SMP Cluster: Reconstruct Bidirectional Sieve for Hybrid Parallelization

Gang Liao¹, Qi Sun¹, Lian Luo¹, Shi-Yuan Zhao¹, Sha Ding²

¹Computer Science and Engineering Department, Sichuan University Jinjiang College, 620860 Pengshan, China
²Software Engineering College, Sichuan University, 610065 Chengdu, China
greenhat1016@gmail.com, sq3737@gmail.com, mr.li7286481@gmail.com, fengxiangxp@163.com, dingsha7320@gmail.com

Abstract. This paper provides an overview and information useful for reconstructing bidirectional sieve for hybrid parallelization based on SMP Cluster, which is the individual computational units joined together by the communication network, are usually shared-memory system with one or more multicore processor. The main emphasis is placed on the problem of high-efficiency optimization based on the Cluster. We propose to average allocate the problem size into nodes and generate double-ended-queues (deque) so as to intranode simultaneously sifting out primes from deque’s head and tail. As well as, each node also create a FIFO queue as dynamic data buffer to ache temporary sieve operators from other nodes. As the result of experimental analysis, the reconstruct algorithm to combine these estimates and produce a high-efficiency model are presented and justified.

Keywords: bidirectional sieve, SMP Cluster, HPC, hybrid parallel

1 Introduction

Research into questions involving primes continues today, partly driven by the importance of primes in modern cryptography. As our computational power increases, researcher often pays more attention to data analysis, climate modeling, protein folding, drug discovery etc. We can also exploit multicores to efficiency solve some problem in the field of number theory.

M.Aigner and G.M.Ziegler [1] presented six quite different proofs of the infinitude of primes. Mills [2] has shown that there is a constant θ such that the function \( f(x) = [θ^x] \) generates only primes. The sieve of Eratosthenes-Legendre [3][4] is an ancient algorithm for finding all prime numbers up to any given limit. In number theory, tests distinguishing between primes and composite integers will be crucial. The most basic primality test is trial division, which tells us that integer \( n \) is prime if and only if it is not divisible by any prime not exceeding \( \sqrt{n} \).

In this paper, the ultimate goal of the experiment is to reconstruct the bidirectional sieve and exploit hybrid parallelization (i.e. MPI + OpenMP [5][6][7]) to perform high-efficiency and speedup. There is far-reaching significance in modern cryptography.
2 Reconstruct Bidirectional Sieve

When you try to reconstruct algorithm so as to parallelization and high-efficiency, there are three topics must be discussed as follows: Coverage or extent of parallelism in algorithm, Granularity of partitioning among processors and Locality of computation and communication. Back in the 1960s, Gene Amdahl [9] that’s become as Amdahl’s Law:

\[ \text{speedup} = \frac{\text{old running time}}{\text{new running time}} = \frac{1}{(1 - p) + \frac{p}{\pi}} \]  

The performance improvement to be gained from using some faster mode of execution is limited by the fraction of the time the faster mode can be used. A more mathematical version of this statement is known as Gustafson’s Law [10]. The common communication cost model denote as follows:

. f: frequency of messages
. o: overhead per message(at both ends)
. l: network delay per messages
. n: total data sent
. m: number of messages
. b: bandwidth along path(determined by network)
. t: cost induced by contention per message
. overlap: amount of latency hidden by concurrency with computation

\[ C = f * (o + l + \frac{n/m}{b} + t - \text{overlap}) \]  

Often, one can see in publications, that applications may or may not benefit from hybrid programming depending on some application parameters (e.g. in [12][13][14][15]). Polf Rabenseifner analyses strategies to overcome typical drawbacks of this easily usable programming scheme on systems with weaker interconnects [16]. Best performance can be achieved with overlapping communication and computation, but this scheme is lacking in ease of use.

2.1 Hybrid Parallel Algorithm Design

The sieve of Eratosthenes does so by iteratively marking as composite the multiples of each prime, starting with the multiples of 2 [4]. We can exploit and improve the sieve of Eratosthenes based on SMP Cluster.

Assume that there are some disorder integers which the scale of n, and when each node sieve the integers in the block that the scale of k, it could achieve high-efficiency optimization. We conjectured that the SMP Cluster requires at least N nodes. The formula as follows:

\[ N = \frac{n}{k} + (n \mod k) & 1 \]  

And each node generates one deque and does with dual-cores. One core is located in the head of the deque. On the contrary, the other one is located in the tail of the deque.
It’s easy to deduction the formula about the amount of cores and deques:

\[ C_{\text{cores}} = D_{\text{deques}} = 2N \]  

(4)

There is another point that’s worth considering. In most cases, the scale of node N is not exactly equal k. We can deal with the state as follows Algorithm I:

\begin{algorithm}
\caption{Algorithm I: the scale of node \( N^{th} \)}
\begin{algorithmic}
\Require \( K \) denote that the currency scale of node \( N^{th} \)
\Ensure \( \kappa \) denote that the general scale of node
\end{algorithmic}
\begin{algorithmic}
\If {\( 0 \leq K \leq \frac{k}{2} \)}
\State Node N assign single core to right or left sieve
\Else
\State Node N assign dual-cores to simultaneous bidirectional sieve
\EndIf
\end{algorithmic}
\end{algorithm}

To improve the efficiency of the method, the following approaching may be applied. Figure 2 shows the schematic diagram of the procedure.

**Fig. 2.** Hybrid Parallel Bidirectional Sieve implementation contains the procedure of each Node’s Double-Ended-Queue and FIFO Queue.

To start with, each dual-core node simultaneously sifts out numbers from the deque’s head and tail. If the number is prime, it would be passed to another node’s FIFO
queue, which is a dynamic data buffer. Over these steps, each node’s FIFO queue can get some primes from another node. Due to overlap communication with computation, the total costs become minimize. After that, before dual-cores exploit bidirectional sieve, it must check the FIFO queue. If the FIFO queue is empty, it has to get the primes from the deque. Similarly, on the contrary, it must get primes from the FIFO queue. Until the FIFO queue is empty that each node can use the primes from deque again. Foster’s methodology [8] [11] provides an outline of steps include:

- Partitioning
- Communication
- Agglomeration or aggregation
- Mapping for parallel programming

And its flow diagram is shown in Fig.3.

![Flow Diagram](image)

**Fig. 3.** High-level diagram of hybrid parallel bidirectional sieve
3 Experimental Analysis

Different programming schemes on clusters of SMPs show different performance benefits or penalties in this paper. We exploit 2.13GHz Quad-core Intel Xeon to build extensible SMP cluster.

In Fig.4 summarizes the result of hybrid parallel bidirectional sieve. It’s obvious that nodes communication would waste most of time when data scale is tiny. Even it is slower than general method. However, if there are big data, hybrid parallel show huge efficiency and optimization. Indeed, sometimes the waste of communication could be neglected. In that case, multicores parallelism is effective approach to solve some problem in number theory.

![Figure 4. Statistics and analysis hybrid parallel bidirectional sieve with general method](image)

4 Conclusion

In this study we have shown that hybrid parallel on SMP cluster is an applicable method to implement bidirectional sieve method. The analysis demonstrated that even hybrid parallel bidirectional sieve is efficiency and optimization solution. As our computational power increases, most HPC systems are clusters of shared memory nodes. Parallel programming must combine the distributed memory parallelization on the node inter-connect with shared memory parallelization inside of each node. And each parallel programming schema on hybrid architecture has one or more significant drawbacks (e.g. sleeping-thread and saturation problem). However, hybrid parallel also has far-reaching significance in many fields.

Some improvements to the reconstruct bidirectional sieve may be brought through additional levels in the hierarchy for more detailed representation of the hybrid parallelization. Thus, first extension of the approach could be, annotated or transformation to modify and eliminate dependences based on automatic parallelize compiler [17] [18].

We believe that hybrid parallel bidirectional sieve method can be properly modeled
using techniques from number theory and this article is just an early trial of using hybrid parallelism to improve speedup and efficiency.

References