AFRL-AFOSR-VA-TR-2019-0094

Series-Stacked Computer Architectures - Leveraging Multi-Core Systems for Extreme Miniaturization of Mobile Computing Platforms

Arijit Banerjee
UNIVERSITY OF ILLINOIS
506 S WRIGHT STREET SUITE 364
URBANA, IL 61801-3649

04/16/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
16-04-2019	Final Performance	01 May 2015 to 30 Apr 2018

4. TITLE AND SUBTITLE
Series-Stacked Computer Architectures - Leveraging Multi-Core Systems for Extreme Miniaturization of Mobile Computing Platforms

5a. CONTRACT NUMBER

5b. GRANT NUMBER
FA9550-15-1-0128

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)
Arijit Banerjee, Robert Pilawa

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY OF ILLINOIS
506 S WRIGHT STREET SUITE 364
URBANA, IL 61801-3649 US

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AF Office of Scientific Research
875 N. Randolph St. Room 3112
Arlington, VA 22203

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/AFOSR RTA2

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-AFOSR-VA-TR-2019-0094

12. DISTRIBUTION/AVAILABILITY STATEMENT
A DISTRIBUTION UNLIMITED: PB Public Release

13. SUPPLEMENTARY NOTES

14. ABSTRACT
With the increased number and sophistication of intelligence gathering devices (e.g., cameras, sensors, radar) on modern
aircrafts and UAVs, there is a continued need for increased computational performance. In the size, energy, power, and weight
constrained platforms operated by the Air Force, it is of utmost importance to perform computation and data analysis in the most efficient manner possible. In this work, we are addressing this challenge through the development of new power delivery architecture that is inherently scalable to massive number of cores, and low-power future data centers.

15. SUBJECT TERMS
computer architecture, Multicore, Mobile Computing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			NGUYEN, TRISTAN
Unclassified	Unclassified	Unclassified	UU		19b. TELEPHONE NUMBER *(Include area code)* 703-696-7796

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Series-Stacked Computer Architectures – Leveraging Multi-Core Systems for Extreme Miniaturization of Mobile Computing Platforms

Robert C. N. Pilawa-Podgurski
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801
E-mail: pilawa@illinois.edu

Final report, May 1st, 2015 - April 30, 2018
ATTN: Dr. Tristan Nguyen

Abstract

With the increased number and sophistication of intelligence gathering devices (e.g., cameras, sensors, radar) on modern aircrafts and UAVs, there is a continued need for increased computational performance. In the size, energy, power, and weight constrained platforms operated by the Air Force, it is of utmost importance to perform computation and data analysis in the most efficient manner possible. In this work, we are addressing this challenge through the development of new power delivery architecture that is inherently scalable to massive number of cores, and low-power future data centers.

Over the last three years, we have developed key control algorithms, hardware validation platform, and system level hardware/software interfaces to demonstrate ultra-efficient and lightweight power delivery for next-generation computing. We have demonstrated – for the first time – how the co-design and implementation of both the hardware and software can yield signifcant energy savings in data centers. Whereas conventional systems operate around 90% efficiency in power delivery, we have demonstrated 98% power delivery efficiency, for realistic compute loads and hot-swapping capable hardware. A highlight of the project was the development of a control algorithm, along with hardware verification, of the first series-to-bus DPP power delivery architecture for datacenters. Moreover, we derived the control limitations of the proposed approach, along with the development of an Intel Next Generation of Computing (NUC) based micro-server test-bed platform. In addition to the development of the hardware platform for power conversion, we have extended our integrated test-bed with improved measurement techniques, as well as scalable software test scenarios for various computational tasks.

Additionally, custom Linux kernel software has been development to enable fine-grain control of individual core activity, important both for characterizing the power consumption for different task, and to improve the overall power delivery system. Several benchmarks on various software routines have been performed, with very promising result.

I. Introduction

With the prevalence of Internet as well as the increasing demands for cloud computing, data centers - the host of these services - are becoming more and more important. As the number and size of data centers continue to grow, so does the electrical power that they consume. In 2014[1], data centers in the U.S. had an estimated electricity consumption of 70 billion kWh, which is approximately 1.8% of the total national electricity consumption that year [**?**]. With such a large energy usage, improving the energy efficiency of data centers has been a focus of research interest. A key component in data centers is the server, which operates at low DC voltage, typically 12V or 48V. Thus, a power delivery architecture is required to convert the utility AC voltage to low DC voltage to provide power for the servers. Increasing the efficiency of this power delivery architecture is a very crucial step in making data centers more energy efficient [**?**], [**?**], [**?**].

Current power delivery architectures for servers consist of cascaded power stages. As a typical example, a central rectifier draws from the utility AC voltage to regulate a DC bus voltage of 48V. Then for each server, a DC-DC converter steps down the 48V to the server's nominal input 12V voltage and supply its power [**?**]. Each power conversion stage process the full server power in this process. The system energy efficiency is directly limited by the efficiency of each power stage.

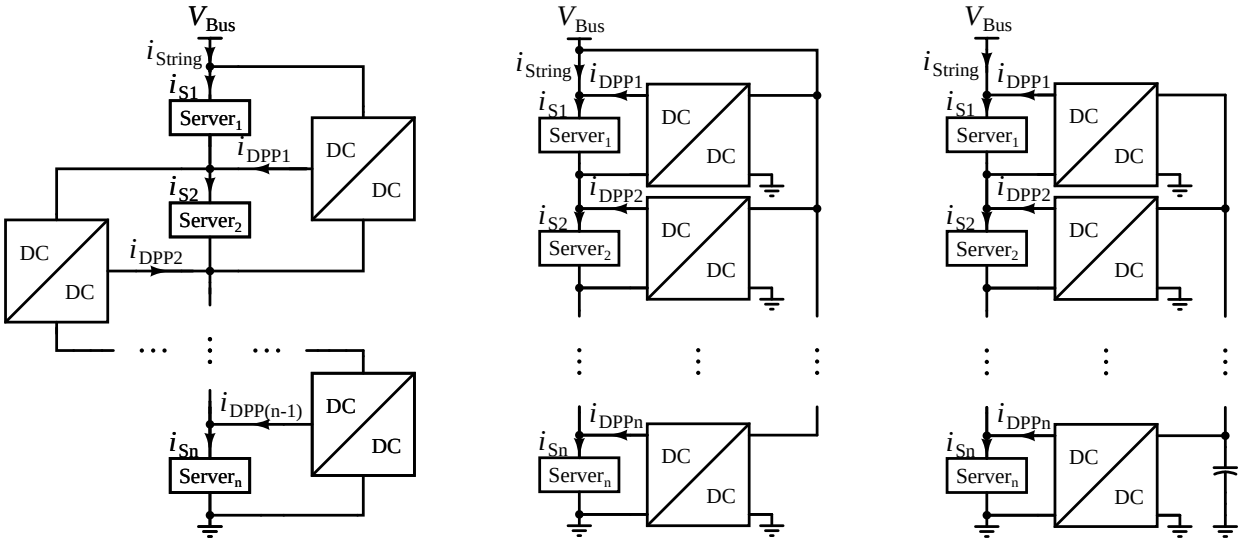
The series-stacked power delivery architectures are proposed to address the limitations of the conventional architectures, and can achieve much higher power delivery efficiency [**?**]. In the series-stacked architectures, the servers are connected in series, and differential power processing (DPP) converters are used to compensate for the mismatch in the stacked servers' currents. Therefore, the bulk power consumed by the servers flows through the series-stack without being processed by the power conversion stage, and only the difference in power between servers is processed by the DPP converters. Using this technique, the amount of power processed, and the corresponding power loss, can be greatly reduced compared to conventional architectures, resulting in extremely high efficiency. There are three basic ways to connect the DPP converters, leading to three basic types of series-stacked architecture: the server-to-server type, the server-to-bus type and server-to-virtual-bus type, which are depicted in Fig. 1. The first type has been explored in [**?**], [**?**]; the third type has been studied in [**?**], [**?**]; the second type has not been investigated in depth experimentally before, and is the focus of this work. In [**?**] the hot-swapping operation of the servers are addressed in detail. There are also hybrid architectures combining the above mentioned three types, as proposed in [**?**]. The DPP idea also applies to series-connected photovoltaic (PV) cells, where the first, second and third types of series-stacked PV cell architecture are investigated in [**?**], [**?**] and [**?**], respectively. There are also variations of the three basic types for PV cells [**?**]. In this work, we focus on investigating the server-to-bus type of series-stacked architecture for data center servers.

The server-to-bus type series-stacked architecture has unique properties in comparison with other types. Two important properties of this type of architecture are that it is able to achieve the minimum power processed in the DPP converters, and that it has an inherent redundancy in its DPP converters that provides a high level of reliability. In this paper, both properties are demonstrated experimentally. The server-to-bus architecture is implemented with four 4-to-1 dual active bridge (DAB) converters employing GaN switches, which constitute the DPP converters. Four real Dell computers are used as the servers. The power delivery architecture is validated in both the normal operation of the servers, as well as the hot-swapping operations

[1]I tried to find more recent statistics, but could not find any.

- an operation scenario which is very important for the reliability of data centers. Very high power conversion efficiency of 99.04% is achieved. Moreover, for the first time reported in literature[2], this work presents that the series-stacked architecture can supply servers which are executing real data center Hadoop computation task.

The remainder of this paper is organized as follows. Section II discusses the unique features of the server-to-bus power delivery architecture. In Section III, control methods for the DPP converters in the server-to-bus architecture are discussed, including how to realize the minimum power processed and how to handle hot-swapping operation. Section IV presents the hardware experimental implementation of the architecture as well as experiment results that verify the reliability property, realize the optimal control and show that real-life computation tasks such as Hadoop can be run using the series-stacked power delivery architecture. Section V concludes the paper.



(a) Server-to-server type. (b) Server-to-bus type. (c) Server-to-virtual bus type.

Fig. 1. Three basic types of series-stacked power delivery architecture.

II. Server-to-Bus Series-Stacked Architecture

The server-to-bus architecture has some desirable features that are not possessed by other series-stacked architectures. Firstly, it can achieve the minimum total processed power in the DPP converters, and thus obtain the highest possible theoretical efficiency in power delivery [**?**]. Secondly, this architecture can tolerate one DPP converter failure while still delivering power to all the servers in the series stack. Thus there is inherently one extra redundancy in this power delivery system, resulting in a high level of reliability. These two points are explained below in greater detail.

A. Minimum Total Processed Power

The server-to-bus architecture can achieve the minimum total processed power in converters for most server load distributions in the series-stack. The other two series-stacked architectures cannot guarantee minimum total processed power in all server load distributions. To better understand this, let us assume that in a certain scenario, the servers are drawing current i_{S1}, i_{S2}, $\ldots$, i_{Sn} as shown in Fig. 1.

[2]can we say so?

For the server-to-virtual-bus and server-to-server series-stacked architecture, each DPP converter's output current i_{DPPk} is uniquely determined (there is only one feasible combination of the operating points of the DPP converters). The server-to-bus series-stacked architecture, however, is an under-determined system, where the DPP converters' output currents have infinite feasible combinations. In fact, there are n limiting variables (i_{S1} through i_{Sn}), but $(n+1)$ controllable variables (i_{DPP1} through i_{DPPn}, plus i_{string}), so there is one extra degree of freedom. For this architecture, string current i_{string} can be set to any value as long as the DPP converters' output currents (i_{DPP1} through i_{DPPn}) are controlled correspondingly. Furthermore, some i_{string} values will yield the combinations of i_{DPP1} through i_{DPPn} with smaller total processed power in the DPP converters than other combinations. We can thus control i_{string} to be the optimal value(s) that results in the minimum total processed power in the converters of the server-to-bus series-stacked architecture. The following analysis will derive the value of the optimal string current in the server-to-bus architecture.

Let us assume, without loss of generality, that the server currents, as shown in Fig. 1b, have the relationship

$$i_{S1} \leq i_{S2} \leq \cdots \leq i_{Sn}. \tag{1}$$

Moreover, for simplicity of analysis, assume 100% efficient converters. Then, the power processed in the k-th DPP converter P_{DPPk} is

$$P_{DPPk} = |i_{DPPk}| \cdot v_{Sk}, \tag{2}$$

where v_{Sk} refers to the k-th server voltage. The objective to minimize, the total processed power in all the DPP converters $P_{DPP,tot}$, is

$$P_{DPP,tot} = \sum_{k=1}^{n} P_{DPPk}. \tag{3}$$

If it is further assumed that the server voltages are all well regulated around the nominal input voltage, and thus

$$v_{S1} \approx v_{S2} \approx \cdots \approx v_{Sn} \approx V_{nominal}. \tag{4}$$

Then

$$P_{DPP,tot} \approx V_{nominal} \sum_{k=1}^{n} |i_{DPPk}|. \tag{5}$$

The new objective function to be minimized is then just the summation part $\sum_{k=1}^{n} |i_{DPPk}|$, which is defined as $f(i_{string})$,

$$f(i_{string}) = \sum_{k=1}^{n} |i_{DPPk}| = \sum_{k=1}^{n} |i_{Sk} - i_{string}|. \tag{6}$$

To determine the optimal i_{string} value that will minimize the objective function in (6), we will first simplify this expression and then look at its derivative. Suppose there are m servers ($0 \leq m \leq n$) whose currents are smaller than i_{string}. Due to the

assumption in (1), these servers are just server 1 to server m. Then $f(i_{string})$ can be simplified as

$$f(i_{string}) = \sum_{k=1}^{m}(i_{string} - i_{Sk}) + \sum_{k=m+1}^{n}(i_{Sk} - i_{string}), \tag{7}$$

and the derivative is

$$f'(i_{string}) = m - (n - m) = 2m - n. \tag{8}$$

To better understand how $f(i_{string})$ changes with i_{string}, two example cases with n being even ($n = 6$) or odd ($n = 7$) are considered, with the server currents in both cases listed in Tables I and II. Example plots for $f(i_{string})$, m and $2m - n$ in the two example cases are shown in Fig. 2 and Fig. 3. It can be seen from the figures that in both cases, as i_{string} increases up from 0, m increases monotonically from 0 to n, and $2m - n$, which is also $f'(i_{string})$, increases monotonically from $(-n)$ to n. Moreover, the objective function $f(i_{string})$ first decreases and then increases, and reaches the minimum (shown in red in the plots) when $(2m - n)$ crosses 0. When $n = 6$, the minimum happens when $m = 3$, when $i_{S3} \leq i_{string} \leq i_{S4}$, or when i_{string} is larger than the currents of three (half of total) servers. In this case, there is a region of i_{string} values that are all optimal. When $n = 7$, the minimum happens when $i_{string} = i_{S4}$, or when i_{string} equals to the server current right at the middle (the 'median'). In general, when n is even, the optimal string current $i_{string,opt}$ is any value between $i_{S\frac{n}{2}}$ and $i_{S(\frac{n}{2}+1)}$, and when n is odd, the optimal string current $i_{string,opt}$ is equal to $i_{S\frac{n+1}{2}}$, as expressed in Eqn. 9. For a more detailed derivation of the optimal string current, readers can refer to [**?**].

It needs to be pointed out that the currents i_{Sk}, i_{string} and voltages v_{Sk} are actually considered in terms of their averaged values in the analysis above. Due to the switching actions or operations in hysteresis mode of the power converters as well as the fast-changing nature of computation loads in servers, the instantaneous values of these currents and voltages may fluctuate or ripple. These ripples are assumed to be small compared with the average values of these quantities within each control time period, so that their effects are neglected.

$$i_{string,opt} = \begin{cases} i_{S\frac{n+1}{2}}, & \text{if } n \text{ is odd} \\ \text{any value} \in [i_{S\frac{n}{2}}, i_{S(\frac{n}{2}+1)}], & \text{if } n \text{ is even} \end{cases} \tag{9}$$

TABLE I
SERVER CURRENTS IN THE EXAMPLE WHERE N IS EVEN

i_{S1} [A]	i_{S2} [A]	i_{S3} [A]	i_{S4} [A]	i_{S5} [A]	i_{S6} [A]
5	6	7	8	9	10

B. 'Redundancy' property for a high level of reliability

Another feature of the server-to-bus architecture is that it offers high reliability/availability compared with other series-stacked architectures. Since the server-to-bus architecture has one extra degree of freedom (one more control handle than quantities to

TABLE II
SERVER CURRENTS IN THE EXAMPLE WHERE N IS ODD

i_{S1} [A]	i_{S2} [A]	i_{S3} [A]	i_{S4} [A]	i_{S5} [A]	i_{S6} [A]	i_{S7} [A]
5	6	7	8	9	10	11

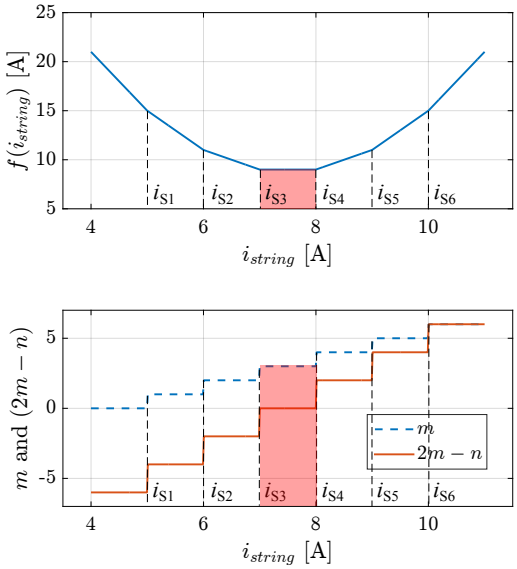


Fig. 2. The $f(i_{string})$, m and $2m-n$ in the example where n is even.

be controlled), it can actually deliver power to n servers with only $(n-1)$ DPP converters. In other words, it can tolerate one converter failure, and can still deliver power to all stacked servers.

For example, if the DPP converter corresponding to the k-th server fails in the server-to-bus architecture, we can control the string current i_{string} to be the k-th server's current (i_{Sk}), and control the other DPP converters to inject or reject currents equal to the difference between this string current and the corresponding server currents [**?**]. This is experimentally verified in Section IV. For the other two series-stacked architectures, in the case of a single failure in the DPP converters, the power delivery can no longer be fulfilled.

As was quantitatively analyzed in detail in [**?**], due to this redundancy property as well as the reduced power processed in the converters of the server-to-bus series-stacked architecture, the reliability (characterized by down time over a 10 year time period) of this architecture is comparable to the reliability standard of a conventional power delivery architecture.

III. CONTROL OF SERVER-TO-BUS ARCHITECTURE FOR BOTH NORMAL AND HOT-SWAPPING OPERATIONS

Control is essential for running a series-stacked architecture, we need to control the DPP converters properly to maintain the server voltages within the allowed band. This section discusses the control method used for the server-to-bus architecture. Firstly, a control method that only requires voltage sensing is described. This method focuses on the voltage regulation of the stacked servers, which means the string current is not always optimized to achieve minimum power processed in the system. A second control method is discussed later in this section that incorporates current sensing, and can control the string current to be optimal, and minimizes the total processed power. In special operation situations of servers, like the hot-swapping operation,

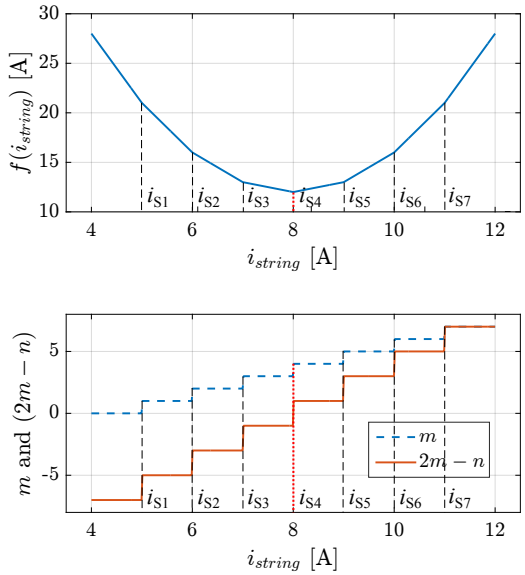


Fig. 3. The $f(i_{string})$, m and $2m-n$ in the example where n is odd.

the mismatch of server currents in the stack can be large. The control method for the series-stacked architecture should be able to handle this, which also is explained in this section.

A. Bidirectional Hysteresis Control with only Voltage Feedback

This control method, the bidirectional hysteresis control with only voltage sensing, originates from [**?**], where a similar control method successfully achieved voltage regulation of stacked servers for the server-to-virtual-bus type of series-stacked architecture. Based on the hysteresis shape shown in Fig. 4, the basic idea of the control method is: each server voltage is sampled and an error is calculated with respect to the reference voltage value. Then this error is compared to predefined hysteresis bands, ε_0, ε_1 and ε_2. Depending on this comparison, each DPP converter is decided to be on or off, and if it is on, the direction of the current flow (current injection/rejection), and the magnitude of current flow (light/full) are also determined. Therefore, for example when the server voltage is too low, the DPP converter inject current into it to raise the voltage, until when the voltage is within allowed range from the reference, then the converter is turned off. The different magnitudes of current flow, light or full, is designed for the normal or hot-swapping operations of servers, respectively. The light magnitude inject/reject $\sim$3.5 A to/from server, and is used for when the server currents are relatively balanced (normal operation). The full magnitude inject/reject $\sim$12 A to/from server, and is used for when there is large mismatch between the server currents (e.g. hot-swapping operation). A detailed explanation of the control algorithm can be found in [**?**].

B. Optimal String Current Control

As was discussed in Section II, only $(n-1)$ DPP converters are needed to keep the n server voltages regulated in the server-to-bus architecture. The control for optimal string current makes use of this feature. To control i_{string} to be the optimal value as shown in Eqn. 9, we can control the string current just to follow the $(\frac{n+1}{2})$-th largest server current if n is odd, or the $(\frac{n}{2}-1)$-th largest server current if n is even. To control the string current to follow a specific server's current, we can

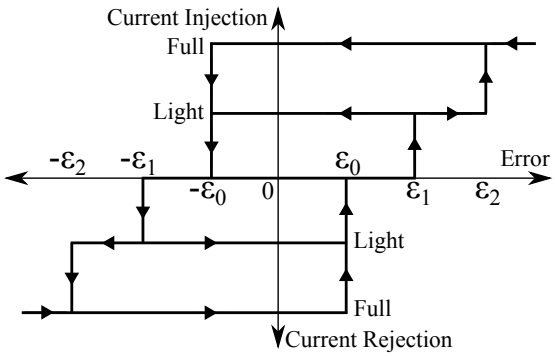


Fig. 4. Bi-directional hysteresis shape [**?**].

turn off the single DPP converter corresponding to that specific server, and keep all other DPP converters operating as normal with the voltage hysteresis control stated in Section III-A. In this control method, the server currents are measured, and then compared to determine which server consumes the $(\frac{n+1}{2})$-th largest current (if n is odd, or the $(\frac{n}{2}-1)$-th largest if n is even). Then the corresponding DPP converter is temporarily turned off. In this manner, the string current is controlled to be optimal in average, and the power processed in the DPP converters are minimized. In this work, a four-server system is investigated. Thus the optimal string current is any value between the second and third largest server current. In the optimal string current control method used in the experiments, the string current is controlled to follow the second largest server current.

IV. Experimental Work

A. Prototype DPP Hardware with combined DPP converter and Hot-Swapping interface

Figure 5 shows the schematic of the prototype DPP hardware for the server-to-bus architecture. The hardware consists of three parts: the DPP converter, the stack initialization circuitry and the hot-swapping circuitry. An annotated photograph of the prototype hardware is shown in Figure 6. The key components are listed in Table III.

The DPP converter in the server-to-bus architecture is required to be bidirectional and isolated, with the primary side rated at 48V and the secondary side at 12V. The dual active bridge (DAB) topology is chosen, due to its symmetrical design and simple modulation [**?**]. GaN transistors are used in both the primary and secondary sides, and the efficiency vs output power plot of the converter is shown in Figure 7.

The stack initialization circuitry is used to achieve the voltage balancing between the series-stacked hardware prototypes when the DC bus is first applied to the series-stack. The hot-swapping circuitry provides complete isolation when the server is swapped out from the series-stack, and also limits the in-rush current caused by charging up the large input capacitor of the server when it is swapped in. Hot-swapping and initialization circuitries follow the principles explained in [**?**], which can also provide further details for interested readers.

B. Testbed

A testbed for the server-to-bus DPP power delivery architecture is developed with four Dell Optiplex SX280 servers running Linux operating systems. Each server has a single 12V motherboard input, and the DC bus voltage for the four series-stacked system is 48V, which is provided by a DC power supply (HP 6674A). The voltages and currents data in the system are sampled

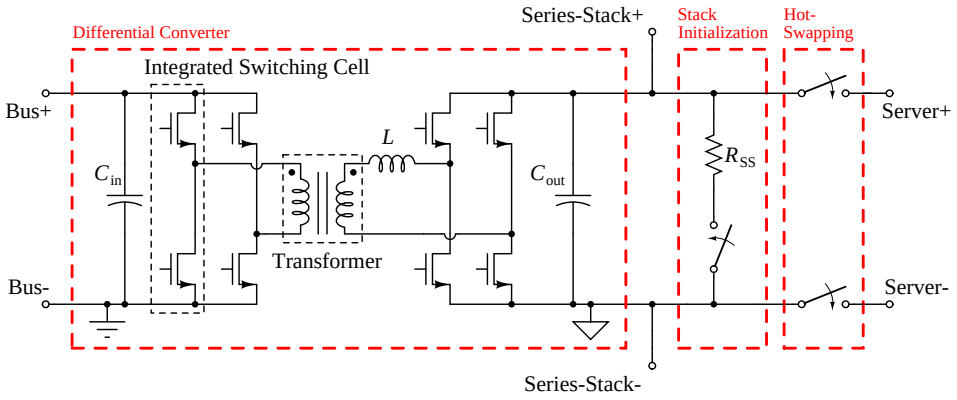


Fig. 5. The simplified schematic of the prototype DPP hardware.

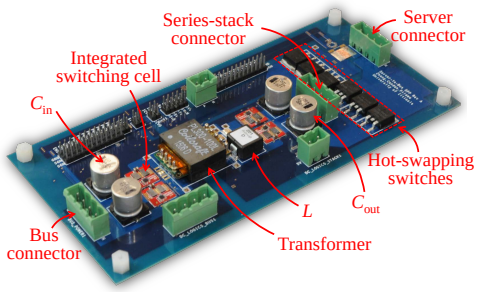


Fig. 6. The prototype DPP hardware.

by data acquisition unit (DAQ) from National Instruments at 5000 samples per second. The server currents, string current and DPP current are converted to voltages with sense resistors and amplifiers (which are calibrated with an Agilent 34461A 6 1/2 digit multimeter) for the DAQ to sample. The DPP current measured are those on the bus side, shown as $i_D PPk, pri$ in Figure 8. The control algorithm for the four DPP converters is implemented on a single off-board microcontroller (TI C2000 Piccolo F28069). Figure 8 and 9 show the schematic drawing and photo of the testbed respectively.

C. Server Computation Tasks

The servers' computational task in the first four of our power delivery experiments is represented by the standard Linux stress utility [**?**]. In our fifth experiment, to test our system in a real data center distributed computational environment, we use Hadoop framework [**?**] to utilize multiple nodes. Our work employing this real data center computation task on series-stack architecture experiments is the first reported in literature, to the best knowledge of the authors[3].

Hadoop is a map-reduce framework [**?**] that its computation model includes two steps. First, the Hadoop workload manager divided the input data into numerous independent chunks feeds each into a map task to produce the intermediate result. Second, the map task computation result is sorted and given to reduce tasks, which returns the final result.

We used the grep application to evaluate our design, in which the application extracts and counts a specific string in the dataset. The extracted strings are then sorted out, and the total number of matches is reported. The dataset used in our experiment is 2 GB in size. The four Dell servers are the four slave nodes of the Hadoop computational task, and an additional server is

[3]can we say so?

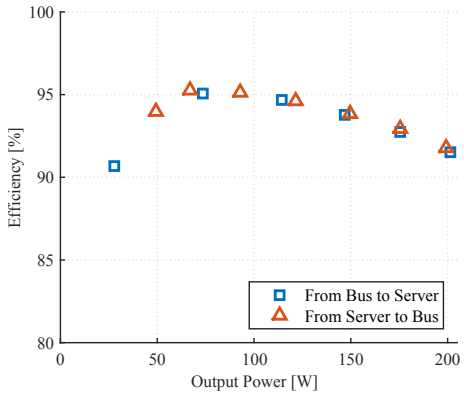


Fig. 7. The efficiency plot of the GaN-based DAB converter.

TABLE III
THE KEY COMPONENTS OF THE DPP CONVERTER

GaN Switches	EPC 2032 (Bus side)
	EPC 2030 (Server side)
GaN Gate Drivers	TI LM5113
Transformer	Coilcraft PL300-100L, 8:2 turns
Inductor	Coilcraft SLC1480-201ML, 200nH
C_{in}	ceramics $8 \times 1\mu$F, TDK
	aluminum $2 \times 68\mu$F SMD, Panasonic
C_{out}	ceramics $8 \times 4.7\mu$F, TDK
	aluminum 2×1mF SMD, Panasonic
Hot-Swapping Switches	Infineon IRLS4030PbF (three in parallel)

used as the master node, which assigns tasks to the four slave servers.

D. Experimental Results

Five experiments were conducted to validate the functionality and above mentioned properties of the server-to-bus power delivery architecture. Experiment 1 presents the entire processes when servers boot up, operate and shut down. Experiment 2 shows that the power delivery system can tolerate one converter failure. In experiment 3, the hot-swapping operation is demonstrated. The optimal string current control is carried out in experiment 4. Lastly, in experiment 5, the scenario when the servers are running real-life data center Hadoop computation is presented.

1) Boot-up, Operation and Shut-down of Servers: The first experiment shows that the server-to-bus power delivery system developed in this work can successfully supply power for data center servers during the entire processes of boot-up, normal operation and shut-down. The detailed current and voltage waveforms during the experiment are shown in Fig. 10, including all server currents and voltages, the DC bus current and the string current. As can be seen, the experiment is 450 seconds in total, and is broken down to five intervals depending on different server load scenarios. Based on the recorded voltage and current data, the input power to the system P_{in} is calculated by multiplying the DC bus voltage and current. The output power to each server is calculated by multiplying its voltage and current, and the total output power P_{out} is the sum of the four server powers. The power loss P_{loss} is the difference between P_{in} and P_{out}.

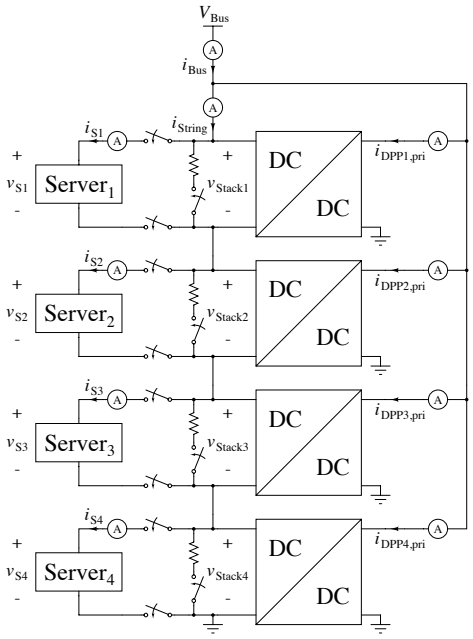


Fig. 8. The schematic of the testbed for the Server-to-Bus architecture.

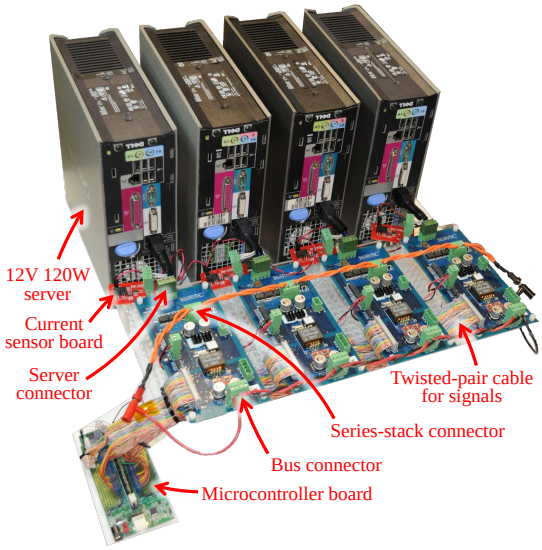


Fig. 9. The photo of the testbed for the Server-to-Bus architecture.

$$P_{loss} = P_{in} - P_{out}. \tag{10}$$

There are various sources for the power loss in the system. Firstly, there is the power losses from the DPP converters, $P_{loss,conv}$. Apart from this, there are power loss from the current sensing resistors, $P_{loss,meas}$; from system wiring, $P_{loss,wiring}$; and from the hot-swapping switches, $P_{loss,HS}$. Thus we have,

$$P_{loss} = P_{loss,conv} + P_{loss,meas} + P_{loss,wiring} + P_{loss,HS}. \tag{11}$$

Based on these understandings, two types of power delivery efficiencies are calculated: One is system-level efficiency, η_{sys},

where the loss from current sensing resistors is excluded,

$$\eta_{sys} = \frac{P_{out}}{P_{in} - P_{loss,meas}}. \tag{12}$$

The other one is power conversion efficiency, η_{conv}, where the power losses from current sensing resistors, system wiring and the hot-swapping switches are excluded,

$$\eta_{conv} = \frac{P_{out}}{P_{in} - P_{loss,meas} - P_{loss,wiring} - P_{loss,HS}}. \tag{13}$$

A similar calculation is explained in detail in [**?**]. A breakdown of the average input and output powers, the average power loss, the system-level efficiency η_{sys} and power conversion efficiency η_{conv} is given in Table IV, and the average current of each server during each interval is also listed at the bottom of the table.

In the first 120-second time interval, the servers boot up. It can be noticed that even though the currents drawn by the servers is fast-changing and has large dynamics during boot-up, the series-stacked architecture can keep the server voltages well regulated at 12 V and supply the server power. The system-level efficiency is calculated as 96.05% and the power conversion efficiency is calculated as 96.92%. The series-stacked system has not yet achieved it highest efficiency, since the server loads are not so balanced during this period. In the next time interval, the Linux stress computation task is run on the all four servers, where their power consumptions all reach maximum. The server loads in the series stack are very balanced, and high efficiencies are achieved here, with a power conversion efficiency of 98.84% and a system-level efficiency of 97.67%. In Interval 3, the servers already finished the stress task, and are all in idle for 30 s, with low power consumption. In the following 120 s, server 1,2 and 4 are stressed, but server 3 are kept idle. This represent a scenario where the server loads are not balanced. The power conversion and system-level efficiencies are 97.43% and 96.31%, respectively. In the last interval, the servers shut down. In the full 450 s period, the system-level efficiency and power conversion efficiency are calculated as 96.70% and 97.74%, respectively. In the entire experiment, the voltage hysteresis control method is used on all four DPP converters. As can be seen in Fig. 10, all server voltages are regulated to their nominal 12 V throughout the experiment, demonstrating good regulation capability of the server-to-bus series-stacked architecture.

2) Tolerance of one converter failure: The second experiment shows that the server-to-bus architecture can tolerate one DPP converter failure while still delivering power to all servers in the series stack. The current and voltage waveforms of the experiment are shown in Fig. 11. The breakdown of the average input and output powers, the average power loss, the system-level efficiency η_{sys} and power conversion efficiency η_{conv} is given in Table V. This 90-second experiment can be broken down to three intervals. In the entire experiment, all four servers are executing the Linux stress computation task. In intervals 1 and 3, all four DPP converters are running the voltage hysteresis control, whereas in interval 2, one DPP converter, converter 2, is simply turned off, to represent one converter failure. The other three converters are still running the voltage hysteresis control normally.

Fig. 12 shows the DPP current in the series-stacked system in experiment 2. The DPP current, denoted as $i_{DPPk,pri}$ is measured on the primary side (bus side) of the DPP converter, and there is roughly a 4x relationship between it and the

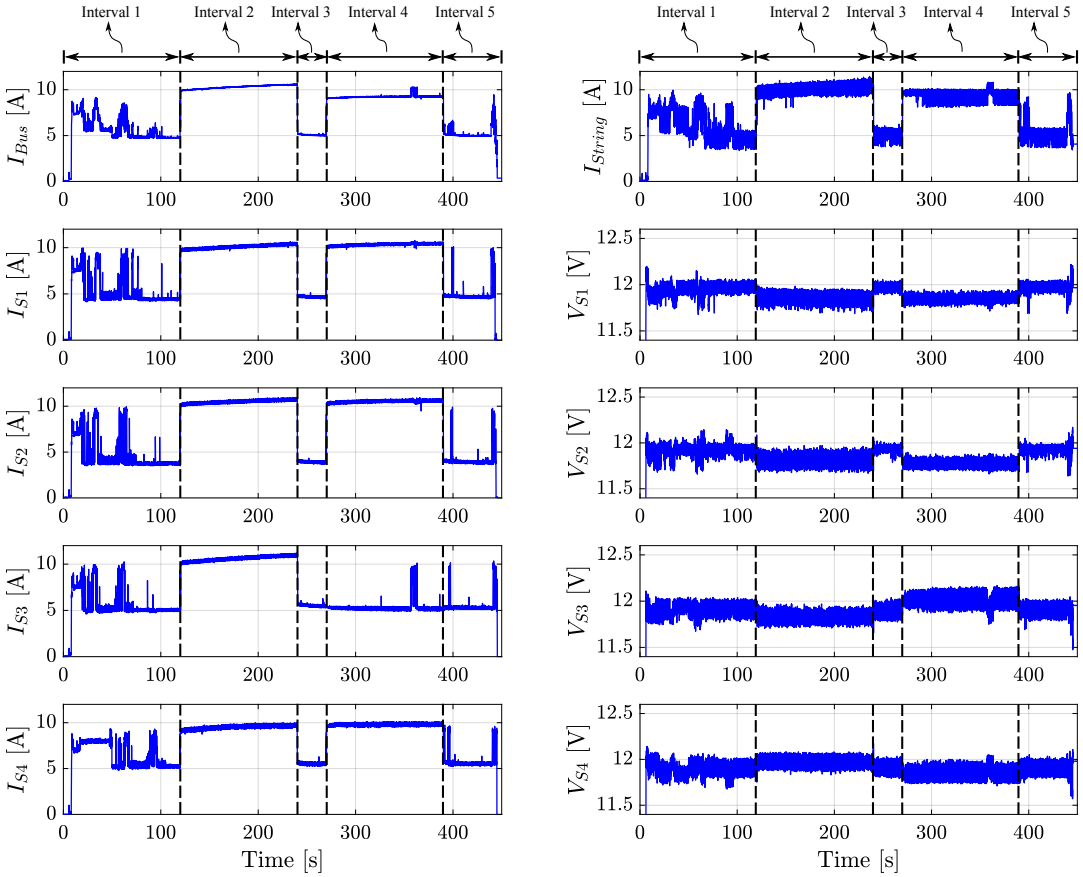


Fig. 10. Current and voltage waveforms in experiment 1. (Measured data is 10 ms window averaged for better illustration of the entire experiment on a single plot.)

TABLE IV
BREAKDOWN OF THE AVERAGE INPUT AND OUTPUT POWERS, THE AVERAGE POWER LOSS, THE SYSTEM-LEVEL EFFICIENCY AND POWER CONVERSION EFFICIENCY DURING EXPERIMENT 1

Time Interval [s]	Interval 1 0 - 120	Interval 2 120 - 240	Interval 3 240 - 270	Interval 4 270 - 390	Interval 5 390 - 450	Overall 0 - 450
$< P_{in} > \ [W]$	256.91	494.41	245.85	444.05	232.07	366.10
$< P_{out} > \ [W]$	246.27	481.37	237.01	426.40	221.46	353.07
$< P_{loss} > \ [W]$	10.64	13.04	8.84	17.65	10.61	13.03
η_{sys} [%]	96.05%	97.67%	96.56%	96.31%	95.60%	96.70%
η_{conv} [%]	96.92%	98.84%	97.28%	97.43%	96.35%	97.74%
$< i_{S1} > \ [A]$	4.99	10.08	4.75	10.33	4.44	7.68
$< i_{S2} > \ [A]$	4.33	10.49	4.00	10.53	3.79	7.53
$< i_{S3} > \ [A]$	5.34	10.61	5.59	5.41	5.07	6.74
$< i_{S4} > \ [A]$	6.06	9.51	5.55	9.80	5.31	7.85
$< i_{string} > \ [A]$	5.72	10.10	5.49	9.60	5.47	7.87

secondary side (server side) DPP current, denoted as i_{DPPk}. A zoomed-in view of the highlighted region in Fig. 12 is shown in Fig. 13, which are the detailed DPP current profiles around $t = 40$ s of interval 2. As can be read from the figures, in intervals 1 and 3, all four DPP converters are operating with voltage hysteresis control the DPP converters execute light current injection, light current rejection, or no action. In interval 2, however, converter 2 is turned off entirely it is configured to stay in no action mode for 60 s. This is effectively the same with when converter 2 fails open. During this interval, only

three converters can inject or reject differential currents. As can be seen in Fig. 11, all four server voltages are kept regulated at 12 V during interval 2. The voltage fluctuation of server 2, the server without its corresponding DPP converter, is larger than in intervals 1 and 3 by a moderate amount as expected, but it is still safely within the allowed band. In interval 3, when converter 2 is put back into voltage hysteresis control, the behavior of the series-stacked system got back to basically the same as in interval 1.

As listed in Table V, the system-level efficiency and power conversion efficiency are very high in all three intervals, because the server load are very balanced. The power conversion efficiency in interval 2, when only three converters are processing power, reaches the highest 99.04%.

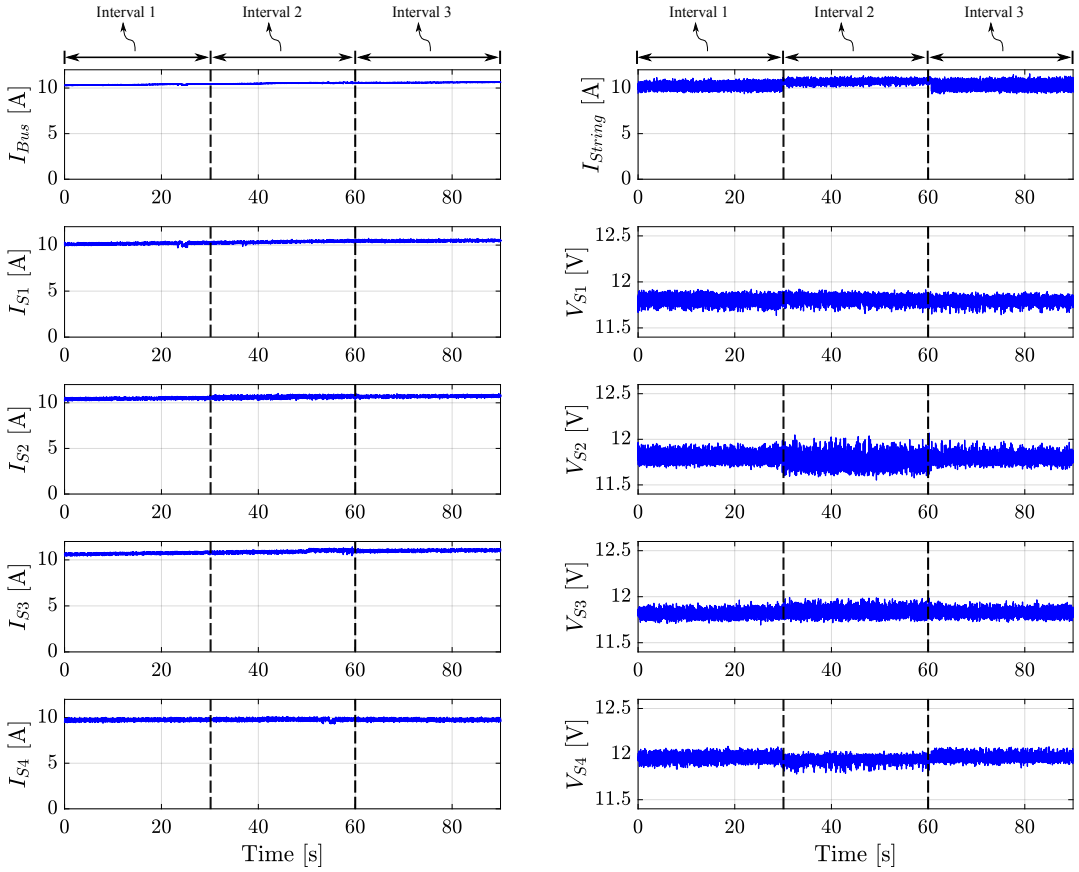


Fig. 11. Current and voltage waveforms in experiment 2. (Measured data is 10 ms window averaged for better illustration of the entire experiment on a single plot.)

TABLE V
BREAKDOWN OF THE AVERAGE INPUT AND OUTPUT POWERS, THE AVERAGE POWER LOSS, THE SYSTEM-LEVEL EFFICIENCY AND POWER CONVERSION EFFICIENCY DURING EXPERIMENT 2

Time Interval [s]	Interval 1 0-30	Interval 2 30-60	Interval 3 60-90	Overall 0-90
$< P_{in} >$ $[W]$	498.12	505.50	509.51	504.38
$< P_{out} >$ $[W]$	485.08	492.16	495.50	490.91
$< P_{loss} >$ $[W]$	13.04	13.35	14.01	13.47
η_{sys} [%]	97.69%	97.68%	97.57%	97.65%
η_{conv} [%]	98.98%	99.04%	98.87%	98.96%

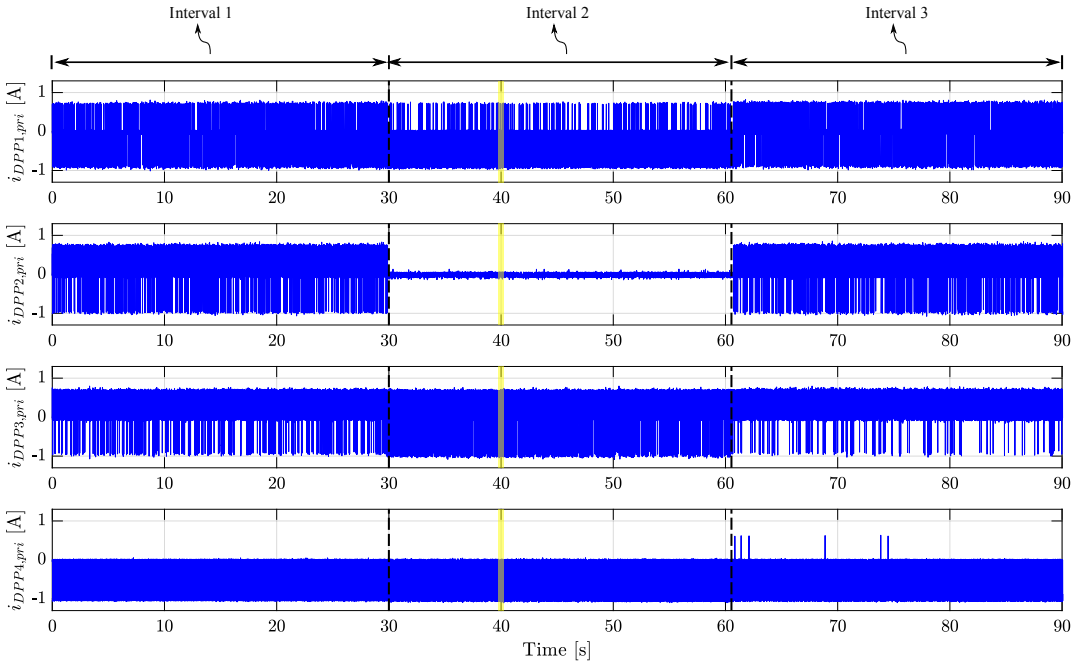


Fig. 12. Measured DPP currents in experiment 2.

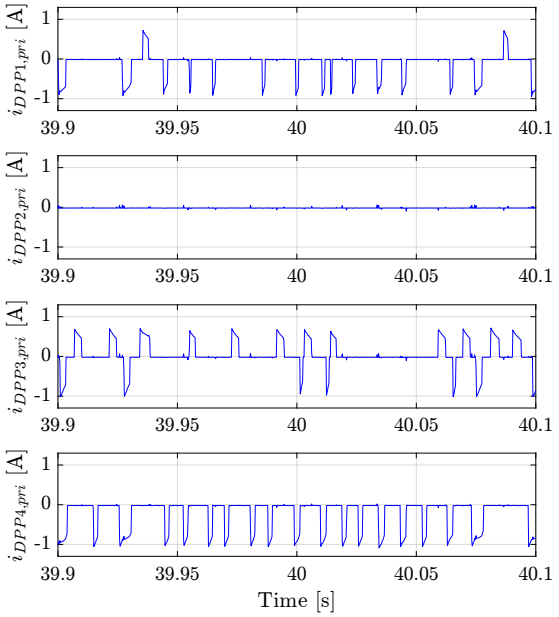


Fig. 13. A zoomed-in view of the measured DPP currents in experiment 2.

3) Hot-Swapping Operation: Experiment 3 shows the hot-swapping operation of the series-stacked server in server-to-bus architecture. The current and voltage waveforms are shown in Fig. 14, and the breakdown of power and efficiency results is given in Table VI. In the entire experiment, the bidirectional hysteresis control with only voltage feedback is used.

As can be seen from Fig. 14, all four servers are running the stress computation task in interval 1. Then at the time around 149 s, server 2 is swapped out, while the other servers remain doing the stress task. The current of server 2, i_{S2} drops to zero instantly, while the other server currents remain basically unchanged. There is a large mismatch between the server currents,

and the DPP converters enter the full current injection/rejection state to regulate the voltages, as can be seen in the measured DPP current waveforms at the swap-out moment shown in Fig. 15. Although V_{S2} drops to zero, the stack voltage at the DPP converter corresponding to server 2, V_{SS2} remains 12 V.[4] During this interval, since the mismatch in server current and thus the processed power in the DPP converters are large, the power conversion efficiency is 95.45%, not as high as in other scenarios. Server 2 is swapped out for 60 s, before it is swapped back in in interval 3 at 209-th s. It can be noticed in Fig. 14 that server 2, after swap-in, start to boot up normally. During all the fast changing currents of server 2, all the server voltages are well regulated at 12 V in the series stack. In interval 4, server 2 goes back to doing stress computation. The waveforms for DPP current during the swap-in moment of server 2 is shown in Fig. 16. DPP converter 2 still remains in full current rejection mode after swap-in, because when server 2 just start to boot up, there is still a relatively large mismatch between its current and the currents of the other servers doing stress computation.

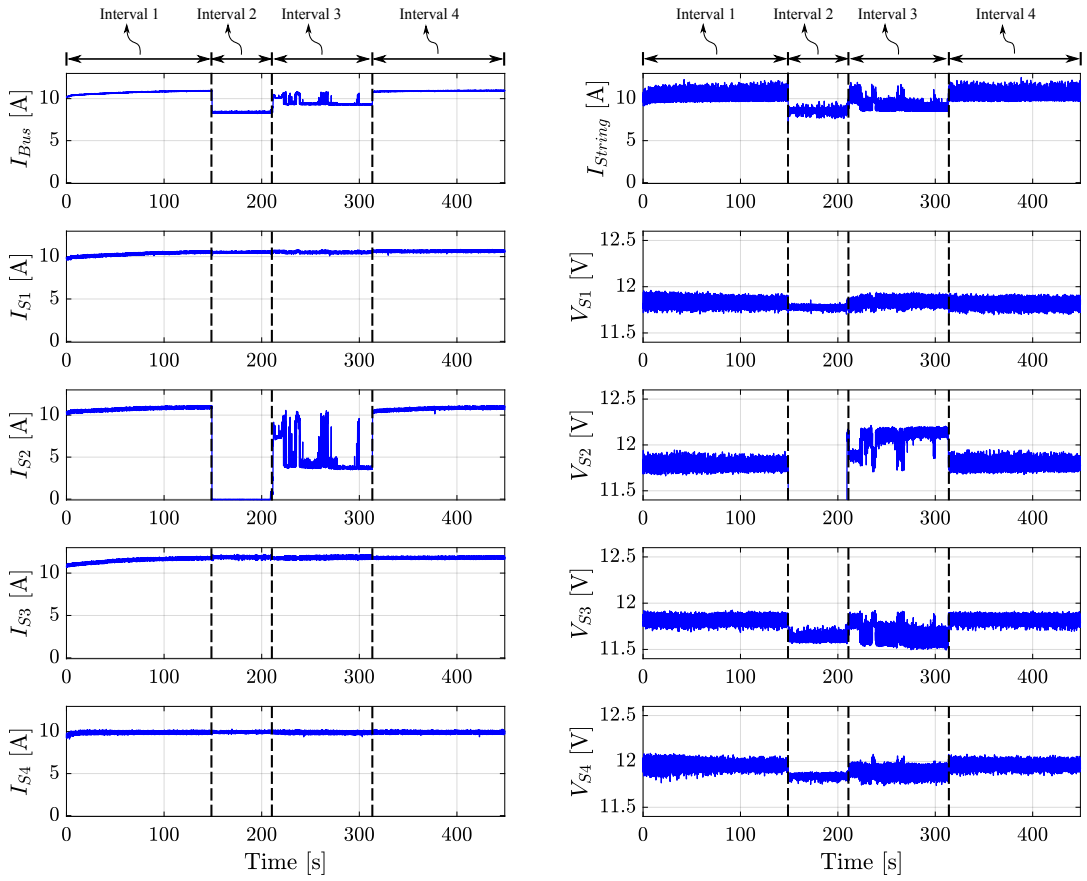


Fig. 14. Current and voltage waveforms in experiment 3. (Measured data is 10 ms window averaged for better illustration of the entire experiment on a single plot.)

4) Optimal String Current Control: In experiment 4, the optimal string current control is tested. The current and voltage waveforms are shown in Fig. 17, and the breakdown of power and efficiency results, including the detailed power loss information is given in Table VII.

Throughout the experiment, the servers 2, 3 and 4 are doing the stress computation, while server 1 is kept idle. In this scenario, three servers are consuming ∼10 A, and one server is consuming ∼4.4 A. According to Eqn. 9, the optimal string

[4]I feel we may also need to show the plots with both V_{S2} and V_{SS2} on it at the swap-out and swap-in moments. I will make these plots.

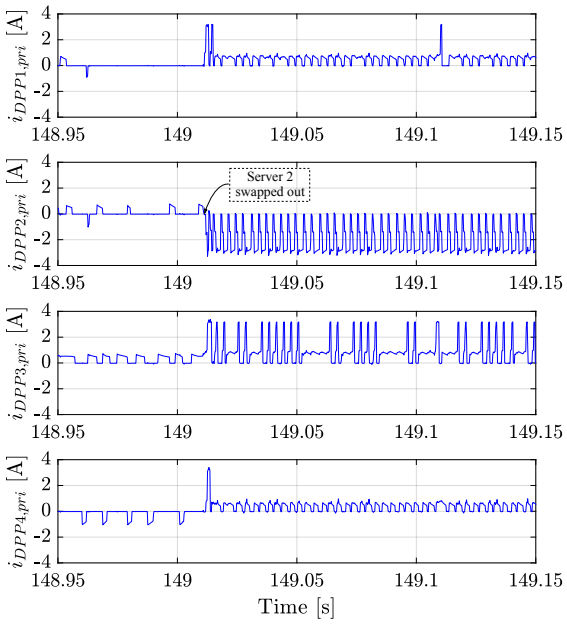


Fig. 15. Measured DPP currents at the swap-out moment of server 2 in experiment 3.

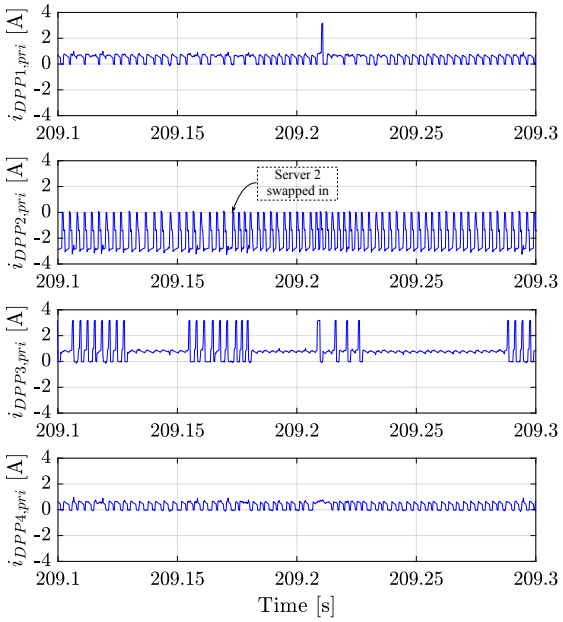


Fig. 16. Measured DPP currents at the swap-in moment of server 2 in experiment 3.

TABLE VI
BREAKDOWN OF THE AVERAGE INPUT AND OUTPUT POWERS, THE AVERAGE POWER LOSS, THE SYSTEM-LEVEL EFFICIENCY AND POWER CONVERSION EFFICIENCY DURING EXPERIMENT 3

Time Interval [s]	Interval 1 0-149	Interval 2 149-209	Interval 3 209-314	Interval 4 314-449	Overall 0-449
$< P_{in} > \ [W]$	514.55	402.96	458.91	524.46	489.58
$< P_{out} > \ [W]$	500.13	378.92	437.89	509.65	472.22
$< P_{loss} > \ [W]$	14.42	24.03	21.02	14.81	17.37
η_{sys} [%]	97.52%	94.36%	95.71%	97.50%	96.77%
η_{conv} [%]	98.78%	95.45%	96.87%	98.79%	98.00%

TABLE VII
BREAKDOWN OF THE AVERAGE INPUT AND OUTPUT POWERS, THE VARIOUS POWER LOSSES, THE SYSTEM-LEVEL EFFICIENCY AND POWER CONVERSION EFFICIENCY DURING EXPERIMENT 4

Time Interval [s]	Interval 1 0-30	Interval 2 30-90	Interval 3 90-120	Overall 0-120
$< P_{in} > \ [W]$	448.66	451.56	451.20	450.74
$< P_{out} > \ [W]$	426.34	431.40	428.70	429.46
$< P_{loss,meas} > \ [W]$	1.32	1.52	1.34	1.43
$< P_{loss,HS} > \ [W]$	2.49	2.46	2.52	2.48
$< P_{loss,wiring} > \ [W]$	2.82	4.73	2.89	3.79
$< P_{loss,conv} > \ [W]$	15.68	11.45	15.75	13.59
η_{sys} [%]	95.31%	95.86%	95.30%	95.58%
η_{conv} [%]	96.45%	97.41%	96.46%	96.93%
$< i_{S1} > \ [A]$	4.40	4.43	4.38	4.41
$< i_{S2} > \ [A]$	10.59	10.96	10.68	10.80
$< i_{S3} > \ [A]$	11.23	11.31	11.38	11.31
$< i_{S4} > \ [A]$	9.87	9.92	9.87	9.89
$< i_{string} > \ [A]$	8.76	11.05	8.83	9.92

current is any value between the second and third largest server current, which would also be $\sim$10 A.

In interval 1 and 3, the DPP converters are controlled using the voltage hysteresis control, whereas in interval 2, the optimal string current control is used. As can be seen from Fig. 17 and Table VII, the string currents in interval 1 and 3 are around 8.8 A, which are not optimal. In interval 2, the control method is changed to optimal string current control, and the string current changes to 11.05 A, which is effectively equal to the second largest sever current, 10.96 A, during interval 2 (the string current is slightly larger because a small portion of it also flows to power the logic circuits of the DPP converters). Due to the change of string current from non-optimal to optimal, the power conversion loss in the DPP converters $P_{loss,conv}$ reduces from $\sim$15.7 W to $\sim$11.5 W, as can be seen in Table VII, which is a 4.2 W or 27% drop in power loss. Consequently, the power conversion efficiency of the power delivery architecture increases from 96.46% to 97.41%, which is a 0.95% boost in efficiency.

Figure 18 shows the differential currents of the DPP converters. Since server 2 is the server whose current consumption is the second largest, the control algorithm in interval 2 decides to keep converter 2 turned off. It can be seen in Fig. 18 that $i_{DPP2,pri}$ stays zero during interval 2.[5] This causes the string current to be equal to server 2 current on average during interval 2.

[5] $i_{DPP2,pri}$ during interval 2 is zero for most of the time, but it looks a little noisy. This should be fine?

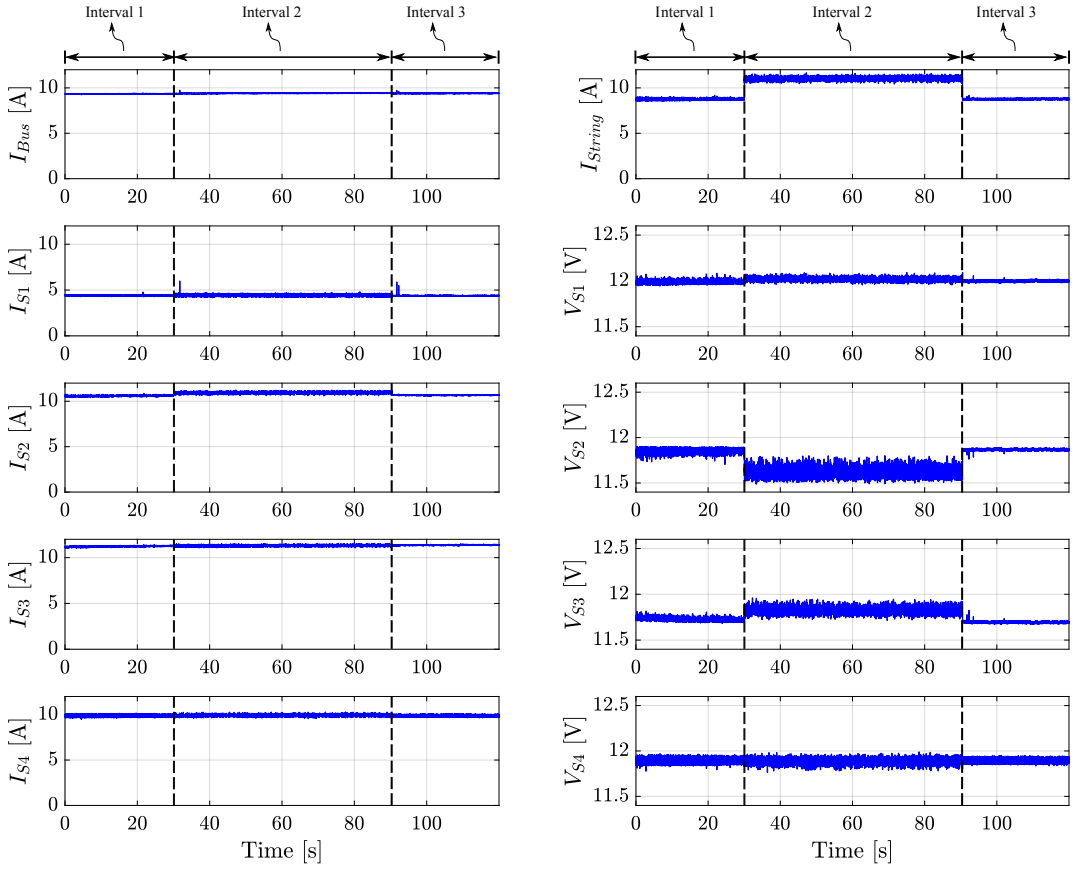


Fig. 17. Current and voltage waveforms in experiment 4. (Measured data is 10 ms window averaged for better illustration of the entire experiment on a single plot.)

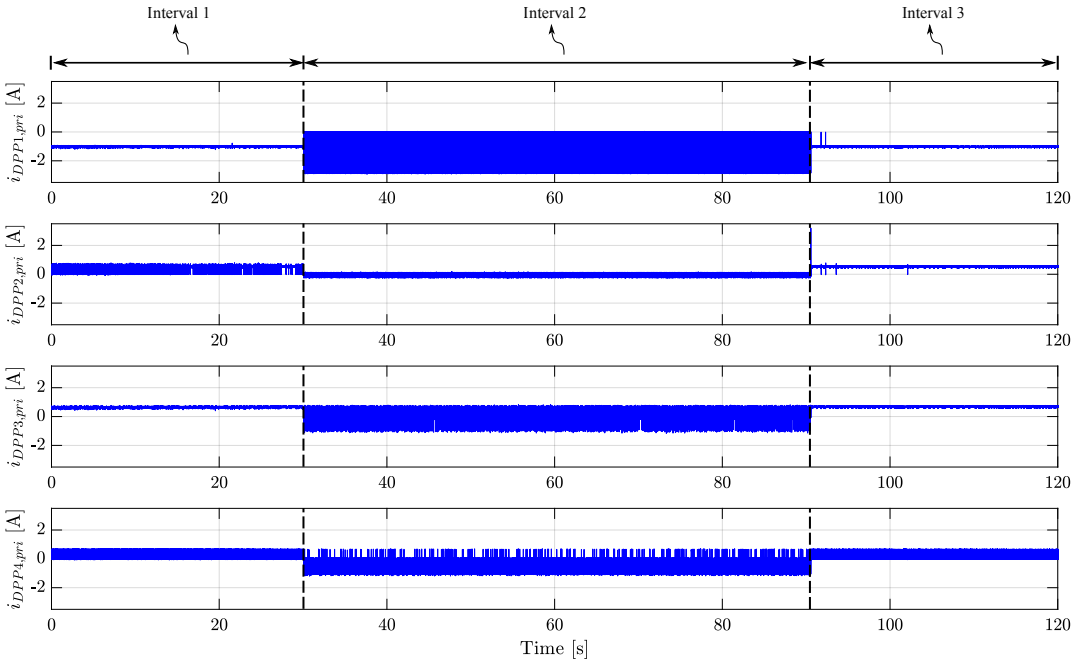


Fig. 18. Measured DPP currents in experiment 4.

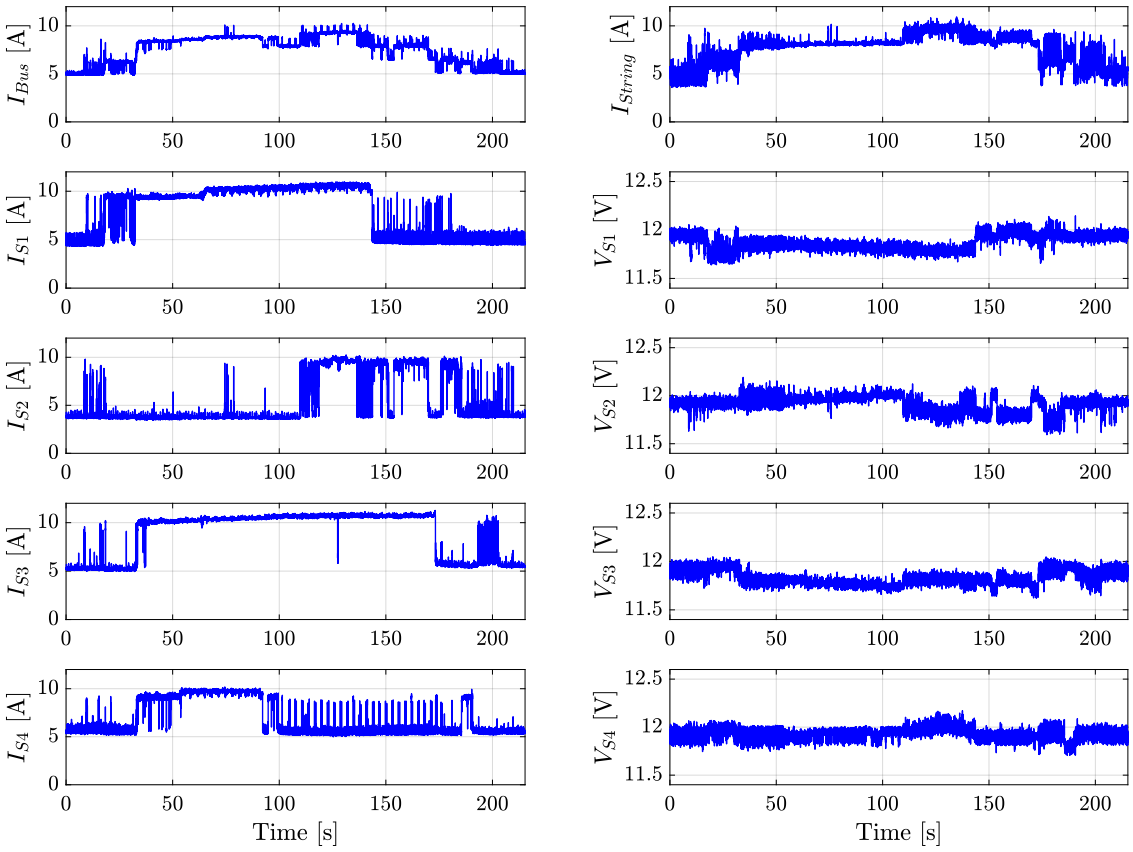


Fig. 19. Current and voltage waveforms in experiment 5. (Measured data is 10 ms window averaged for better illustration of the entire experiment on a single plot.)

5) Real data center computational task - Hadoop: Experiment 5 demonstrates that the series-stacked architecture can power servers while they are executing real data center distributed computational task, the Hadoop grep application. The current and voltage waveforms are shown in Fig. 19, and the power and efficiency results are given in Table VIII. In the entire experiment, the bidirectional hysteresis control with only voltage feedback is used.

As can be seen in Fig. 19, the four servers are idle initially, and then start to execute the Hadoop grep computation, finished the task successfully, and return to idle at the end of the experiment. During computation, the server currents are very fast-changing and not necessarily balanced, depending on the real-time computation need. The power conversion efficiency during the entire experiment is calculated as 96.43%. This experiment shows that the series-stacked architecture can supply power for the servers while they execute real-data-center Hadoop computational task.

V. Conclusion

The series-stacked power delivery architecture for data center servers can achieve a much higher energy efficiency than conventional architectures. In this paper, we provide the theoretical analysis and experimental validation of the 'server-to-bus type of series-stacked power delivery architecture. The server-to-bus type architecture has some unique properties that are not possessed by other types. Two important properties are analytically discussed. Firstly, the server-to-bus architecture is able to

TABLE VIII
THE AVERAGE INPUT AND OUTPUT POWER, THE AVERAGE POWER LOSS, THE SYSTEM-LEVEL EFFICIENCY AND POWER CONVERSION EFFICIENCY DURING EXPERIMENT 5

Time Interval [s]	0-215.3
$< P_{in} >$ $[W]$	358.04
$< P_{out} >$ $[W]$	340.67
$< P_{loss} >$ $[W]$	17.37
η_{sys} [%]	95.39%
η_{conv} [%]	96.43%

achieve the minimum power processed in the DPP converters. The value of the optimal string current for achieving minimum processed power is derived. The second important property of the server-to-bus architecture is its inherent 'redundancy in its DPP converters, yielding a high level of reliability. It can deliver power to n servers with only $(n-1)$ DPP converters in operation, i.e. the server-to-bus power delivery architecture can tolerate one converter failure. Both properties are demonstrated experimentally in this work, where four dual-active-bridge converters with GaN switches are used to build the series-stacked power delivery architecture, and four real computers are used as the server loads. Both normal and hot-swapping operations are experimentally validated, with the highest achieved power conversion efficiency being 99.04%. This work also presents the series-stacked architecture supplying power for servers while they execute real data center Hadoop computational task, which is the first time reported in literature[6].

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1509815, Texas Instruments, and the Air Force Office of Scientific Research under Grant No. FA9550-15-1-0128.

[6]can we say so?