Energy-efficient fuzzy control model for GPU-accelerated packet classification

Guo Li | Dafang Zhang | Yanbiao Li | Jintao Zheng | Keqin Li

College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China

Correspondence
Dafang Zhang, College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China.
Email: dfzhang@hnu.edu.cn

Summary
As a core component of many network infrastructures, packet classification requires matching packet headers against a series of predefined rules. Its performance determines, to some extent, how fast packets can be processed. There already exists many proposals, which optimize the throughput of packet classification, but few of them take power consumption into account. To meet the requirements of green network computing, this paper focuses on energy-efficient solutions that provide reasonable throughput as well. Similar to recent advancements, the graphics processing unit (GPU) is adopted to accelerate rule matching. Then, inspired by the frequency-variable energy-consuming model for air conditioners, a fuzzy control–based energy efficiency optimizing model is proposed for GPU-accelerated packet classification. As demonstrated in the evaluation experiments, when the GPU is in the idle status, the proposed model can save 10 W. In running status, the fuzzy control–based energy efficiency optimizing model can avoid GPU shutdown issue caused by GPU self-protection mechanism when the GPU temperature rises to 95°C. Furthermore, by improving the resource configuration of GPU kernels according to the model, the overall energy efficiency is enhanced by up to 15.5%, while simultaneously keeping throughput at the same level.

KEYWORDS
energy-efficient, fuzzy control, GPU, packet classification

1 | INTRODUCTION

Packet classification is the key functional module in many network devices, such as Firewall, priority routing-enabled routers, and OpenFlow switch, to name only a few.

Its core operation is to match packet headers in the rule table and then to process the input packet according the matched rule(s).

Performance has long been a hot topic in packet classification, while energy efficiency is gaining traction recently. In comparison to conventional solutions, ternary content addressable memory (TCAM)-based schemas achieve really high performance. But their high power consumption indeed restricts their use in practice.

Recently, the graphics processing unit (GPU) has been shown to be of value in supporting high-speed packet processing. It is also more controllable than TCAM and thus poses more chances to control power consumption while maintaining superior performance.

It is also why more supercomputers in Green500 list than Top500 list use GPUs as cooperating processors.

In TCAM-based solutions, the overall power consumption is determined by the number of activated blocks to process matching, the total number of entries, and the lengths of entries as well. Hence, to reduce consumed energy, some smart techniques have been adopted to group or preprocess the rules. While in the GPU scenario, there are much more controllable factors, which affect power consumption during packet classification, such as the number of threads activated and even how they are arranged, the detail behavior of memory accesses and calculations during the kernel execution, and so on. We have more chances to control power consumption, which, correspondingly, makes the task more challenging. To our best knowledge, till the writing of this paper, no current work conducted focuses on energy-efficient optimization of GPU-accelerated packet classification.

The key idea of our approach is partly derived from a daily energy-consuming product, the air conditioner, whose power consumption can be significantly reduced via a frequency-variable control system. And such a frequency-variable feature is also enabled in some advanced models of modern GPUs. Therefore, in this paper, we...
propose the fuzzy control–based energy efficiency optimizing (FCEEO) model that introduces a fuzzy control mechanism to reduce GPU power consumption when processing packet classification, while keeping a high throughput at the same time.

The rest of this paper is organized as follows. An overview of prior work on the subject is given in Section 2. Section 3 describes the packet classification. In Section 4, we provide details of the energy model. The experimental results are presented in Section 5, and finally, Section 6 summarizes the main conclusions of this work.

2 | RELATED WORK

The most widely used packet classification hardware is TCAM; however, its high power consumption has restricted the development of TCAM. Therefore, there are many studies on the energy-efficient issue of TCAM.

Agrawal proposed a TCAM power model, describing how TCAM power is scaled with parameters such as voltage, operating frequency, number of entries, length of entries, and circuit-level parameters. Meiners used a TCAM power model for optimizing the power consumption of packet classification with the proposed TCAM SPIIT architecture.

In recent years, packet classification based on GPU has become a research focus. In the meanwhile, GPU also suffers from the problem of high power consumption.

In the field of low-power GPU research, Rhu designed a locality-aware memory hierarchy to improve the GPU performance and energy efficiency by adaptively adjusting the access granularity. Ma proposed GreenGPU, a holistic energy management framework for GPU-CPU heterogeneous architectures. By distributing workloads and throttling the frequencies of the GPU cores and the memory dynamically, GreenGPU can maximize energy savings with only marginal performance degradation.

GPUWattch is a GPU power consumption model, which has configurable clock cycle-level power modeling tools. Therefore, GPUWattch has high accuracy for energy modeling. Using GPUWattch for measurement, it shows that dynamic voltage and frequency scaling (DVFS) algorithms are useful for reducing dynamic power consumption in general-purpose GPU workloads.

However, GPUWattch is not suitable for packet classification energy consumption calculation analyses. Further, the supported modeling products are limited, such as GeForce GTX480, Quadro FX5800, and Tesla C2050, and do not include Tesla K20, which is our experimental platform.

Combining the features of multiparameters and variable frequency on GPU platform, some methods can be used for reference with respect to energy saving.

proposed a workload-dependent dynamic power management model in a multicore server environment, to reduce energy consumption through M/M/m queuing models and digital circuit power models. This technique considers parameters such as energy supply, core speed, task response time, and task processing speed by optimizing the average task response time, to improve the system performance and reduce the power consumption.

Although TCAM has a query speed of \( O(1) \) by querying all entries in parallel, the high energy consumption is a serious problem. Compared to TCAM, GPU has better parallel controllability, which

![Figure 1](image_url)
brings us a method to control the power consumption of packet classification.

The packet classification forward principle for GPU is similar to that for TCAM. The difference is that the GPU launches multithreads to deal with a batch of packets simultaneously. Further, GPU has better programmable and extensible functions.

The speed of the GPU memory is faster than that of an ordinary memory. A mainstream PC memory is DDR3 SDRAM with an equivalent frequency of 1600 MHz, and the mainstream GPU memory is GDDR5, which can reach an equivalent frequency of 5400 MHz. GDDR5 is based on the DDR3 SDRAM memory but is specifically for GPU use and has higher computational performance.

If we put the rules in a linear mode on the basis of the PC memory, we cannot achieve the desired performance as in the case of TCAM, because TCAM is parallel hardware. There are some packet classification algorithms that can improve the matching speed, such as hash, tries tree, bit vector, HiCuts, HyperCut, and EffiCuts. However, computing with CPU and PC memory leads to a hardware performance bottleneck. Hence, we implement the HiCuts algorithm with a parallel-accelerating hardware GPU. The HiCuts algorithm changes the linear rule placement to a multidimensional space placement, searching in the subregions recursively. The use of the GPU hardware considerably enhances the throughput performance. By comparing with the result of linear algorithms, our improved HiCuts algorithm will not affect the accuracy of packet classification. We did not study the energy problem in our previous work but have now realized that the issue of energy efficiency is important in practical GPU execution. Therefore, we conducted this further study.

4 | MODEL-DRIVEN ENERGY OPTIMIZATION

4.1 | Fuzzy control model

Fuzzy control is based on fuzzy logic, which is applied to the fields of control, artificial intelligence, and so on. The term “fuzzy” refers to the logic that cannot be exactly expressed by “true” or “false” but by “partially true.” Fuzzy control simulates the operation of the human-reasoning process. When the complexity of the system increases, the values of the variable parameters may not just increase but also change frequently. Some of these factors are difficult to grasp; instead, people focus on the main part and ignore the secondary part. Thus, in fact, the description of the system is fuzzy.

As shown in Figure 2, the basic structure of fuzzy control contains fuzzy control rules, fuzzy inference, and input and output components. The process of fuzzy inference is based on a collection of fuzzy logic rules in the form of IF-THEN statements, where the IF part is called the “antecedent” and the THEN part is called the “consequent.”

For example, a set of fuzzy rules may look like the following:

IF temperature IS very cold THEN stop fan
IF temperature IS cold THEN turn down fan
IF temperature IS normal THEN maintain fan
IF temperature IS hot THEN speed up fan

The main fuzzy inference process is as follows:

IF X₁ is A₁, IF X₂ is A₂, … and IF Xₙ is Aₙ, THEN Y is B.

A special characteristic is that the interval of fuzzy variables may have some overlaps, as shown in Figure 3. This is an important difference between fuzzy inference and classical inference.

In the variable-frequency air conditioner, which uses fuzzy control model, more energy can be saved. Figure 4 is an example of energy consumption comparison between variable-frequency air conditioner and fixed-frequency air conditioner.

4.2 | Energy-efficient fuzzy control GPU model

From the TCAM energy consumption analysis model and the GPUWatch model, we can conclude that the power consumption will be affected by many parameters, such as voltage, frequency,
temperature, computing tasks, parallel program efficiency, and computing time. Further, some parameters are difficult to measure or control. If we unilaterally increase the device frequency to add the task processing speed, the reduction of task execution time can decrease the power consumption. However, on the other hand, the higher temperature caused by the speed will increase the total power consumption.

Hence, various parameters constitute a game system. To design an energy consumption optimization solution based on the GPU for packet classification, by introducing a fuzzy control model that has achieved better energy-saving results in the case of a variable-frequency air conditioner, we propose the FCEEO model.

In the case of the energy optimization solution for GPU-based packet classification, we find that the following groups of parameters are related to energy consumption:

(1) GPU hardware configuration parameters
- GPU compute mode
  Graphics processing unit has 4 compute modes: Default, Exclusive_Thread, Exclusive_Process, and Prohibited. By adjusting the mode, we can adopt the corresponding hardware features for computing, and this may lead to different power consumption values.
- GPU running frequency
  If sufficient power supply and thermal headroom are available, increasing the GPU core and memory clock frequency can enhance the GPU performance within a reasonable range. Nvidia Tesla K20 GPU supports 6 running frequencies.

(2) GPU software configuration parameters
- GPU thread scheduling
  Reasonable task decomposition and appropriate thread scheduling can increase the speed of parallel computing, thereby reducing the total computation time to save energy. The GPU can allocate the number of grids, blocks, and threads to schedule the threads. In fact, the interval of these allocating variables may have overlaps.
- Algorithm optimization
  The GPU has different levels of memory; an efficient use of the GPU memory can reduce the computation time to save energy.
- Data calculation scale
  Network packet traffic has a seasonally changing regularity, such as leisure time and busy time, and this trend can be measured and predicted. During the busy time, a relatively large-task calculation scale will consume more energy.

(3) Other relative parameters
- GPU temperature
  Heavy computing tasks may lead to an increase in temperature, which may in turn lead to an increase in overall power consumption. Although it is difficult to compute the effects of temperature on GPU energy consumption, by monitoring the GPU temperature, we can obtain a reference for power consumption. Meanwhile, GPU Tesla K20m has a high temperature self-protection mechanism; the GPU will shut down if the GPU temperature reaches 95°C.
- Throughput speed
  In packet classification, the primary guarantee is throughput speed; therefore, energy savings cannot come at the cost of high-throughput performance.

Here, we use $P$ to represent the GPU total power; $S$, the packet classification throughput (speed); $N$, the task calculation scale (the number of the tasks); $F$, the GPU running frequency; $T$, the working temperature; $A$, the GPU thread allocating optimization level (will be divided into several levels of optimization); and $O$, the algorithm optimization level ($O_1 = \text{unoptimized}$ and $O_2 = \text{optimized}$).

We use the fuzzy control model to integrate these parameters. The FCEEO model is specified in Figure 5; the fuzzy inferformulas are in Equations (1) and (2); here, $P$ denotes the power parameter that should have as small a value as possible. The total power $P$ is in direct proportion to $S, N, F,$ and $T$, and in inverse proportion to $A$ and $O$. The variables $F, A,$ and $O$ can be adjusted in the GPU configuration. Further, $S$ represents the speed parameter, which should be as fast as possible; $S$ has a direct relationship with only $A, O,$ and $F$.

$$P \propto \frac{SNFT}{AO}. \quad (1)$$

$$S \propto \frac{AO}{F}. \quad (2)$$
For example, a set of fuzzy rules is as follows:

- IF GPU IS idle THEN set Compute Mode in Prohibited
- IF GPU IS running THEN set Compute Mode in Default
- IF power IS high THEN decrease Frequency
- IF speed IS low THEN increase Frequency
- IF power IS high THEN decrease Grid, Block, or Thread
- IF speed IS low THEN increase Grid, Block, or Thread

In the next experiment, we use FCEEO to find the minimum total power $P$ while ensuring that the value of $S$ is as high as possible. First, we fix the $N$ scale and maximum $S$ and then adjust the $F$, $A$, and $O$ parameters, to find the minimum value of $P$.

5 | EXPERIMENTS AND EVALUATION

To simulate a compute node in a heterogeneous supercomputing system, we set up the experimental environment on a Dell PowerEdge T620 server, which runs Ubuntu Server 12.04 systems and is equipped with 2 Nvidia Tesla K20 GPUs and 2 Intel E5-2630 CPUs. We measure the real-time power and temperature via the Nvidia management library.*

Firstly, we conduct an experiment that collect real-time running status of GPU while performing packet classification. The results are shown in Figure 6. As depicted, both the power and the temperature change continuously. During the first few seconds, the power changes significantly because of the device initialization. Then, the main program only runs on GPU1. And the power consumption drops obviously when the main program finishes 61 seconds later.

Next, we studies various types of parameters, which may affect the energy consumption.

*Note that K20 is in the limited list of supported models of Nvidia management library.
Therefore, we can set the GPU compute mode to the prohibit mode when there is no program running. Compared to the default mode, the prohibit mode can save 10 W of power.

5.1.2 | GPU Running Frequency
Tesla K20 GPU has six supported combination frequencies as Table 1; by adjusting the frequency, we can change the GPU speed.

The experimental parameters are as follows: data trace no. 10467100, rule acl_1k, linear algorithm, and default compute mode.

The result is shown in Figure 8. When the GPU core and the memory clock frequencies are set to 705 and 2600, respectively, the energy consumption is the lowest and the throughput is relatively high. In fact, this combination of frequencies is the default setting.

5.2 | GPU software configuration parameters
5.2.1 | GPU thread scheduling
We change the allocation of the number of grids, blocks, and threads, to find their impact on the energy consumption.
The experimental parameters are as follows: data trace no. 10467100, rule acl_1k, linear algorithm, and default compute mode.

As shown in Figure 9, when the minimum number of threads is allocated in {1 grid-1 block-256 threads}, the energy consumption is the highest at 2429.83 J. The top square shows the projection to assist the 3D image.

When allocating more thread resources, we find that the energy and the program-running time show a corresponding relationship, as shown in Figure 10; that is, the more the parallel computing resources, the shorter is the running time and the less is the energy consumption. However, the lowest energy is not allocated in {1 grid-8 blocks-1024 threads}, instead by {1 grid-8 blocks-768 threads} allocating method with 133.84 J.

This implies that the allocation of superfluous parallel-computing resources will result in an increase in the energy consumption.

### 5.2.2 Algorithm optimization

Figure 11 shows an energy comparison in different algorithms. Here, “Linear” is the baseline algorithm, “DVFS” is the traditional energy-efficient algorithm used on GPU, “HiCuts on GPU” is the HiCuts algorithm, which changed the linear rule placement to a multidimensional space placement and implemented on GPU platform, and “FCEEO” is the optimized algorithm based on “HiCuts on GPU.”

We find that the “Linear” algorithm has a larger energy consumption due to that the computing time is proportional to data scale. “DVFS” algorithm has limited adjustable parameters. “FCEEO” algorithm is energy efficient because of that the fuzzy control model can adjust more parameters and select a better GPU running solution.

### 5.2.3 Data calculation scale

Next, we conducted a packet classification experiment by changing the scale of the trace data. The other parameters were fixed as follows: rule acl_5k, linear algorithm, and default compute mode.

As shown in Figure 12, with an increase in the data scale along the x-axis, while the throughput is stable, both the GPU computing time and the energy consumption exhibit a trend of linear growth. This implies that the GPU handles more data with more energy consumption and that these 2 parameters have a linear relationship.
TABLE 2  The accuracy of packet classification with acl_1k data

<table>
<thead>
<tr>
<th>Trace No.</th>
<th>Linear on CPU</th>
<th>Hicuts on CPU</th>
<th>Hicuts on GPU</th>
<th>Hicuts on GPU (Prohibited Mode)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>591</td>
<td>591</td>
<td>591</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>7868</td>
<td>407</td>
<td>407</td>
<td>407</td>
<td>0</td>
</tr>
<tr>
<td>7869</td>
<td>716</td>
<td>716</td>
<td>716</td>
<td>0</td>
</tr>
<tr>
<td>7870</td>
<td>571</td>
<td>571</td>
<td>571</td>
<td>0</td>
</tr>
</tbody>
</table>

Abbreviations: CPU indicates central processing unit; GPU, graphics processing unit.

FIGURE 14  Accuracy of packet classification

5.3  Other relative parameters

5.3.1  GPU temperature
The experimental GPU product model is Tesla K20m, a passive cooling GPU without a fan. Here, we record the changes in temperature. The experimental parameters are as follows: data trace no. 8289120, rule acl_5k, linear algorithm, and default compute mode.

As shown in Figure 13, when the GPU is in the idle state, the temperature is relatively low 51°C (123.8°F). After the GPU changes into the working state, the temperature increases along with the working load. The highest temperature is 88°C; this will increase the energy consumption of the computer-cooling process.

5.3.2  Packet classification accuracy
We conducted an additional experiment on the accuracy of packet classification for evaluation. Table 2 shows the accuracy of packet classification with acl_1k data. The first column is trace number, the second column linear on CPU is a baseline algorithm, which provides correct matched rule number. The third and fourth columns are matched rule numbers with our algorithms; the same results indicate that the accuracy is 100%. The other types of firmware and interprocess communication data have the same results as shown in Figure 14. We used different configuration parameters according to FCEEO model but found that when the GPU is set to prohibited mode, the accuracy is 0% as shown in column 5 of Table 2. This is normal because GPU has been stopped. Therefore, using the FCEEO model to change GPU frequency and adjust other parameters will not affect the accuracy of packet classification when GPU is in working status.

5.3.3  Throughput and energy
We find that the allocation of a different number of grids, blocks, and threads will result in different performance values. To achieve a higher throughput, we need to reasonably assign GPU hardware resources. Previous experiments mainly considered only the aspect of high performance and did not take into account the energy efficiency while maintaining the high performance of GPU-based packet classification. We conducted experiments on power consumption issues with different resource allocation combinations, as shown in Figure 15. As indicated in the horizontal units, we assigned different numbers of thread and block resources.

These experiments showed that when the block-thread combination was [8 blocks-256 threads], [8 blocks-1024 threads], and [12 blocks-256 threads], we could achieve high throughput performance, but only the 8-256 combination was energy efficient. This allocation not only ensured a throughput of 63 Mbps with a relatively high performance but also saved 132.92 J of energy at most. When different thread-scheduling combinations were adopted, the maximum energy cost was 4396.55 and the minimum energy cost was 3714.51, leading to a saving of up to 15.51%.

This can be attributed to the fact that when we use superfluous parallel threads, the throughput performance will not improve but will lead to increased power consumption. According to the fuzzy control model to select the appropriate hardware resource allocation scheme, we will be able to achieve higher performance at lower power consumption.

6  CONCLUSION
Packet classification is one of the most important components of network packet processing, which suffers from not only high performance
issues but also challenges on energy efficiency. In this paper, we focused on the GPU platform that can significantly accelerate rule-matching process, studied on reducing power consumption while keeping a high throughput and finally proposed an FCEEO model to achieve our objectives. As demonstrated in the evaluation results, by switching the computing mode of GPU according to the model-based analysis, we can save 10 W when GPU stays idle. Monitoring GPU temperature is useful to prevent a program interruption, because the high-temperature self-protection mechanism will shut down GPU when the temperature reaches 95°C. Furthermore, the proposed model also directs us to arrange thread configurations for kernel executing, through which the overall power consumption decreased by up to 15.5%. At the same time, the FCEEO model can keep a high throughput at the same level.

**ACKNOWLEDGMENTS**

This work is supported by the National Science Foundation of China under grant 61472130, the National Basic Research Program of China (973) under grant 2012CB315805, the Prospective Research Project under grant 61472130, the National Basic Research Program of China. This work is supported by the National Science Foundation of China.

**REFERENCES**