<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>Peak power reduction and workload balancing by space-time multiplexing based demand-supply matching for 3D thousand-core microprocessor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>P. D., Sai Manoj; Kanwen Wang; Yu, Hao</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>Sai Manoj, P. D., Wang, K., &amp; Yu, H. (2013). Peak power reduction and workload balancing by space-time multiplexing based demand-supply matching for 3D thousand-core microprocessor. Design Automation Conference (DAC) 2013 50th ACM / EDAC / IEEE.</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>2013</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10220/18220">http://hdl.handle.net/10220/18220</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>© 2013 IEEE. This is the author created version of a work that has been peer reviewed and accepted for publication by Design Automation Conference (DAC) 2013 50th ACM / EDAC / IEEE, IEEE. It incorporates referee’s comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at: <a href="http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&amp;arnumber=6560768&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6560768">http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&amp;arnumber=6560768&amp;url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6560768</a>.</td>
</tr>
</tbody>
</table>
Peak Power Reduction and Workload Balancing by Space-Time Multiplexing based Demand-Supply Matching for 3D Thousand-core Microprocessor

Sai Manoj P. D., Kanwen Wang, and Hao Yu
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore 639798

ABSTRACT

Space-time multiplexing is utilized for demand-supply matching between many-core microprocessors and power converters. Adaptive clustering is developed to classify cores by similar power level in space and similar power behavior in time. In each power management cycle, minimum number of power converters are allocated for space-time multiplexed matching, which is physically enabled by 3D through-silicon- vias. Moreover, demand-response based task adjustment is applied to reduce peak power and to balance workload. The proposed power management system is verified by system models with physical design parameters and benchmarked power traces, which show 38.10% peak power reduction and 2.60x balanced workload.

Categories and Subject Descriptors: B.7.2 [Design Aids]

Keywords: Demand-supply matching, Peak power reduction, Workload balancing, 3D thousand-core

1. INTRODUCTION

Exa-scale cloud computing for big-data applications requires integration of many-core microprocessors on a single chip [1, 2] at thousand-core scale. Though 3D integration is one promising solution [3] to increase integration density and communication bandwidth, the provision of many-core power supply voltages with maintenance of low power density has become an unresolved issue to address [4, 5, 6, 7, 8]. Supplying same voltage-level to all cores will result in high power density because the demand of each core can be different at different time instant. As such, a demand-supply matched dynamic voltage and frequency scaling (DVFS) scenario needs to be employed during power management for both peak power reduction and workload balancing.

From physical hardware perspective, an optimal demand-supply matching requires on-chip power converters [5, 6, 7, 8], which can provide prompt DVFS management with efficient power delivery. However, one power converter for one core has large area overhead in presence of non-scalable buck inductor. The design of single-inductor-multiple-output (SIMO) power converters [6] utilizes one common single buck inductor to provide different voltage-levels at different time slots in a time-multiplexed manner. The capability of SIMO is, however, still limited for many-core microprocessors at thousand-core scale. Moreover, considering hundreds of cores to be integrated on one chip, the remaining area is quite limited to consider on-chip power converter with buck inductor. The 3D integration introduces additional room for on-chip power converters. The recent work in [8] has demonstrated the possibility to design power converter on one die and 64-tile network-on-chip on the other die, which are integrated by through-silicon-via (TSV).

From cyber management perspective, the power management for many-core power-supply system will no longer be the same as the one for the traditional single-core. For big-data applications, there may exist various power patterns deployed on many-cores with multi-time-scale demands for power supply. Moreover, there are many microprocessor cores but limited power converters. A number of power management works for many-core microprocessor system have been explored before [5, 6, 7, 8] but with not fully resolved challenge that requires to not only match various demands from microprocessors with limited number of power converters, but also to reduce peak power and to balance workload on a power converter. As such, the smart power management of many-core microprocessor has similarity as smart-grid though at different time-scale with different workload behaviors. Thereby, the study of workload behavior with classification and also the demand-response can be leveraged from smart-grid management [9] to deal with the on-chip demand-supply matching problem.

In this paper, a space-time multiplexing (STM) based DVFS power management is utilized for demand-supply matching between many-core microprocessors and power converters. In each power management cycle an adaptive clustering is developed such that the minimum number of power converters are allocated for different groups classified by power-magnitudes, called space multiplexing. In one group, power converters are further reused in different time slots for different subgroups classified by power-phases, called time multiplexing. Such a space-time multiplexed matching is physically enabled by designing a reconfigurable power switch network with the use of 3D through-silicon- vias (TSVs). Moreover, demand-response based task adjustment is applied to reduce peak power and to balance workload. The proposed power management system is verified by system models in SystemC-AMS. The physical design parameters are based on 130nm CMOS process with TSV models. Experiment results show that the proposed power management can achieve 38.10% peak power reduction and 2.60x balanced workload.

The rest of this paper is organized as follows. In Section 2, we present the 3D many-core microprocessor system architecture with space-time multiplexing (STM) problem formulation towards demand-supply matching. In Section 3, we show the solution by STM-based resource allocation of power converters with use of adaptive clustering, which is based on singular-value-decomposition (SVD) analysis of workload correlation. We further show the demand-response based task scheduling to utilize demand slacks and to adjust tasks for both peak power reduction and workload balancing. The experiment results are included in Section 5 with conclusion in Section 6.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC’13, May 29-June 07 2013, Austin, TX, USA.
Copyright 2013 ACM 978-1-4503-2071-9/13/05 ...$15.00.
2. 3D SYSTEM ARCHITECTURE WITH SPACE-TIME MULTIPLEXING

In this section, 3D many-core microprocessor system architecture with a reconfigurable power switch network is reviewed with a space-time multiplexing (STM) problem formulated for power management. Table 1 summarizes necessary notations used in this paper.

### Table 1: Notations and definitions

<table>
<thead>
<tr>
<th>Notations</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$V$</td>
<td>Set of voltage levels</td>
</tr>
<tr>
<td>$I$</td>
<td>Set of core current loads</td>
</tr>
<tr>
<td>$R$</td>
<td>Set of power converters</td>
</tr>
<tr>
<td>$C$</td>
<td>Set of cores</td>
</tr>
<tr>
<td>$P$</td>
<td>Set of power trace patterns</td>
</tr>
<tr>
<td>$S$</td>
<td>Set of switch boxes</td>
</tr>
<tr>
<td>$G$</td>
<td>Set of groups</td>
</tr>
<tr>
<td>$f$</td>
<td>Set of subgroups</td>
</tr>
<tr>
<td>$k$</td>
<td>Set of priorities</td>
</tr>
<tr>
<td>$r$</td>
<td>Set of slack</td>
</tr>
<tr>
<td>$s$</td>
<td>Set of workloads</td>
</tr>
<tr>
<td>$B$</td>
<td>Set of priorities</td>
</tr>
<tr>
<td>$v_{ci}(c)$</td>
<td>Demanded voltage-level of core $c_i$</td>
</tr>
<tr>
<td>$v_{ci}(C)$</td>
<td>Supplied voltage-level to core $c_i$</td>
</tr>
<tr>
<td>$v(C)$</td>
<td>Output voltage-level of converter $C$</td>
</tr>
<tr>
<td>$r(C)$</td>
<td>Output driving ability of converter $C$</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Maximum converter inductance current</td>
</tr>
<tr>
<td>$I_{CC}$</td>
<td>Maximum core supply-voltage drop</td>
</tr>
<tr>
<td>$H$</td>
<td>Time slot for time-multiplexing</td>
</tr>
<tr>
<td>$P_{th}$</td>
<td>Peak power threshold</td>
</tr>
</tbody>
</table>

- **Power Demand**: a set of cores $C$ with demanded voltage-levels with set-size $N_C$. Each core $c_i$ has a demanded voltage-level $v_{ci}(c_i)$ to meet the deadline of its running workload. In addition, $v_{ci}(c_i)$ is the allocated voltage-level to $c_i$ after power management.
- **Power Supply**: a set of power converters $R$ with set-size $N_R$. Each power converter outputs the voltage-level $v(r) \in V$ to supply the cores, where $V$ is the set of available voltage-levels before power management.
- **Power Switch Network**: a set of reconfigurable switch-boxes $S$ with set-size $N_s$ to connect between $R$ and $C$ for demand-supply matching.

### 2.2 Space-Time Multiplexing Problem

As aforementioned in the introduction, the primary challenge in 3D thousand-core system to support exa-scale computing is to solve a large-scale demand-supply matching problem. Though there are various big-data applications with different power patterns, most of their power profiles can be still classified by magnitudes and phases. As such, if one can perform a detailed power profile characterization by clustering cores with similar power behaviors, the complexity in matching may be accordingly reduced. With the further consideration for implementation with the minimum cost of power converters, it is still feasible to formulate a resource (power converter) allocation problem with constraints of demand and supply matching. As such, one can formulate the first subproblem as follows.

**Subproblem 1: Resource Allocation Problem** is to decide the minimum number of power converters such that demand-supply matching can be satisfied.

What is more, due to spatial and temporal variation of power profiles, there may exist lots of power slacks to be utilized for a demand-response based workload scheduling. Without violating the workload execution priority or deadline, one can delay over-loaded workloads in one time-slot to the other time-slot with under-loaded workloads. As such, the peak power can be reduced as well as the workload can be balanced at power converters, which can be formulated as the second subproblem below.

**Subproblem 2: Workload Scheduling Problem** is to delay over-loaded workloads to under-loaded time-slots based on availability of slack and without violation of priority.

In this paper, we show that based on the aforementioned 3D system architecture, a space-time multiplexing (STM) based power management can be developed to solve the two subproblems in sections 3 and 4, respectively.
profile characteristic within one power management control-cycle $T_c$.

### 3.1 Grouping by Power Magnitude for Space Multiplexing

Grouping is the process of clustering different cores, which have similar power magnitudes and hence will demand the similar voltage-level.

Note that the $z$-th group $g_z, g_z \in G$, can be formed by the following criteria

$$g_z = \{ c_i; v_d(c_i) = v_d(c_j) = v_z, \forall i, j = 1, \ldots, N_c, z \leq N_v \}.$$ (1)

Here, $v_z, v_z \in V$ represents the $z$-th voltage-level and $c_i, c_j \in C$ and $v_d(c_i) \in V$.

Based on the power magnitude levels, different groups are formed. Each group may contain different number of cores, which can have similar power magnitudes but maybe different power phases. The group formulation can change at different control-cycle. Based on the partitioned groups, power converters can also be partitioned in space to provide the specified voltage-levels for groups. This grouping process has less complexity because it involves just numerical comparisons.

### 3.2 Subgrouping by Power Phase for Time Multiplexing

Subgrouping is the process of clustering different cores, which have similar power phase (or pattern) and are within the same group.

Subgroup $k_s, k_s \in K$, can be formed by the following criteria

$$k_s = \{ c_i; (v_d(c_i) = v_d(c_j) = v_z) \& (p_i \sim p_j), \forall i, j = 1, \ldots, N_c \}.$$ (2)

Here, $p_1, p_i \in P$, represents the phase or pattern of one power trace of the core $c_1, c_j \in C$. $v_d(c_i) \in V$ represents the demanded voltage-level of core $c_i$ and $v_z$ represents the $z$-th voltage-level, $v_z, v_d(c_j) \in V$. However, the subgrouping by phase is more difficult than grouping by magnitude and may consume bit more time in clustering. In the next subsection, we show a solution by means of spectral clustering to perform subgrouping of power profiles, which can be easily deployed to make power management faster compared to the one without subgrouping. Moreover, all the computations can be pre-stored in a look-up-table for implementing a real-time control.

### 3.3 Spectral Clustering for Subgrouping

Spectral clustering algorithm is discussed below. To find similarity between two power profiles $p_i$ and $p_j, p_i, p_j \in P$, with $N$ samples in one control-cycle, correlation in term of covariance matrix can be evaluated by

$$X = \frac{1}{N} \sum_{i,j=1}^{N} (p_i - \overline{P})(p_j - \overline{P})^T$$ (3)

where $\overline{P}$ is the mean of all power profiles ($\frac{1}{N} \sum_{i=1}^{N} (p_i)$).

Based on the order of covariance matrix, number of clusters, $K$ can be analyzed by the singular-value-decomposition (SVD) of covariance matrix

$$X = U \times S \times V^{-1}.$$ (4)

Matrices $U$ and $V$ are orthogonal matrices with $S$ as the diagonal matrix. Based on the rank analysis of $S$, the number of clusters $K$ can be decided. A new matrix can be formed with $K$ independent vectors, extracted from either of orthogonal matrices. Let the newly formed matrix be $V_K$, assuming it is extracted from $V$. The product of $V_K$ with the covariance matrix $X$

$$X_K = X \times V_K$$ (5)

will result in a reduced matrix $X_K$, which becomes the basis of spectral clustering for subgrouping. For example, one core will be allocated to $i$-th subgroup if the value of $X_K(i,j)$ is the maximum in $j$-th row. The procedure for subgrouping is described in Algorithm 1.

#### Algorithm 1 Subgrouping by correlation extraction and spectral clustering

**INPUT:** Power trace matrix $P$ with $p_i$ power trace vectors after grouping
1. Compute covariance matrix $R \in R^{p \times p}$
2. Perform SVD: $R = U \times S \times V^{-1}$
3. Determine number of clusters: $K = rank(S)$
4. Compute the first K singular-value vectors $v_1, \ldots, v_K$ of $V$
5. Let $V_K = [v_1, \ldots, v_K] \in R^{N \times K}$ and $R_K = R \times V_K$
6. Add $j$th core to $k$th cluster if $R_K(i,j)$ is maximum in the $i$th row
7. Form $P_K$ matrices by finding corresponding indices in power trace matrix $P$

**OUTPUT:** New clustered subgroup matrices $P_K, (k = 1, \ldots, K)$

![Figure 2: Grouping and subgrouping based on power levels and power phases](image)

The formulation of groups and subgroups are illustrated in Fig. 2. At one control-cycle, power traces $p_1, p_2, p_3, p_4$ and $p_5$ are operating at one power magnitude level and other cores are working at a different power magnitude level. As such, one can form two groups with two voltage-levels $v_1$ and $v_2$. Inside the group supplied by voltage-level $v_1$, one can observe that $p_1, p_2, p_3$ have a similar power phase compared to $p_4$ and $p_5$; so $p_1, p_2$ and $p_3$ further form a subgroup and $p_4$ along with $p_5$ forms another subgroup. The formed groups and subgroups can change at the next control-cycle.

In the following, we show that with the help of adaptive clustering, one can find the minimum number of power converters to satisfy the demand-supply matching. Moreover, by clustering, the complexity from the demand (power profiles) can be significantly reduced. As such, the large-scale demand-supply matching can be efficiently solved by the proposed two-step clustering in every control-cycle.

### 3.4 Solution to Subproblem 1

Once subgroups are formed, the maximum workloads of one subgroup can be determined. As such, the minimum number of power converters can be also determined to supply that subgroup. This results in one feasible solution to solve the Subproblem 1 in Section 2 as rephrased below.

$$\min \sum_{i=1}^{N_c} r_i$$

s.t.: (i) $v_d(c_j) \geq v_d(c_i), \forall c_j \in C$.
(ii) $d(r_i) \leq N_{max}, \forall r_i \in R$.

If one can determine the minimum number of power converters $r_i$ for each group, the total number of power converters can be correspondingly minimized. Note that constraint (i) guarantees that the supplied voltage-level $v_d(c_j)$ from power converter will
satisfy the demanded voltage-level \( v_{l,cj} \) from core \( c_j \). Moreover, constraint (ii) imposes the driving ability \( d(r_i) \) of each power converter is \( N_{\text{max}} \), i.e., the maximum number of cores to drive. The driving ability can vary with the voltage-level: the higher the voltage-level is, the lower the number of cores that one power converter could drive.

Next, we show that the minimization of total number of power converters can be solved by grouping and subgrouping. By performing grouping, power converters are shared in space among \( N_c \) number of groups and subgrouping makes sharing of power converters inside a group in time. Based on the driving capability \( d^i_k \) of \( i \)-th power converter in group \( g_j \), \( g_j \in G \), having \( k \) subgroups, and the maximum number of cores among different subgroups, \( \max(c_j) \), \( c_j \in C \), the minimum number of power converters for group \( g_j \) can be determined as:

\[
r_{g_j} = \max(c_j)/d^i_k.
\]

As such, for the whole system, the total number of power converters needed will be \( \sum(r_{g_j}) \), which is the minimum number to satisfy the demand-supply matching.

### 4. DEMAND RESPONSE BASED WORKLOAD SCHEDULING

This section deals with peak reduction and load balancing after the minimum number of power converters are allocated. A demand-response based workload scheduling will be developed towards uniform distribution of workload with reduction in peaks at one power converter.

#### 4.1 Peak Power Envelope Extraction

To deal with peak reduction and load balancing, we first discuss the extraction of peak power envelope in one control-cycle, because it is impractical to perform power management in continuous form. Based on the extracted peak power envelope, one can build workload behavior model for each subgroup to be used in scheduling.

Assume that in one control-cycle \( T^i \) for the \( i \)-th group, \( g_i \), \( g_i \in G \) with \( N_j \) number of subgroups, each core is assigned with one workload. One can have time slot \( T^i_j \) which is the amount of time to finish all workloads in a subgroup, \( k_j, k_j \in K \). Relation between \( T^i \) and \( T^i_j \) is:

\[
T^i = \sum_{j=1}^{N_j} T^i_j. \tag{7}
\]

As such, in one time-slot \( T^i_j \), peak power envelope \( P_e \) is extracted for workload \( p(t) \) of one subgroup by:

\[
P_e(T^i_j) = \max(p(t)). \tag{8}
\]

This is repeated for whole control cycle \( T^i \). Thus peaks are extracted and one envelope is formed. Peak extraction by forming one envelope is shown in Fig. 3. The control-cycle \( T^i \) is 400ns with time-slot \( T^i_j \) of 100ns. At each time slot, the power envelope is formed on the peak value.

#### 4.2 Peak Reduction and Load Balancing

When the peak envelope of subgroup \( k_i, k_i \in K \) is compared with one threshold power \( P_{th}^i \) of group \( g_i \), the slack can be calculated by:

\[
a^i_j = P_e(T^i_j) - P_{th}^i. \tag{9}
\]

If the value of slack is positive, then the allocated power converter, \( r_j, r_j \in R \), is overloaded and not capable of handling extra load at that time-slot; otherwise, the power converter \( r_j \) is underloaded and can be allocated with additional workloads. After calculating the amount of slack, the workload of the power converter \( r_j \) can be rescheduled such that priority is not violated.

We call such a scheduling as demand-response based workload scheduling. The procedure for scheduling is described in Algorithm 2. It is deployed after clustering to decide the time slot. The first step in scheduling is to calculate the threshold and slack. Line 2-4 of Algorithm 2 explains the scheduling of task from a power converter that is overloaded and reduction of corresponding load. Similarly Line 6-8 describes adding of workloads on an underloaded power converter. In short, it can be viewed as re-clustering or refinement. The overhead includes the time to perform the calculation and movement, which is negligible in the whole control cycle.

**Algorithm 2 Demand-response based workload scheduling**

1. **INPUT:** Initial set Workload \( L \), Slack \( A \)
2. if \( a^i_j > 0 \) then
3. Decrease workload on \( r_j \)
4. \( l(r_j) = -\);
5. else
6. while \( a^i_j < 0 \) do
7. Increase workload on \( r_j \)
8. \( l(r_j) = +\);
9. \( a^i_j = +\);
10. end while
11. end if

Example in Fig. 4 shows the peaks of four subgroups. Before performing demand-response based workload scheduling, subgroup 2 and subgroup 3 are overloaded and subgroups 1 and 4 have slacks for scheduling. The peak value in subgroup 2 and 3 is 5, which means there are 5 peaks in those two subgroups. The peak power reduction is then achieved with the comparison of the highest value in subgroups before and after the demand-response scheduling. After the demand-response scheduling, the peak value will be reduced to 4. So, a 20% peak power reduction will be achieved.

#### 4.3 Solution to Subproblem 2

The aforementioned demand-response based workload scheduling can be deployed to solve the Subproblem 2 addressed in Section 2, which is reformulated as:

\[
\begin{align*}
\min: & \quad \sum_{j=1}^{N_j} \sum_{i=1}^{K_i} a^i_j \\
\text{s.t.:} & \quad P_e(T^i_j) < P_{th}^i
\end{align*} \tag{10}
\]
reduction has to be performed followed by load balancing. So, initially peak demand power more than the threshold. So, initially peak reduction has to be performed followed by load balancing. Based on the value of slack for a subgroup $s_j$, if the slack is positive, then the workload on that subgroup needs to be delayed or advanced to other time-slot. As such, the workloads are allocated to subgroups with highly negative slack, and the differences in slack is reduced. As a result, peak reduction and load balancing can be achieved eventually.

Table 5: Comparison of peak power reduction and workload balancing by demand-response scheduling

<table>
<thead>
<tr>
<th>Group</th>
<th>Peak Reduction</th>
<th>Balance before</th>
<th>Balance after</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group 1</td>
<td>90%</td>
<td>0%</td>
<td>90%</td>
</tr>
<tr>
<td>Group 2</td>
<td>80%</td>
<td>10%</td>
<td>90%</td>
</tr>
<tr>
<td>Group 3</td>
<td>70%</td>
<td>20%</td>
<td>90%</td>
</tr>
<tr>
<td>Group 4</td>
<td>60%</td>
<td>30%</td>
<td>90%</td>
</tr>
<tr>
<td>Average</td>
<td>80%</td>
<td>15%</td>
<td>90%</td>
</tr>
</tbody>
</table>

Solution to this problem is to minimize the overall sum of slacks. This can be achieved by rescheduling workloads that demand power more than the threshold. So, initially peak reduction has to be performed followed by load balancing. Based on the value of slack for a subgroup $s_j$, if the slack is positive, then the workload on that subgroup needs to be delayed or advanced to other time-slot. As such, the workloads are allocated to subgroups with highly negative slack, and the differences in slack is reduced. As a result, peak reduction and load balancing can be achieved eventually.

5. SIMULATION RESULTS

5.1 System Modeling and Settings

The proposed system is validated by Matlab and system-level models built from SystemC-AMS. Table 2 summarizes the system design specifications. All units are scaled or modeled at CMOS 130nm CMOS process. The specification of low-power MIPS microprocessor core [10] is taken as the core model. Each core has the nominal frequency of 250MHz with the maximal power consumption of 0.4W. Benchmarks from SPEC2000 [11] are simulated by Watch [12] to generate power profiles. The extracted power profiles are used as workload models, which are distributed to different cores randomly. The typical control cycles for power management is 400ns.

A 2-phase multi-output power converter [13] is designed to generate different voltage-levels. As the driving ability of power converter depends on supply voltage-level, driving abilities are set as 3, 2, 1 for voltage-levels of 0.6V, 0.8V, 1.0V and 1.2V respectively. Moreover, the inductance value in power converter is set as $L_1$ to support the maximum current on the buck inductor. Such an inductor requires an area of 0.25mm$^2$, occupying 30% area of the power converter. The local super-capacitor for each core is set as $C_1$, to support time-multiplexing scheme between clusters. The design of on-chip power converter thereby needs to consider the limitation of inductor and capacitor area, which are both placed in 3D fashion and hence has the minimum area overhead to cores all on the other tier.

In addition, the vertical TSV [14] with size of 500$\mu$m$^2$ works as connections between cores and power converters. According to the model in [15], it has a dc-resistance of 20m$\Omega$. Considering the maximum current of 330mA, the IR-drop of is around 7mV, which is quite small. Note that the capacitor of TSV is in F-scale and hence does not influence the load capacitance. What is more, for each TSV channel, one switch box is assigned with $N_r$ power switches to support the core-converter connection. The switch box offers a compact reconﬁgurable unit driven by the controller. The power switch inside each switch box occupies 520$\mu$m$^2$ and is able to deliver the maximum core current with switching time of 300ns. As such, the TSV coupling is also quite small to consider under such a slow power switching.

5.2 Results and Comparisons

Firstly, we take 32-core and 64-core microprocessors as two examples to show results under adaptive clustering. The input power traces are first grouped into 4 based on the power magnitudes, then in each group subgroups are formed based on their power phases. Fig. 5 illustrates the adaptive clustering result of 32-core between two consecutive control cycles. Different filling-shapes represent different groups or voltage-levels. Different clustering numbers on the down-right corner of cores represent different subgroups. For example, in the first control cycle, the 30th core will be assigned to subgroup 4 with voltage-level 4 (group 4). And in the next control cycle, it will be assigned to subgroup 2 with voltage-level 1 (group 1). For 64-core case, Table. 3 summarized the clustering results with the value in the table to represent the core ID. One can also observe that the runtime of clustering is small at the scale of 200ns.

Next, we use the space-time multiplexing (STM) scheme to perform the demand-supply matching. The first step is for resource allocation and adaptive clustering is deployed. After clustering, we extract simplified workload models to represent the peak power in one control cycle, and also determine the minimum number of power converters for each group. When comparing two schemes, namely space-multiplexing (SM) and time-multiplexing (TM) with the same driving ability and time slot, the STM-based approach takes the advantage of both space and time to minimize the number of converters. Table. 4 shows the comparison for 32-core and 64-core cases with the three schemes. One can observe that 55.00% (SM) and 35.71% (TM) number of power converters can be reduced for the case of 32-core, while 41.67% (SM) and 36.36% (TM) number of power converters can be reduced for the case of 64-core. Therefore, STM based adaptive clustering can satisfy the demand-supply matching with the minimum number of power converters to reduce the area overhead and also on-chip implementation cost.

Lastly, we perform demand-response based workload scheduling for time-multiplexing of power converters inside one
The peak power reduction is defined as the difference of peak power value before and after the scheduling. The workload balancing is defined as the number of cores which one power converter drives over control cycles. We compare the peak power reduction by averaging the reduction in each group; and compare workload balancing by demand-response scheduling. One can observe an average of 38.10% peak power reduction and 2.60x workload balancing.

6. CONCLUSION

A space-time multiplexed power management is developed for large-scale demand-supply matching between on-chip power converters and many-core microprocessors. The power switch network is configured to perform space-time multiplexing between power converters and cores by vertical TSVs in 3D. Based on adaptive clustering of cores classified by both power magnitudes and power phases, the minimum number of power converters are allocated to supply the demanded voltage-levels from cores. What is more, demand-response based workload scheduling is deployed by utilizing the power slacks, such that peak power can be reduced as well as workload can be balanced. As verified by system-level behavior models implemented in SystemC and SystemAMS, and also physical-level models with design parameters, experiment results show that the space-time multiplexing can reduce peak power by 38.10% and improve load balancing by 2.60x improvement on average with the minimum number of allocated power converters.

7. ACKNOWLEDGMENTS

This work is sponsored by Singapore MOE TIER-2 fund MOE2010-T2-2-037 (ARC 5/11) and A*STAR SERC-PSF fund 11201202015. Please address comments to haoyu@ntu.edu.sg.

8. REFERENCES