ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

Building Parallel Applications Using Spatiotemporal Patterns

Mr. B P KolheM.E. – Student, Department of Computer Science Engineering,

Government College of Engineering, Aurangabad

Abstract: The exploration of design pattern concepts in the parallel programming domain is not something new. In fact, parallel computing is becoming an integral part in several major application domains, for instance: space, medicine, geological survey, cancer and genetic research, graphics and animation, image processing-to name a few. This paper addresses detail study on spatial data partitioning strategy.

Keyword: Design patterns, parallel computing, parallel programming, spatiotemporal pattern

I. Introduction

In software engineering, a design pattern is a general repeatable solution to a commonly occurring problem in software design. A design pattern isn't a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations. These patterns provide a set of general solutions which are based on recurring designs used by experts in the field. When engineers encounter a problem developing a parallel application, they can look for a pattern that provides insight and develops intuition for possible solutions. With the advent of fast interconnecting networks of workstations and PCs, it is now becoming increasingly possible to develop high-performance parallel applications using the combined computing powers of these networked -resources, at no extra cost. A design pattern is implemented as a reusable code Skelton for quick and reliable development of parallel applications. Developing a program that employee such complex parallel structure would requires significant amount of time and effort.

This paper reviewed to support the framework for parallelizing an application while considering throughput, latency or both with detailed case study. This framework pattern are called spatiotemporal portioning strategies because they distinguish between parallelizing an application's data or instructions and whether that parallelization is done in time or space

Three step processes is used to select a partitioning strategy given an application's performance goals.

Step- I: Defined spatiotemporal indices of an application's data and instruction.
Step- II: Using decision tree for selecting initial strategy.
Step - III: (Optional) iteratively applying multiple strategies if single partitioning strategy is fails or too complicated for implementation

II. BASIC CONCEPT AND DEFINATION

The term program or application is used to refer to the problem to be decomposed into concurrent parts. A process is the basic unit of program execution, and a parallel program is one that has multiple processes actively performing computation at one time. Programs operate by executing instructions to manipulate data. Both the data and instructions of a program have spatial and temporal components because programs execute over time while reading and writing space (memory).

In order to partition the execution of a program, one must first define the spatial and temporal dimensions of the program's execution. Programs are parallelized to increase their performance. The performance of a parallel program can be further broken down into throughput, which measures the rate of computation, and latency, which measures the time taken to perform a component of the total computation. Often, these values are reported in terms of inputs; throughput is measured in inputs per time unit, while latency is measured in terms of the average time taken to process an individual input. Typically, a program is parallelized to increase throughput, decrease latency, or both [1].

.

III. PARTITIONING STRATEGIES

In this section we are defining the four partitioning strategies: spatial data partitioning (SDP), temporal data partitioning (TDP), temporal instruction partitioning (TDP), and spatial instruction partitioning (SIP).

The methodology for choosing a partitioning strategy consists of several distinct steps. First, one defines the spatiotemporal indices of the program's instructions and data. Second, a decision tree is used to determine the best strategy to meet the application's throughput and latency goals. Third, if no single strategy appears to provide a good solution one can iteratively apply multiple strategies to break

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

a complicated program into several parts that are individually simpler to deal with. [1]

A. Spataial Data Partitioning

With the help of spatial data partitioning strategy, data are divided among processes according to spatial data index. Generally each process performs all assigned instruction.

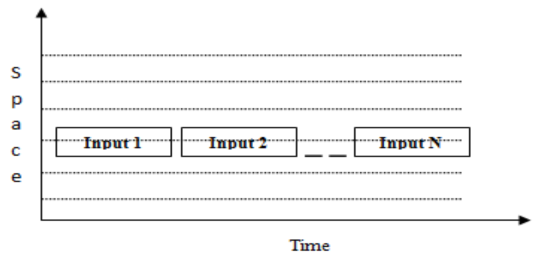


Fig. 3.1 SDP

B. Temporal Data Partitioning

Using the temporal data partitioning strategy, data are divided among processes according to temporal index.

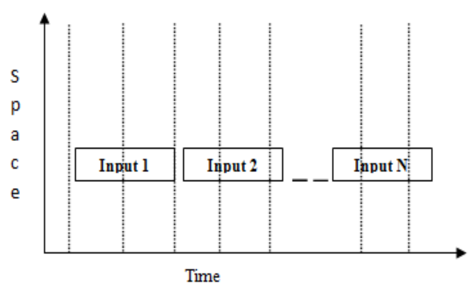


Fig. 3.2 TDP

C. Spatial Instruction Partitioning

Using the spatial instruction partitioning strategy, instructions are divided among processes according to spatial instruction index.

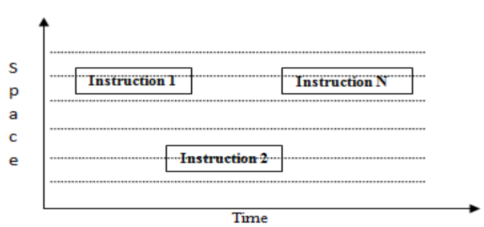


Fig. 3.3 SIP

D. Temporal Instruction Partitioning

Using the temporal instruction partitioning strategy, instructions are divided among processes according to temporal data index.

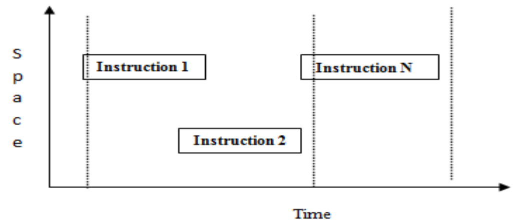


Fig. 3.4 TIP

IV CASE STUDY

In this section present detailed case study showing how the spatiotemporal partitioning strategies and the accompanying selection methodology can guide the development of parallel programs

Arsenic in Ground Water of the United State

In this case study we are using a dataset arsenic_may2000 of U.S. Department of Geological Survey related to Arsenic in Groundwater of the United States. This dataset is intended to represent the potable ground-water resource. This dataset therefore does not include thermal and saline water (temperature greater than 50 degrees C or dissolved solids greater than 3000 mg/L or specific conductance greater than 4000 uS/cm). In addition, this dataset includes only the most recent arsenic analysis available for each well, and only analyses performed by hydride generation or ICP/MS.

A. Background Information:

Arsenic is a naturally occurring trace element found in rocks, soils, and the waters in contact with them. Arsenic has long been recognized as a toxic element and is also considered a human health concern because it can contribute to skin, bladder, and other cancers. Arsenic concentrations are measured in units of micrograms per liter (µg/L), which is equivalent to parts per billion. This map layer was compiled by the U.S. Geological Survey, National Water-Quality Assessment Program (NAWQA), which is responsible for developing long-term, consistent, and comparable information on streams, ground water, and aquatic ecosystems. This information supports national, regional, State, and local water-management and policy decisions that protect drinking water and other water resources, as well as public health. NAWQA information on arsenic in ground water is used by the U.S. Environmental Protection Agency (USEPA) to help set national standards for arsenic in drinking water, as mandated by the Safe Drinking Water Act. In 2001, the USEPA lowered the maximum level of arsenic permitted in drinking water from 50 µg/L to 10 µg/L .[11]

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

Fig. 4.1 Arsenic in groundwater of the United States [11]

B. Basic Dataset Element

STAID: USGS station identifier; based on latitude and longitude (each well in the National Water Information System has a unique USGS station identifier).
STATE: Two-letter postal code for the U.S. state in which the sample was collected.
FIPS: Federal Information Processing Standard state and county codes

LAT_DMS: Latitude of well, in degrees, minutes, and seconds
LONG_DMS: Longitude of well, in degrees, minutes, and seconds
WELLDPTH: Depth (below land surface) of finished well
SAMPDATE: Date water sample was collected
SAMPTIME: Time water sample was collected
AS_RMRK: Remark code qualifying the analytical result in AS_CONC.
AS_CONC: Concentration of arsenic in sample, in micrograms per liter (ug/L) as arsenic

C. Spatiotemporal Partioning Strategies

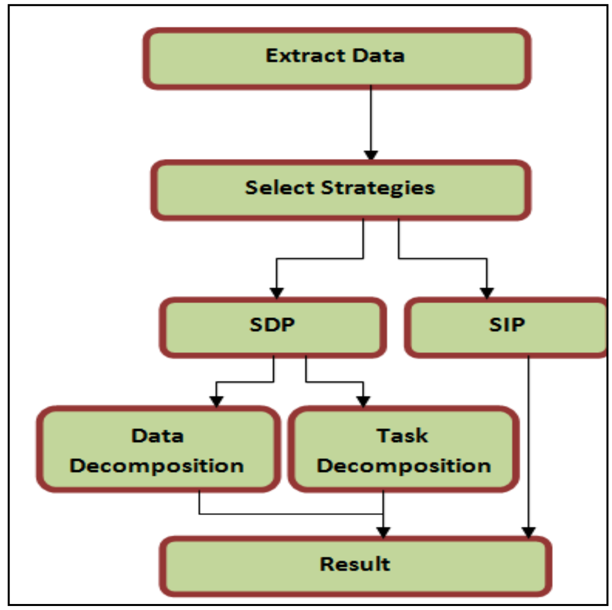
To apply the partitioning strategies, a program's temporal and spatial indices must first be defined. The following procedure determines the spatiotemporal indices of a program's instructions and data to prepare for partitioning

1. Determine what constitutes an input to define the temporal dimension of the program's data. The sequence of inputs represents the temporal data index.

2. Determine the distinct components of an individual input to define the spatial dimension of the Program's data. These components represent the spatial data index.

3. Determine the distinct functions required to process an input to define the spatial dimension of the program's instructions. The distinct functions represent spatial instruction indices.

4. Determine the partial ordering of function execution required to process a single input. [1]



5. Fig. 4.2 Proposed system archetecture

Using the spatial data partitioning strategy, data are divided among the processes according to spatial data index. To implement SDP strategy on dataset arsenic_may2000.

Step I: First proposed system extract the arsenic_may2000 dataset with all key element in Fig. 4.3

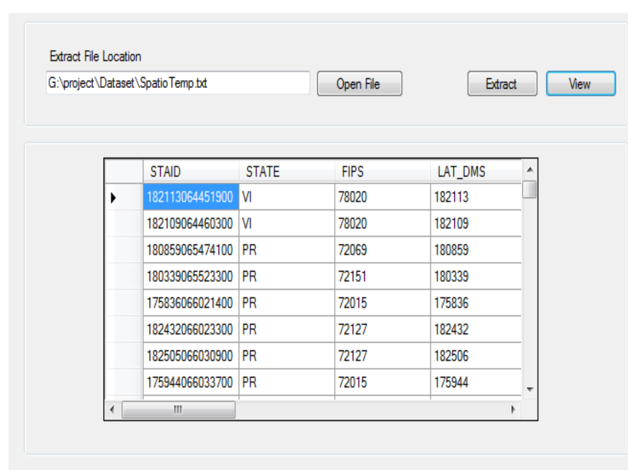


Fig. 4.3 Extract Arsenic dataset

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

Step II : A single water sample is an input, so the sequence of samples represent the temporal dimesnsion of the arsenic dataset and each sample is composed of distinct element STATEID with STATE element.

The spatial data partitioning strategy can be implemented using either DataDecomposition or TaskDecomposition. To implement this strategy with Data Decomposition, simply assign each process a separate set of spatial data indices. To implement the strategy with TaskDecomposition, each task should be defined to execute all instruction indices on data from a single spatial data index

To implement the SDP strategy with data decomposition in the arsenic data, separate process work simultaneously on distinct component for example to analysis well depth water level in all state for this purpose two process contain two separate set of spatial data indices one set contain all STATE and another contain all WELLDEPTH in Fig. 4.4

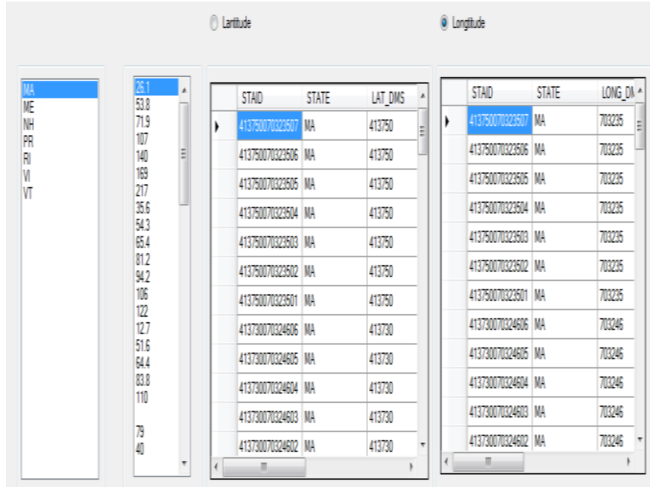


Fig. 4.5 Task Decomposition

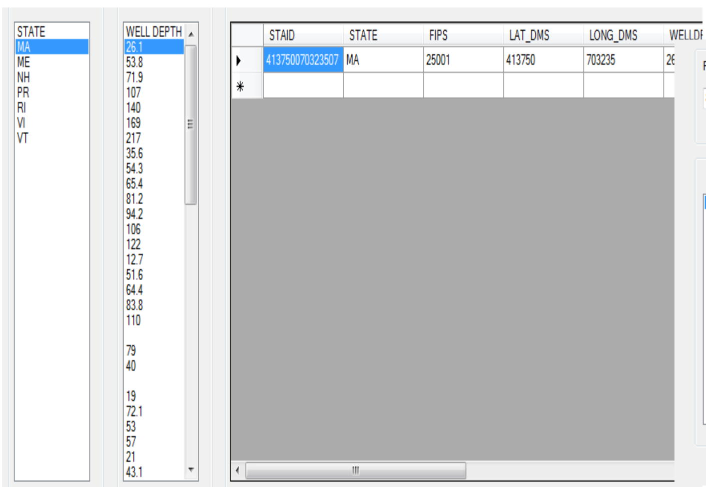


Fig. 4.4 Data Decomposition

Step III : With the help of Task Decomposition , each task should be defined to execute all instruction on data from spatial index.For example we are find the missing level of in depth so that task can separate two spatial index STATE and WELLDEPTH in Fig. 4.5

StepIV :Using the spatial instruction partitioning strategy , instructions are divided among processes according to spatial instruction index. To implement SIP strategy on dataset arsenic_may2000 based on the Longitude and Longitude to find the SD between range in Fig. 4.6

StepV : Fig. 4.7 show the latency of the proposed system parallelized with the SDP strategy. The figure clearly shows how using SDP can improve latency for this system.

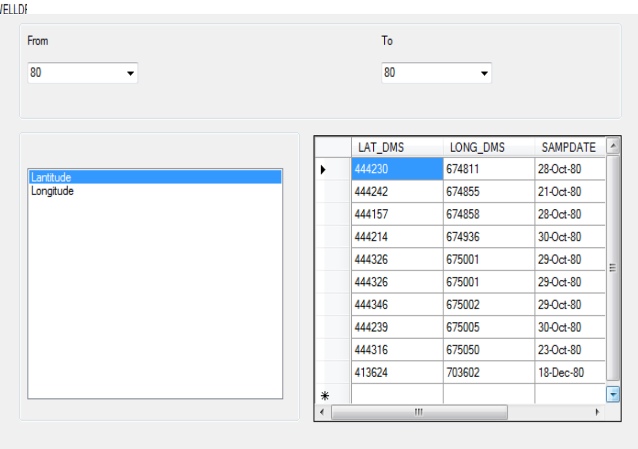


Fig. 4.6 SIP

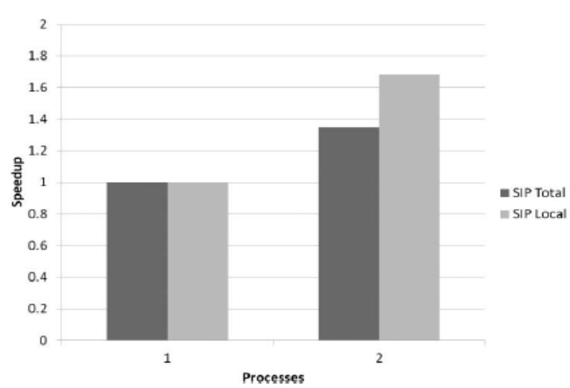


Fig. 4.7 Latency of Proposed system

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 5, May 2013

V CONCLUSION

This paper presents a generic model for designing and developing parallel applications, and is based on the idea of spatiotemporal patterns. The model is an ideal candidate for implementation parallel application using any four mentioned strategy. Normally for parallel computing domain expert application designer are required, but with the help of this model parallel computing accessible to non expert people.

VI ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude to my guide Prof. S. G. Shikalpure for their valuable and firm suggestions, guidance, encouragement and constant support throughout this work without which it would not be possible to do this work. They took deep interest in checking the minute details of the paper and guided me throughout the same. He has been a constant source of inspiration. I also feel a deep sense of gratitude to Prof. V. P. Kshirsagar, HOD of Computer Science and Engineering department for their continuous cooperation, guidance and encouragement. I am also sincerely thankful to Dr.P.S.Advani, Principal, Government College of Engineering, Aurangabad for being source of motivation for this work.

REFERENCES

[1] H. Hoffmann, A. Agarwal, S. Devadas, Selecting Spatiotemporal Patterns for Development of Parallel Applications. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 10, OCTOBER 2012.

[2] M. Snir, "Parallel Programming Patterns," http://www.cs.uiuc. edu/homes/snir/PPP/, 2012.

[3] H. Hoffmann, A. Agarwal, and S. Devadas, "Partitioning Strategies: Spatiotemporal Patterns of Program Decomposition," Proc. 21st Int'l Conf. Parallel and Distributed Computing and Systems, 2009

[4] x264, http://www.videolan.org/x264.html, 2011

[5] Berna L. Massingill, Timothy G. Mattson, Beverly A. Sanders, "Patterns for Parallel Application Programs"

[6] S Siu, M Simone, A Singh, "Design Pattern for parallel programming"

[7] C. Bienia, S. Kumar,J Singh,Kai Li, "The PARSEC Benchmark Suite: Characterization and Architectural Implications"

[8] T. Purcell, I. Buck, W. Mark, P. Hanrahan, "Ray Tracing on Programmable Graphics Hardware"

[9] P. Christensen, J Fong, D. Laur, D. Batali, "Ray Tracing for the Movie 'Cars'"

[10] "Supra-Linear Packet Processing Performance with Intel Multi-Core Processors," technical report, Intel, 2006

[11] "National water-quality assesment (NAWQA) program , U.S. Department of the Interior , U.S. Geological Survey "

[12] Martin Erwig ,"Toward Spatiotemporal Patterns", School of Electrical Engineering and Computer Science, Oregon State University,

Corvallis, Oregon 97331, USA

[13] Berna L. Massingill, Timothy G. Mattson, Beverly A. Sanders, "Patterns for Parallel Application Programs", University of Florida

[14] R. van de Geijn,Using PLAPACK—Parallel Linear Algebra Package.

MIT Press, 1997.

BIOGRAPHY

Babasaheb P. Kolhe received his B.E. degree in Information Technology from P.E.S. College of Engineering, Aurangabad, India, in 2006 and pursuing M.E. degree in Computer Science and engineering from Government College of engineering, Aurangabad, India. His research interest includes Parallel application and Spatial Data Processing.