mu$	2.4μ	N/A	N/A
Feed Through Capacitance	$2 \sim 3$fF	≈ 0	N/A	2.7fF
Series Resistance	$< 0.35\Omega$	≈ 0	≈ 0	≈ 0

Table 2. 3D stacking technology parameters[23][12][32]

lower via density for face to back bonds attribute to through silicon vias—TSVs that punch through silicon bulk. Using the bonding techniques in 3D stacking technology, we can achieve a synergistic effect of stacking heterogeneous dies together. For example, architectures that stack conventional DRAM and logic manufactured from different process steps. Furthermore, with the added third dimension from the vertical axis, the overall wire interconnect length can be reduced and wider bus width can be achieved at lower area costs. The parasitic capacitance and resistance for 3D vias are negligible compared to global interconnect. We also note that the dimensions and pitches of 3D vias add a modest area overhead. 3D via pitches are equivalent to 22λ for $90nm$ technology. They are also expected to reduce as this technology becomes mature. Our PicoServer architecture uses face to face bonds for the logic die to DRAM die and face to back bonds for DRAM to DRAM die stacking.

2.2 Tier 1 server workload characteristics

It is well-known that Tier 1 server workloads like Web, Video streaming, NFS server applications display a high degree of thread-level parallelism since connection level parallelism through client connections can be easily translated into thread level parallelism (TLP). Conventional general-purpose processors, however, are typically optimized to exploit instruction level parallelism (ILP). Web and NFS workloads, with plenty of TLP, suffer from a high cache miss rate regularly stalling the machine. This leads to a low IPC and poor utilization of processor resources. Our studies have shown that except for computation intensive Video streaming servers, out-of-order processors have an IPC of between $0.21 \sim 0.54$ for typical Tier 1 server workloads requiring modest computation even with a large L2 cache of 2MB. These workloads do not perform well because much of the requested data has been recently DMAed from the NIC to main memory, invalidating cached data at that address and necessitating a cache miss. Video streaming workloads do not suffer from high cache miss rates, but require more threads than Web Server workloads resulting in more threads to schedule at the kernel level. The timing-driven nature of each client connection stresses the kernel with many threads. Many threads are better handled on a multiprocessor than a uniprocessor. Therefore, we can generally say that single-thread optimized out-of-order processors do not perform well on Tier 1 server workloads. Another interesting property of most network workloads including these is the vast amount of time spent in kernel code. This section of code is largely comprised of interrupt handling in the NIC driver, packet transmission and network stack processing.

To perform well in these types of workloads an architecture must have a great deal of TLP run multiple threads as well as the large amount of processing to decode and encode the requested data. Thus a CMP or SMT architecture should be able to better utilize the processor die area.

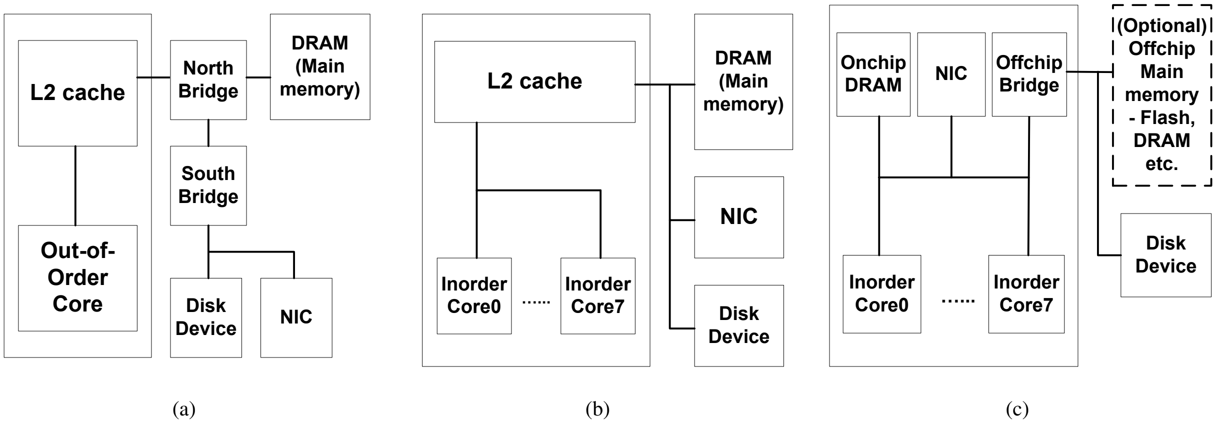


Figure 3. Block diagram of commonly used platform configurations. (a) general purpose processor platform, (b) conventional CMP platform without 3D stacking, (c) PicoServer platform using 3D stacking

2.3 Previous Work

Our work leverages 3D stacking technology and applies it to server workloads executed on a chip multiprocessor architecture. Therefore, we present previous work on chip multiprocessors and 3D stacking technology.

2.3.1 Chip Multiprocessor Architectures

Olkuton et. al[37] initially presented the possibility of building a simple multiprocessor architecture that were friendly to heavily threaded applications. In doing so they looked at SPLASH benchmarks along with highly parallel media benchmarks. However, the working set of these benchmarks was not large and the overall platform level evaluation of a chip multiprocessor was not presented in this work. Li and Martinez[31] looked at the power reduction in parallel architectures. They presented an analytical power estimation model for multiprocessor architectures and evaluated this model on SPLASH2 benchmarks. An optimal power, thermal point was presented due to the diminishing return for increasing parallel processing width. As parallel processing width is increased, leakage power increases linearly. Barroso et. al[14] presented a chip multiprocessor architecture to support server applications. A simple 8 in-order multiprocessor architecture was presented with a shared L2 cache. However, this work did not investigate the impact of network bandwidth in the architecture. The estimated power consumption was not mentioned in this work either. Additionally, the benchmarks presented in this paper focussed on Tier 2, 3 server applications. Kozyrakis et. al[28] applied the embedded DRAM technology to DSP applications. Specialized vector engines which existed adjacent to the embedded DRAM, contributed to the speedup in computation for vector friendly applications. This work performed extremely well for multimedia and matrix-oriented vector applications. Compared to previous work on chip multiprocessors, our work extends the benefits of chip multiprocessors by leveraging 3D stacking technology that allows us to add more cores and reduce core clock frequency in conventional simple in-order cores. We also modify the memory hierarchy exploiting the reduced access latency to large capacity on-chip DRAM, so the L1 caches in a chip multiprocessor architecture access on-chip DRAM directly and treat it as a part of main memory.

2.3.2 3D Stacking Technology

Black et. al[16] have presented circuit-level potential in this technology. They applied 3D stacking on an x86 core by implementing the floating point unit on the top layer and putting the rest of the execution units on the bottom layer. Their preliminary studies showed a 20% improvement in performance was achieved for SPEC2000 benchmarks and 3D stacking reduced the number of repeaters traditionally required for long global interconnects. With respect to work on 3D stacking technology alone, we have not seen any work that identifies a good application space for this technology and provides a thorough analysis.

3. Methodology

To explore the design space for 3D stacking technology, we estimated die area size based on previous publications and developed models for delay and power. They were derived from data published by industry and academia [11][23][38][3][12][16]. DRAM timing and power values were obtained from IBM and Micron technology datasheets [4]. The architectural aspects of our studies were obtained from a microarchitectural simulator called M5 [15] that is able to run Linux and evaluate full system-level performance. A Tier 1 server connected to multiple clients is modeled. The client requests are generated from user level network application programs. A detailed description of our methodology is described in the following subsections.

3.1 Simulation Studies

3.1.1 Full system architectural simulator

To evaluate the performance of our server we used the M5 full system simulator. M5 boots an unmodified Linux kernel on a configurable architecture. Multiple systems are created in the simulator to model the clients and server, and connected via an ethernet link model. The server side executes Apache—a web server, Fenice—a video streaming server, and NFS—a file server. The client side executes benchmarks that generate representative requests for dynamic and static web page content, video stream requests and network file commands respectively. For comparison purposes we defined a Pentium 4-like class system [40], and a chip multiprocessor-like system similar to [26]. We also looked at configurations using 3D stacking technology on these platforms to investigate in general the usefulness of this technology. The conventional Pentium 4-like

	OO4-small baseline / w. 3D stacking	OO4-large baseline / w. 3D stacking	Conventional CMP MP4/8 w.o. 3D stacking	PicoMP4-500MHz /1000MHz*	PicoMP8-500MHz /1000MHz*	PicoMP12-500MHz*
Operating Frequency	4GHz	4GHz	1GHz	500MHz / 1GHz	500MHz / 1GHz	500MHz
Number of Processors	1	1	4 / 8	4	8	12
Processor Type	out-of-order	out-of-order	in-order	in-order	in-order	in-order
Issue Width	4	4	1	1	1	1
L1 cache	2 way 16KB	2 way 128KB	4 way 16KB per core	4 way 16KB per core	4 way 16KB per core	4 way 16KB per core
L2 cache	8 way 256KB 7.5ns unloaded latency	8 way 2MB 7.5ns unloaded latency	8 way 2MB 16ns unloaded latency	N/A	N/A	N/A
Memory bus width	64bit@400MHz / 1024bit@250MHz	64bit@400MHz / 1024bit@250MHz	64bit@250MHz	1024bit@250MHz	1024bit@250MHz	1024bit@250MHz
Main Memory	512MB DDR2 DRAM	512MB DDR2 DRAM	512MB DDR2 DRAM	128MB DDR2 DRAM	192MB DDR2 DRAM	256MB DDR2 DRAM

* PicoServer platform using 3D stacking technology. The core clock frequency of PicoServer is typically 500MHz. PicoServer configurations with 1GHz core clock frequency are later used to show the impact of 3D stacking technology.

Table 3. Commonly used simulation configurations. Main memory latencies are generated from DDR2 DRAM models. L2 cache unloaded latency for single core and multicore configurations differ due to longer global interconnect lengths in multicore platforms[29].

class system without 3D stacking is illustrated in Figure 3(a). Figure 3(b) shows a conventional chip multiprocessor-like system and the PicoServer multiprocessor architecture is shown in Figure 3(c). We assume with 3D stacking technology, wider bus widths can be implemented with lower power overhead. Table 3 shows commonly used configurations in our simulations.

3.1.2 Server Benchmarks

We use several Tier 1 type benchmarks that directly interact with client requests. We used two web content handling benchmarks SURGE[13] and SPECweb99[9] to measure web server performance. A video streaming benchmark—Fenice[7] that uses the RTSP protocol along with the UDP protocol is chosen to measure behavior for on-demand workloads. Finally, for a file sharing benchmark we use an NFS server and stress it with dbench.

SURGE The SURGE benchmark represents client requests for static web content. SURGE is a multi-threaded, multi-process workload. We modified the SURGE fileset and used a Zipf distribution to generate reasonable client requests. Based on the Zipf distribution a static web page which is approximately 12KB in file size is requested 50% of the time in our client requests. We configured the SURGE client to have 20 outstanding client requests. It has been shown in [19] that the number of requests handled per second is consistent after 10 concurrent connections implying 20 concurrent connections is more than enough to fully utilize the server.

SPECweb99 To evaluate a mixture of static web content and simple dynamic web content, we used a modified version of SURGE to request SPECweb99 filesets. We used the default configuration for SPECweb99 to generate client requests. 70% of client requests are for static web content and 30% are for dynamic web contents. We also fixed the client request of SPECweb99 to have 20 outstanding requests with the same principle applied from [19].

Fenice On-demand video serving is also an important workload for Tier 1 servers. For copyright protection and live broadcasts, the RTSP protocol is commonly used for real-time video playback. Fenice is an open source streaming project [7] that provides workloads supporting the RTSP protocol. We modified it to support multithreading. Client requests were generated with a modified version of 'nemesi', a RTSP supporting MPEG player. 'nemesi' is also from the open source streaming project. Unlike event driven HTTP requests seen in SPECweb99 or SURGE, Video streaming servers require timing driven real-time packets to be constantly scheduled such that MPEG video frames are delivered precisely to the client side. Therefore, it is reasonable to assume the number of concurrent connections to be equal to the number of requests handled. We generated multiple client requests that fully utilized the server CPUs for a high quality 16Mbps frame rate, 720 × 480 resolution MPEG2 standard file.

	Pentium 4 90nm	ARM11 130nm	Xscale 90nm*	PicoServer MP 90nm†
L1 cache	16KB	16KB	32KB	16KB
L2 cache	1MB	N / A	N / A	N / A
Total Power(W)	89-103W	250mW @ 550MHz	850mW @ 1.5GHz	190mW @ 500MHz
Total Die Area(mm^2)	112	5-6	6-7	4-5

* Die area for a 90nm Xscale excludes L2 cache[39]

† For the PicoServer core, we estimated our power to be in the range of an ARM11, Xscale

Table 4. Published power consumption values for various microprocessors[18][1][39][40]

dbench This benchmark is commonly used to stress NFS daemons. In our tests we used the in-kernel NFS daemon which is multithreaded and available in standard Linux kernels. We generated NFS traffic using dbench on the client side that stressed the file server. dbench generates workloads that both read and write to the file server while locking these files so that a different client could not access it simultaneously. We assumed a ramdisk-like storage device for these experiments as we are not interested in the IO performance of our disk model.

3.2 Estimating Power and Area

Power and Area estimation at the architectural level is difficult to do with great accuracy. To make a reasonable estimation and show general trends, we resorted to industry and academia publications on die size area and we compared our initial analytical power models with real implementations and widely used cycle level simulation techniques. We discuss this further in the next subsections.

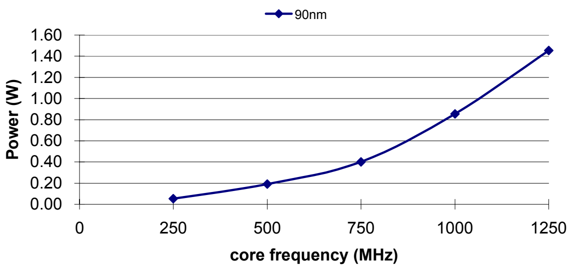


Figure 4. Processor power versus frequency plot generated from calibrating the well-known cubic law and voltage, frequency plot for 24FO4 using PTM 90nm process technology[6]

3.2.1 Processors

We relied to a large extent on figures reported in [18][1][39] for an ARM processor to estimate processor power and die area. The ARM is representative of a simple in-order 32 bit processor that would be suitable for the PicoServer. Due to the architectural similarities with our PicoServer cores, we extrapolated the die area and power consumption for our PicoServer cores at 500MHz from published data in [18][1][39]. Table 4 lists these estimates along with values listed in [1][39] and a Pentium 4 core for comparison. The power values listed in Table 4 include static power. Our estimates for a 500MHz PicoServer core are conservative compared to the ARM core values, especially with respect to [39]. Considering 850mW is consumed at 1.5GHz and 1.3V for the Xscale core, a power consumption of 190mW at 500MHz for our 90nm PicoServer core is conservative when applying the 3× reduction in clock frequency and the additional opportunities to scale voltage. For power consumption at other core clock frequencies, for example 1GHz, we generated a power versus frequency plot by calibrating the well-known cubic law [21] and frequency, voltage plot for 24 FO4—fan out of 4 inverter chain using PTM 90nm process technology. We assumed the logic depth of our PicoServer core to be 24FO4 from [39]. Figure 4 shows our plot.

64 bit support for a PicoServer core seems inevitable in the future for address sizes above 4GB. We expect the additional area and power overhead for 64 bit support in a PicoServer core to be modest when we look at the additional area and power overhead for 64 bit support in commercially available cores like MIPS and Xeon. As for the L2 cache, because we noticed numerous problems in power, area models in nanometer technology for caches, we referred to [41] and scaled the area, power number generated from real measurements. We assumed the power numbers in [41] were generated when the cache access rate was 100%. Therefore, we scaled the L2 cache power by size and access rate while assuming leakage power would consume 30% of the total L2 cache power.

3.2.2 Interconnect considering 3D stacking technology

	130nm	90nm
on-chip 2D 12mm	5.6nF	5.4nF
on-chip 3D 8mm	3.7nF	3.6nF
off-chip 2D	16.6nF	16.6nF

Table 5. Parasitic interconnect capacitance for on-chip 2D,3D and off-chip 2D for a 1024 bit bus

	Density	Area(mm^2)	Process
Samsung	64MB	71	90nm
Infineon - A*	64MB	87	110nm
Infineon - B	128MB	176	N / A
Micron	128MB	180	110nm

* Infineon A, B are 2 different types of DRAM chips

Table 6. DRAM die size from various vendors noted in Semiconductor SourceInsight 2005 [33]

For the purposes of this study, we have adopted the data published in [11][23][38] as typical of 3D stacking interconnects. In general, we found wafer to wafer—3D via, interconnect capacitance to be below 3fF. We also verified this with extracted parasitic capacitance values from 3D Magic, a tool recently developed at MIT. The extracted capacitance was found to be 2.7fF, which agrees with the results presented in [23]. By comparison with 2D on-chip interconnect, based on [24], a global interconnect wire was estimated to have capacitance of 400fF per millimeter. Therefore, we can assume that the additional interconnect capacitance in 3D stacking vias are negligible. As for the number of connections that are possible between dies a figure of 10,000 connects per square millimeter is reported. Our needs are much less. From our studies, we just need 1056 connections: 32 bits for our address bus and 1024 bits for the data bus. For estimating the interconnect capacitance on our processor and peripheral layer, we referred to [24] to generate analytical and projected values. We selected a wire length of 12mm to account for 1.3 times the width / height of a 80mm^2 die and scaled the wire length accordingly for smaller die sizes. We assumed we would gain a 33% reduction in wire capacitance compared to a 2D on-chip implementation from projections on interconnect wire length reduction shown in [20]. Based on these initial values, we calculated the number of repeaters required to drive the interconnect range from 250 $\sim$ 400MHz from hspice simulations. We found we needed only a maximum of 2 $\sim$ 3 repeaters to drive this bus since the frequency of this on-chip wide bus was relatively slow.

We measured the toggle rate and access rate of these wires and calculated power using the well-known dynamic power equation to calculate interconnect power. Table 5 shows the expected interconnect capacitance for 1024bits in the case of 2D on-chip, 3D stacking, and 2D off-chip implementations. Roughly speaking, on-chip implementations have at most 33% capacitance of an off-chip implementation. Furthermore, since the supply voltages in IO pads—typically 1.8 $\sim$ 2.5V, are generally higher than the core supply voltage, we find the overall maximum power for an off-chip implementation consumes an order of magnitude more power than an on-chip and off-chip average interconnect power. With modest toggle rates, small to modest access rates for typical configurations found in our benchmarks and modest bus frequency—250MHz, we conclude that on-chip interconnect power contributes very little to overall power consumption.

3.2.3 DRAM

We made DRAM area estimates for the PicoServer using the data in Table 6. Currently, it is reasonable to say that 80mm^2 of chip area is required for 64MB of DRAM in 90nm technology.

Conventional DRAM is packaged separately from the processor and is accessed through IO pad pins and wires on a PCB. However, for our architecture, DRAM exists on-chip and connects to the processor and peripheral through a 3D stacking via. Therefore, the pad power consumed by the packages, necessary for driving signals off-chip across the PCB, should not be added to our design. Using the Micron DRAM spreadsheet calculator[4], modified to not add pad power, and profile data from M5 including the number of cycles

	SDRAM	DDR2 DRAM	XDR DRAM	L2 Cache @1.2GHz	On-chip DRAM 3D IC
Bandwidth (GB/sec)	1.0	5.2	31.3	21.9	31.3
Average access latency(ns)	30ns	25ns	28ns	16ns	25ns

Table 7. Bandwidth and latency suggest on-chip DRAM can easily provide enough memory bandwidth compared to an L2 cache noted in [29][41]. Average access latency for DRAM is estimated to be t_{RCD}+t_{CAS} where t_{RCD} denotes RAS to CAS delay and t_{CAS} denotes CAS delay. For, XDRAM t_{RAC-R} is used where t_{RAC-R} denotes the read access time.

spent on DRAM reads, writes and page hit rates, we generated an average power for DRAM. We compared the estimated power from references on DRAM and especially with the DRAM power values generated from the SunFire T2000 Server Power Calculator[10]. The Micron spreadsheet uses actual current measurements for each DRAM operation—read, write, refresh, bank precharge etc. We assumed a design with a 1.8V voltage supply.

3.2.4 Network Interface Controller—NIC

Network Interface Controller power was difficult to model analytically due to lack of information on the architecture of NICs. For our simulations, we looked at the National Semiconductor 82830 gigabit ethernet controller. This chip implements the MAC layer of the ethernet card and interfaces with the physical layer—PHY using the Gigabit Media Independent Interface—GMII interface. For power, we analyzed the datasheet and found the maximum power consumed by this chip to be 743mW[5]. This power number is for 180nm technology. We assumed maximum power is consumed when all the input and output pins were active. By doing so, we calculated the number of bytes written and read from the chip and normalized it to the maximum case. We assumed static power consumed 30% of the maximum chip power.

4. PicoServer Architecture

Table 7 shows the latency and bandwidth achieved for conventional DRAM, XDR DRAM, L2 cache and on-chip DRAM using 3D stacking technology. With a 1024 bit wide bus, the memory latency and bandwidth achieved in a 3D stacking on-chip DRAM is comparable to an L2 cache and XDR DRAM. This suggests an L2 cache is not needed if stacking is used. Furthermore, the removal of off-chip drivers in conventional DRAM reduces access latency by more than 50% [35]. We have ignored this in our calculations, so our results are conservative in this regard. However, this strengthens our argument that on-chip DRAM can be as effective as a L2 cache. Another example that strengthens our case is that DRAM vendors are producing and promoting DRAM implementations with reduced random access latency. Therefore, our PicoServer architecture does not have an L2 cache and the on-chip DRAM is connected through a shared bus architecture to the L1 caches of each core. The role of this on-chip DRAM is a primary main memory.

The PicoServer architecture is comprised of single issue in-order processors that together create a chip multiprocessor which is a natural match to applications with a high level of TLP [26]. Each PicoServer CPU core is typically clocked at 500MHz and has an instruction and data cache, with the data caches using a MESI cache coherence protocol. Our studies showed the majority of bus traffic is generated from cache miss traffic, not cache coherence. This is due to the properties of the target application space and the small cache—16KB size per core. With current densities, the capacity of the on-chip DRAM stack in PicoServer is hundreds of megabytes. In the near future this will rise to several gigabytes as noted in the Table 1. Other components such as the Network Interface Controller (NIC), DMA controller, and additional peripherals that are required in implementing a full system are integrated on the CPU die.

	130nm	110nm	90nm
DRAM stack 4 layer each layer 80mm^2	128MB	192MB	256MB

Table 8. Projected on-chip DRAM size for varying process technologies. Area estimates are generated based on Table 6. 80mm^2 of die size is similar to that of a Pentium M at 90nm.

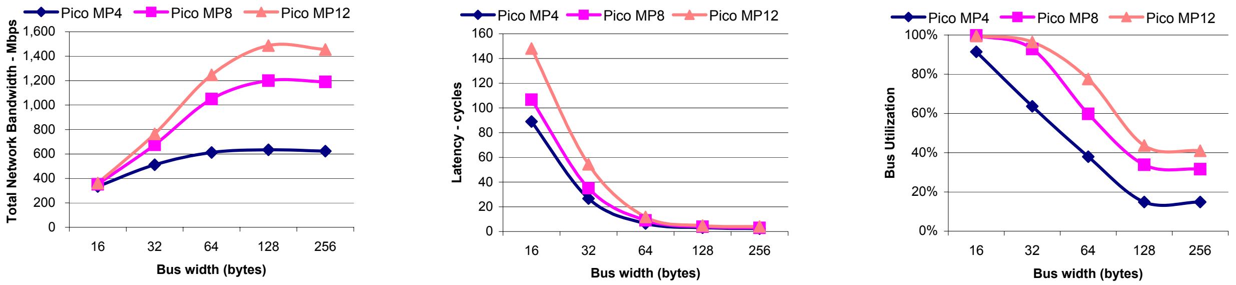
4.1 Wide shared bus architecture

PicoServer adopts a wide shared bus architecture that provides high memory bandwidth and fully utilizes the benefits of 3D stacking technology. Our bus architecture was determined from SURGE runs on M5. SURGE generated a representative cache miss rate per core on our benchmarks. To explore the design space of our bus architecture, we first ran simulations for varying the bus width on a single shared bus—ranging from 128 bits to 2048 bits. Network performance is measured to determine the impact of bus width on the PicoServer. As shown in Figure 5(a), a relatively wide data bus is necessary to achieve scalable network performance to satisfy the outstanding cache miss requests. This is because of the high bus contention on the shared data bus for high bus traffic that is generated for narrow bus widths as shown in Figure 5(b) and Figure 5(c). As we decrease the bus width, the bus traffic increases resulting in superlinear increase in latency. Reducing bus utilization implies reduced bus arbitration latency, thus improving network bandwidth. Wide bus widths also help speedup NIC DMA transfers by allowing a large chunk of data be copied in one transaction. A 1024 bit bus width seems reasonable for our typical PicoServer configurations—4, 8, 12 multiprocessors, since network performance saturated after that point. We also looked at interleaved bus architectures but found with our given L1 cache miss rates, a 1024 bit bus is sufficient enough to handle the bus requests. For architectures and workloads that generate higher bus requests by increasing the number of cores to 16 or more with higher L1 cache miss rates—more than 10%—then interleaving the bus is more effective.

4.2 The role of on-chip DRAM

We projected the DRAM roadmap in PicoServer for a die size of 80mm^2 in Table 8. To obtain a total DRAM size of 256 MB, we assume DRAM is made up of 4 layers of the final die allowing us to integrate aggressive capacity for on-chip stacked DRAM. With current technology—90nm, it is feasible to create a 4 layer stack containing 256MB of physical memory. Although a large amount of physical memory is typical in server farms (4 to 16GB), with the short sample time of a simulator, it is difficult for a benchmark to touch such a large memory space. Depending on the workload of each PicoServer, 256MB of physical memory may be enough. From our measurements on memory usage for Tier 1 applications, we found a modest amount—no more than 64MB—of main memory is occupied by the user application, data and the kernel OS code. The remainder of the memory is either free or used as a disk cache. Considering the fact that 256MB can be integrated on-chip, we project a large portion of on-chip DRAM to be used as a disk cache. Therefore, for Tier 1 applications that require small/medium filesets, a on-chip DRAM of 256MB is enough to effectively handle client requests.

For large filesets, there are several options to choose from. First, we could add additional on-chip DRAM by stacking additional



(a) Network bandwidth vs. Bus width

(b) Overall loaded latency vs. Bus width

(c) Bus utilization vs. Bus width

Figure 5. Network performance for various shared bus architectures based on our L1 cache size—16KB on SURGE. We assumed a CPU clock frequency of 500MHz for these experiments. Our bus architecture must be able to handle high bandwidths as the number of processors increase.

DRAM dies. From the ITRS roadmap in Table 1, we recall that the number of stacked dies we assume is conservative. With aggressive die stacking, we could add more die stacks to improve on-chip DRAM capacity—ITRS projects 7 $\sim$ 9 layers to be possible in the next 2 $\sim$ 4 years. Another alternative is to add a secondary off-chip main memory which functions as a disk cache device. From our simulations, we found that the access latency of this secondary off-chip main memory could be as slow as 50μs and still generate equal network bandwidth. The multithreaded nature of Tier 1 applications hide thread stalls due to the long access latency to off-chip DRAM through interleaved thread execution. An access latency as slow as 50μs implies that Flash memory that consumes less active / standby power can be used as off-chip secondary main memory. Therefore, for workloads requiring large filesets, we could build a NUMA system with fast on-chip DRAM and relatively slower off-chip secondary main memory that could be implemented from Flash memory or slow low power DRAM. The fast on-chip DRAM would primarily hold code, data and a small disk cache and the slow off-chip main memory would function as a large disk cache device.

To maximize the benefits of 3D stacking technology, the conventional DDR2 DRAM interface needs to be modified for PicoServer's 3D stacked on-chip DRAM. Conventional DDR2 DRAMs are designed assuming a small pin count and use address multiplexing and burst mode transfer to make up for the limited number of pins. With 3D stacking technology, there is no need to use narrow data widths and furthermore address multiplexing which requires multi-phase commands such as a RAS followed by a CAS. Instead, the additional logic required in latching and muxing narrow address / data can be removed. The requested addresses can be sent as a single command while data can be driven out in large chunks. DRAM vendors already provide interfaces that do not require address multiplexing such as Reduced Latency DRAM from Micron [8] and NetDRAM[2] from Samsung. This suggests the interface for 3D stacked on-chip DRAM can be tailored with minor tweaks.

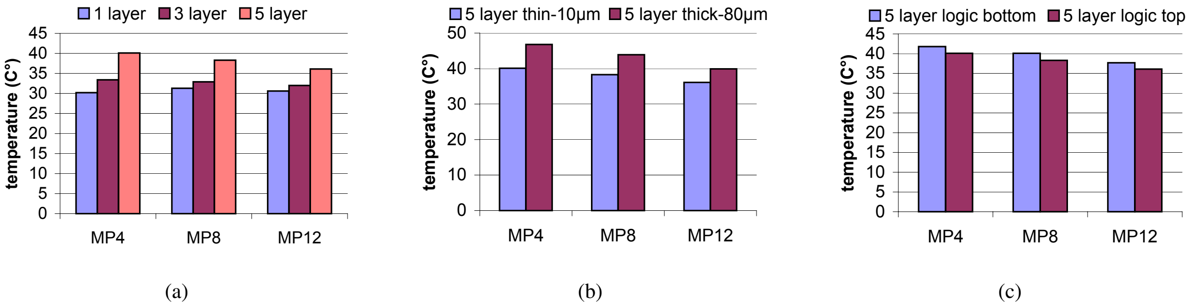
4.3 The need for Multiple NICs on a CMP architecture

A common problem of servers with large network pipes is handling the hundreds of thousands of packets that might arrive each second. Interrupt coalescing is one method of dealing this problem. This method works by starting a timer when a non-critical event occurs. Any other non-critical events that occur before the timer expires are coalesced into one interrupt and thus the number of interrupts can be reduced. Even with this technique however the number of interrupts received by a relatively low frequency processor, such as one of the PicoServer cores, can overwhelm it. In our simulations we get around this difficulty by having multiple NICs each having their interrupt lines routed to a different processor. Although this method would be valid, a smarter single NIC that could route interrupts to multiple CPUs, each with separate DMA descriptors and TX/RX queues, could be built. For an 8 chip-multiprocessor architecture with 1 NIC and an on-chip DRAM integrated by using 3D stacking technology, we found the average utilization per processor to be below 60% as one processor could not manage the NIC by itself. To fully utilize each processor in our multiple processor architecture, we inserted 1 NIC for 2 processors. For example, a 4 CMP architecture would have 2 NICs, a 8 CMP architecture would have 4 NICs and so forth. However, as mentioned above, this could be one NIC either with multiple interface IP addresses or an intelligent method of load balancing packets to multiple processors. Such a NIC would need to keep track of network protocol states at the session level. There have been previous studies of intelligent workload balancing on NICs to achieve optimal throughput on platforms[19]. TCP splicing and handoff are also good examples of intelligent load balancing at higher network layers[34].

4.4 Thermal concerns in 3D stacking

A potential concern with 3D stacking technology is heat containment. To address this concern, we investigated the thermal impact of 3D stacking on the PicoServer architecture. Since we could not measure temperature directly on a real 3D stacked platform, we modeled the 3D stack onto the grid model in Hotspot [25]. Mechanical thermal simulators such as FLOWTHERM and ANSYS were not considered in our studies due to the limited information we could obtain from the 3D stacking process. We believe Hotspot's RC equivalent heat flow model is adequate to show trends and potential concern in 3D stacking. Since this paper describes the usefulness of integrating 3D stacking into the server space, instead of describing the details in heat transfer, we present general trends. We leave detailed studies to future work and published references that cover heat in 3D stacking as a primary topic.

The primary thermal issue in devices utilizing 3D stacking in heat containment due to the interface material — SiO_2 — and the free air interface between silicon and air. Silicon and metal conduct heat much more efficiently. We configured our PicoServer architecture for various scenarios by 1) varying the amount of stacked dies, 2) varying the location of the primary heat generating die—the



(a) (b) (c)

Figure 6. Maximum junction temperature for sensitivity experiments on Hotspot. (a)varying the number of layers, (b)varying 3D interface thickness, (c)varying location of logic die. A core clock frequency of 500MHz is assumed in calculating power density. For 1, 3 layer configurations, we assumed no on-chip DRAM or a smaller capacity of DRAM—2 layers is integrated.

logic die on our platform, 3) varying the thickness of the SiO_2 insulator that is typically used in between stacked dies. Our baseline configuration assumes a logic die directly connected to a heat sink assuming 27C° room temperature. We assumed a naive floorplan of our PicoServer architecture and varied the number of processors. Hotspot requires input for properties in material and power density to generate steady state temperature throughout the platform. We extracted 3D stacking properties from [27][32][42] and assigned power density at the component level based on area and power projections for each component. Components were modeled at the platform-level—processor, peripheral, global bus interconnect, etc. We generated maximum junction temperature in our PicoServer architecture shown in Figure 6.

Figure 6(a) shows the sensitivity to the number of stacked layers. We find a $5 \sim 10$C° increase in maximum junction temperature in our 5 layer PicoServer architecture. Figure 6(b) shows the sensitivity to the 3D stacking dielectric interface. We compared the effect of the SiO_2 thickness (the interface material) for $10\mu m$ and $80\mu m$. In [16][27][32][42] we find the maximum thickness of the interface material does not exceed $10\mu m$ for 3D stacking. The $80\mu m$ point is selected to show the impact of heat containment as the thickness is increased substantially. It results in a 6 degree increase in junction temperature. While notable this is not a great change given the dramatic change in material thickness. Figure 6(c) shows the sensitivity to placement in the stack—top or bottom layer. We find the primary heat generating die is not sensitive to the geographic location.

We believe heat containment for having multiple stacked layers is not a major limitation in the PicoServer platform. The power density is relatively low for our architecture. It does not exceed 5W/cm^2. As a result, the maximum junction temperature does not exceed 50C°. 3D vias can also act as heat pipes, which we didn't take into account in our analysis, however this is expected to improve the situation. An intelligent placement would assign the heat generating layer (the processor layer) adjacent to the heat sink resulting in a majority of the heat being transferred to the heat sink. There is independent support for our conclusions in [17][22].

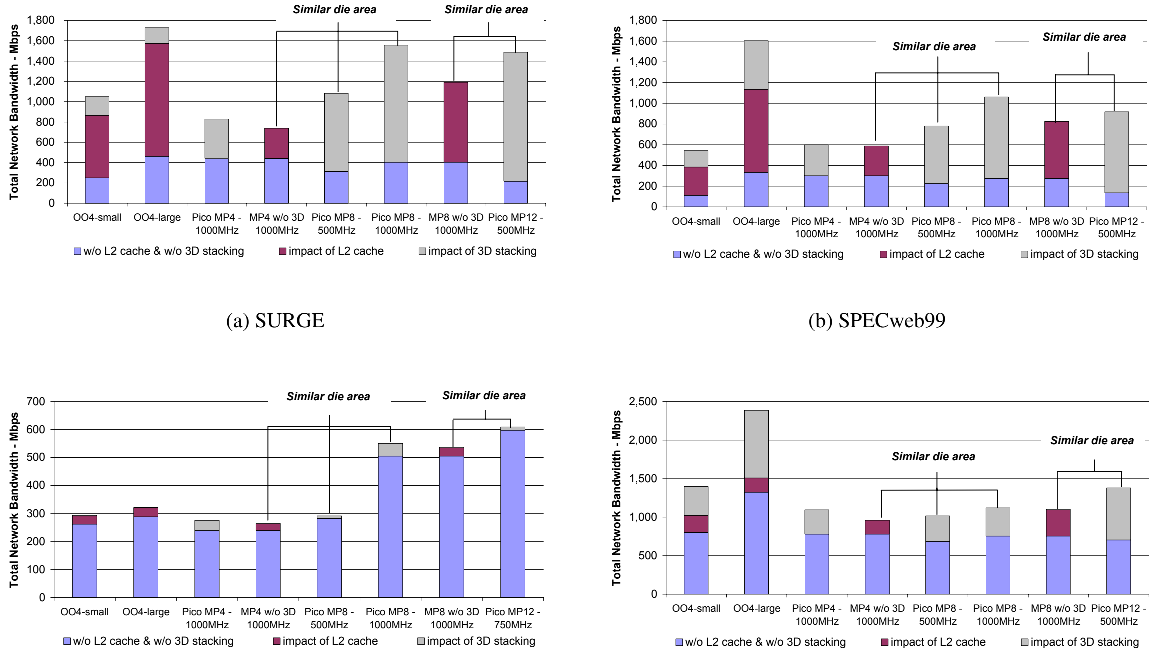
5. Results

To evaluate the PicoServer architecture two metrics are important—network bandwidth and power. Network bandwidth is a good indicator of overall system performance because it is a measure of how many requests were serviced times the average size of each request. This metric is better than requests responded to per second because of the high variability in the requested data size. In this section, we compare various PicoServer configurations to other architectures first in terms of achievable network bandwidth and then in terms of power. Since the PicoServer has not been implemented, we use a combination of analytical models and published data to make a conservative estimate about the power dissipation of various components. Finally we present a pareto chart showing the energy efficiency of the PicoServer architecture.

5.1 Overall Performance

Figure 7 shows the network bandwidth for our Tier 1 workload runs. We break down the contribution to network bandwidth with respect to a baseline with no L2 cache and a narrow (64bit) bus width, having an L2 cache, and implementing the benefits of 3D stacking technology. For simple multicore architectures, we recall in Table 7 the L2 cache unloaded latency is similar to the DRAM access latency. Hence, we are able to make comparisons that differentiate the impact of 3D stacking technology with the impact of having an L2 cache. We show in separate bars the impact of having an L2 cache or adopting 3D stacking technology. It shows that 3D stacking technology alone improves overall performance equal to or more than having an L2 cache. A fair comparison for a fixed number of cores, for example, would be a Pico MP4-1000MHz versus a conventional CMP MP4 without 3D-1000MHz. In general workloads that generated modest to high cache miss rates, SURGE, SPECweb99 and dbench, showed dramatic improvement from adopting 3D stacking technology. Fenice that is bounded by video stream computation, generated low cache miss rates resulting in marginal improvement with adding an L2 cache or adopting 3D stacking technology. Since Video streaming workloads inherently support many client connections, we found TLP friendly architectures perform well for this benchmark—the more cores you have, the higher the network bandwidth.

For OO4 configurations, we combine the impact of having an L2 cache and 3D stacking since the unloaded L2 cache latency on a uniprocessor is likely to be smaller than the access latency to a large capacity DRAM making it less appealing to only have a high bandwidth on-chip DRAM implemented from 3D stacking. We find that 3D stacking improves performance by 15% on OO4 configurations. When we compare a OO4 architecture without 3D stacking with our PicoServer architecture, a PicoServer MP8 operating at 500MHz performs better than a 4GHz OO4 processor with a small L1 and L2 cache of 16KB and 256KB respectively. For a similar die area comparison, we believe comparing PicoServer MP8 and a OO4-small architecture is a fair comparison considering the additional die area required for a OO4-large that has a L1 cache size of 128KB and a L2 cache size of 2MB.



(a) SURGE

(b) SPECweb99

(c) Fenice

(d) dbench

Figure 7. Network performance measured for varying processor frequency and processor type. For PicoServer CMPs, we fixed the on-chip data bus width to 1024bits and bus frequency to 250MHz. For a Pentium 4-like configuration, we placed the NIC on the PCI bus and assumed the memory bus frequency to be 400MHz. For a MP4, MP8 without 3D stacking configuration, to be fair we assumed no support for multithreading and a L2 cache size of 2MB. The external memory bus frequency was assumed to be 250MHz.

When we assume that the area occupied by the L2 cache in our conventional CMP MP4/8 without 3D stacking technology is replaced with additional processing cores—a benefit made possible by using 3D stacking technology—a comparison in network performance for similar die area can be conducted on Pico MP8-500MHz versus a conventional MP4 without 3D-1000MHz and Pico MP12-500MHz versus a conventional MP8 without 3D-1000MHz—for Fenice, compare with Pico MP12-750MHz. Our results suggest that on average, additional processing elements and reducing core clock frequency by half on our Tier 1 workloads improve network performance by 10 $\sim$ 20% for more than 50% reduction in power—shown in Section 5.2. Our estimated area for adding extra cores are extremely conservative suggesting more cores could be added thereby resulting in even more improvement in network bandwidth.

5.2 Overall Power

Processor power still dominates overall power in PicoServer architectures. Figure 8 shows the average power consumption based on our power estimation techniques for Tier 1 server application runs. We find PicoServer with a core clock frequency of 500MHz is estimated to consume somewhere between 2 $\sim$ 3 Watts for 90nm process technology depending on the design points (optimal power or network bandwidth). Overall power is primarily used by the simple in-order cores. NIC power consumed a considerable amount due to the increase in number of NICs when increasing the number of processors, however as described in section 4.3 an intelligent NIC designed for this architecture could be more power efficient as you would only need one. Other components such as the interconnect and the DRAM make marginal contributions to overall system power due to the modest access rates and toggle rates of these components. For example, DRAM is accessed no more than 30% of the time. Therefore, using [4], we find the DRAM power consumption to be no more than 400mW, which is far smaller than the combined power consumed by the in-order cores.

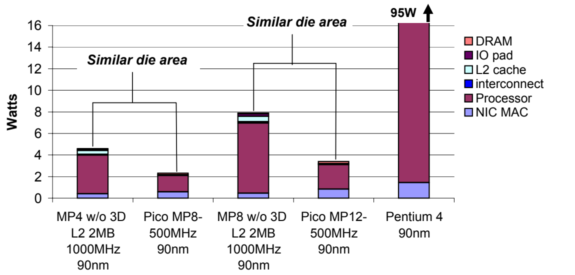
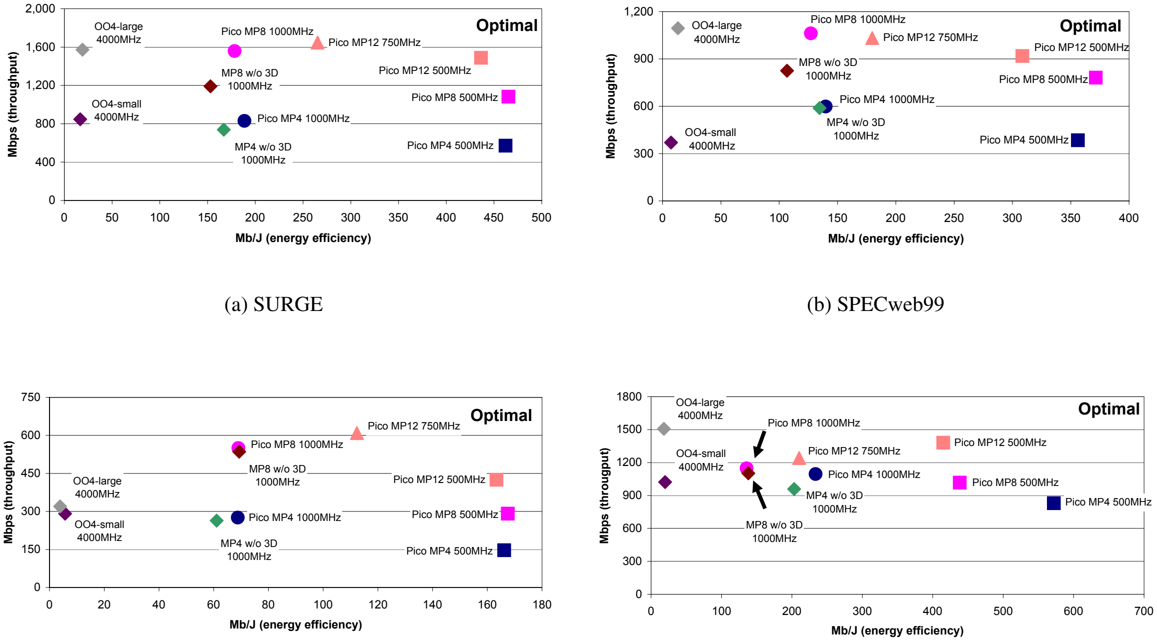


Figure 8. Breakdown of average power for 4, 8,12 PicoServer architectures using 3D stacking technology for 90nm process technology. We expect 2 $\sim$ 3W to be consumed at 90nm. An MP8 without 3D stacking operating at 1GHz is estimated to consume 8W at 90nm.

Comparing our PicoServer architecture with other architectures, we do very well. For a similar die area comparison, we use less than half the power when we compare Pico MP8 / 12-500MHz with a



(a) SURGE

(b) SPECweb99

(c) Fenice

(d) dbench

Figure 9. Energy efficiency, Performance pareto chart generated for 90nm process technology. 3D stacking technology enables new CMP architectures that are significantly energy efficient.

conventional MP4 / 8 without 3D stacking with an L2 cache at 1000MHz. We also recall in section 5.1 that performance-wise for a similar die area, our PicoServer architectures perform on average $10 \sim 20\%$ better than conventional CMP configurations. Furthermore, we use less than 10% of the power of a Penitum4 processor and as in the previous section perform comparably. At 90nm technology, it can be projected that the power budget for a typical PicoServer platform satisfies mobile / handheld power constraints noted in ITRS projections. This suggests the potential of implementing server-type applications in ultra small form factor platforms. We generally find that our PicoServer architecture provides excellent performance while using energy efficiently.

5.3 Energy efficiency, Network Performance Pareto Chart

In Figure 9, we present a pareto chart for PicoServer depicting the energy efficiency and network performance. The points on this plot show the large out-of-order cores and the conventional CMP MP4/8 without 3D stacking processors we have described up to this point as well as our PicoServer with four, eight, or twelve cores. On the y-axis we present Mbps and on the x-axis we show Mb/J. From Figure 9, it is possible to find the optimal configuration of processor number and frequency for a given energy efficiency / network bandwidth constraint.

Additionally from Figure 9, we find our PicoServer architectures clocked at modest core frequency—500MHz are $2 \sim 4\times$ energy efficient than conventional chip-multiprocessor architectures without 3D stacking technology. The primary powersavings can be attributed to 3D stacking technology that enables a reduction in core clock frequency while providing high network bandwidth. A sweetspot in system-level energy efficiency for our plotted datapoints can also be identified within our PicoServer architectures when looking at Pico MP4/8/12-500MHz. These sweetspots in energy efficiency come from diminishing return in network bandwidth improvement for increasing parallel processing width. The increase in parallel processing width—adding additional cores, raises many issues related to inefficient interrupt balancing, kernel process / thread scheduling and resource allocation that result in diminishing return.

6. Conclusions

In this paper, we presented the potential of 3D stacking technology for implementing Tier 1 servers. The resulting systems have significantly improved energy efficiency in a compact form factor. An 8-way PicoServer running at 500MHz can deliver up to 1.1Gbps of network bandwidth within a 3W power budget in a 90nm process technology. These power results are $2 \sim 3\times$ better than a multicore architecture without 3D stacking technology and an order of magnitude better than what can be achieved using a general purpose processor. The ability to tightly couple large amounts of memory to the cores through wide and low-latency interconnect pays dividends by reducing system complexity and creates opportunities to implement main memory with non-uniform access latency. With the access latency of on-chip DRAM being comparable to the L2 cache, we found the L2 cache die area can be replaced with additional cores resulting in core clock frequency reduction while achieving higher throughput. For an area-equivalent PicoServer configuration using 12 processors at 500MHz without an L2 cache yields a substantial improvement in network bandwidth while reducing power consumption by more than 50% compared to a conventional 8-way 1GHz chip multiprocessor with a 2MB L2 cache. For future work, we plan to expand our application space to Tier 2 and 3 servers.

Acknowledgments

We would like to thank the reviewers for providing helpful comments about our paper. This project is supported by the National Science Foundation under grants NSF-ITR CCR-0325898 and CCR-0219640. This work was also supported by gifts from Intel, an Intel Fellowship, a Gates Millennium Fellowship, and a Sloan Research Fellowship.

References

[1] ARM 11 MPcore. `http://www.arm.com/products/CPUs/ARM11MPCoreMultiprocessor.html`.

[2] Evolution of network memory. `http://www.jedex.org/images/pdf/jack_troung_samsung.pdf`.

[3] FaStack 3D RISC super-8051 microcontroller. `http://www.tachyonsemi.com/OtherICs/datasheets/TSCR8051Lx_1_5Web.pdf`.

[4] The Micron system-power calculator. `http://www.micron.com/products/dram/syscalc.html`.

[5] National semiconductor DP83820 10 / 100 / 1000 Mb/s PCI ethernet network interface controller.

[6] Predictive technology model. `http://www.eas.asu.edu/~ptm`.

[7] (LS)^3-libre streaming, libre software, libre standards an open multimedia streaming project. `http://streaming.polito.it/`.

[8] RLDRAM memory. `http://www.micron.com/products/dram/rldram/`.

[9] SPECweb99 benchmark. `http://www.spec.org/osg/web99/`.

[10] Sun Fire T2000 Server Power Calculator. `http://www.sun.com/servers/coolthreads/t2000/calc/index.jsp`.

[11] ITRS roadmap. Technical report, 2005.

[12] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat. 3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration. *Proc. of IEEE*, 89(5):602–533, May 2001.

[13] P. Barford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *Measurement and Modeling of Computer Systems*, pages 151–160, 1998.

[14] L. Barroso, K. Gharachorloo, R. McNamara, A. Nowatzyk, S. Qadeer, B. Sano, S. Smith, R. Stets, and B. Verghese. Piranha: A scalable architecture based on single-chip multiprocessing. In *Proc. Int'l Symp. on Computer Architecture*, June 2000.

[15] N. L. Binkert, R. G. Dreslinski, L. R. Hsu, K. T. Lim, A. G. Saidi, and S. K. Reinhardt. The M5 simulator: Modeling networked systems. *IEEE Micro*, 26(4):52–60, Jul/Aug 2006.

[16] B. Black, D. Nelson, C. Webb, and N. Samra. 3D processing technology and its impact on iA32 microprocessors. In *Proc. Int'l Conf. of Computer Design*, pages 316–318, 2004.

[17] T.-Y. Chiang, S. J. Souri, C. O. Chui, and K. C. Saraswat. Thermal analysis of heterogeneous 3-D ICs with various integration scenario. In *IEDM Technical Digest*, pages 681 – 684, Dec. 2001.

[18] L. T. Clark, E. J. Hoffman, J. Miller, M. Biyani, Y. Liao, S. Strazdus, M. Morrow, K. E. Verlarde, and M. A. Yarch. An embedded 32-b microprocessor core for low-power and high-performance applications. *IEEE Journal of Solid State Circuits*, 36(11):1599–1608, Nov. 2001.

[19] E. L. Congduc. Packet classification in the NIC for improved SMP-based internet servers. In *Proc. Int'l Conf. on Networking*, Feb. 2004.

[20] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, and P. D. Franzon. Demystifying 3D ICs: The pros and cons of going vertical. *IEEE Design & Test of Computers*, 22(6):498–510, 2005.

[21] M. J. Flynn and P. Hung. Computer architecture and technology: Some thoughts on the road ahead. In *Proc. Int'l Conf. on Engineering of Reconfigurable Systems and Algorithms*, pages 3–16, 2004.

[22] B. Goplen and S. S. Sapatnekar. Thermal via placement in 3D ICs. In *Proc. Int'l Symp. on Physical Design*, pages 167–174, Apr. 2005.

[23] S. Gupta, M. Hilbert, S. Hong, and R. Patti. Techniques for producing 3D ICs with high-density interconnect. `www.tezzaron.com/about/papers/ieee_vmic_2004_finalsecure.pdf`.

[24] R. Ho and M. Horowitz. The future of wires. *Proc. of the IEEE*, 89(4), Apr. 2001.

[25] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusam. Compact thermal modeling for temperature-aware design. In *Proc. Design Automation Conf.*, June 2004.

[26] P. Kongetira, K. Aingaran, and K. Olukotun. Niagara: A 32-way multithreaded Sparc processor. *IEEE Micro*, 25(2):21–29, Mar. 2005.

[27] M. Koyanagi. Different approaches to 3D chips. `http://asia.stanford.edu/events/Spring05/slides/051205-Koyanagi.pdf`.

[28] C. Kozyrakis, J. Gebis, D. Martin, S. Williams, I. Mavroidis, S. Pope, D. Jones, D. Patterson, and K. Yelick. Vector IRAM: A media-oriented vector processor with embedded DRAM. In *Hotchips*, Aug. 2000.

[29] J. Laudon. Performance/watt: the new server focus. *SIGARCH Computer Architecture News*, 33(4):5–13, 2005.

[30] K. Lee, T. Nakamura, T. Ono, Y. Yamada, T. Mizukusa, H. Hashimoto, K. Park, H. Kurino, and M. Koyanagi. Three-dimensional shared memory fabricated using wafer stacking technology. In *IEDM Technical Digest.*, pages 165–168, Dec 2000.

[31] J. Li and J. F.Martinez. Power-performance implications of thread-level parallelism in chip multiprocessors. In *Proc. Int'l Symp. on Performance Analysis of Systems and Software*, Mar. 2005.

[32] J. Lu. Wafer-level 3D hyper-integration technology platform. `www.rpi.edu/~luj/RPI_3D_Research_0504.pdf`.

[33] G. MacGillivray. Process vs. density in DRAMs. `http://www.eetasia.com/ARTICLES/2005SEP/B/2005SEP01_STOR_TA.pdf`.

[34] D. A. Maltz and P. Bhagwat. TCP splicing for application layer proxy performance. Research Report RC 21139, IBM, Mar. 1998.

[35] R. E. Matick and S. E. Schuster. Logic-based eDRAM: origins and rationale for use. *IBM Journal of Research and Development*, 49(1), Jan. 2005.

[36] T. Mudge. Power: A first-class architectural design constraint. *IEEE Computer*, 34(4), Apr. 2001.

[37] K. Olukotun, B. A. Nayfeh, L. Hammond, K. Wilson, and K. Chang. The case for a single-chip multiprocessor. In *Proc. Int'l Conf. on Arch. Support for Prog. Lang. and Oper. Sys.*, Oct. 1996.

[38] A. Rahman and R. Reif. System-level performance evaluation of three-dimensional integrated circuits. *IEEE Trans. on VLSI*, 8, Dec. 2000.

[39] F. Ricci, L. T. Clark, T. Beatty, W. Yu, A. Bashmakov, S. Demmons, E. Fox, J. Miller, M. Biyani, and J. Haigh. A 1.5GHz 90nm embedded microprocessor core. In *Proc. Symp. on VLSI Circuits*, June 2005.

[40] J. Schutz and C. Webb. A scalable X86 CPU design for 90 nm process. In *Proc. Int'l Solid-State Circuits Conference*, Feb. 2004.

[41] D. Wendell, J. Lin, P. Kaushik, S. Seshadri, A. Wang, V. Sundararaman, P. Wang, H. McIntyre, S. Kim, W. Hsu, H. Park, G. Levinsky, J. Lu, M. Chirania, R. Heald, and P. Lazar. A 4MB on-chip l2 cache for a 90nm 1.6GHz 64b SPARC microprocessor. In *Proc. Int'l Solid-State Circuits Conference*, Feb. 2004.

[42] L. Xue, C. C. Liu, H.-S. Kim, S. Kim, and S. Tiwari. Three-dimensional integration: Technology, use, and issues for mixed-signal applications. *IEEE Trans. on Electron Devices*, 50:601–609, May 2003.