NEW BOOK PROPOSAL

AUTHOR AND TITLE INFORMATION

1. Tentative book title and subtitle (if any):

Parallel Computation: A Fundamental Approach

1. Authors/Editors full name/s:

Sean Halle

1. Position and affiliation:

ERCIM Fellow, CWI, Netherlands (Centrum Wiskunde & Informatica)

1. Full mailing address/es:

Science Park 123
1098 XG Amsterdam
Netherlands

5. Authors/Editors phone, fax, email:

Name	Telephone:	Fax	Email
Sean Halle	+31 616434534		seanhalle@yahoo.com

6. If an edited book, approximately how many contributors do you expect to comprise the book?

N/A

SUBJECT MATTER

7. a. Definition of topic (in laymen's terms—so that a non-scientist can understand why there would be a market-need for this sort of content):

The topic is the fundamental mechanisms that underly parallel computation.

b. Short "blurb"/brief description of YOUR book which could be used for promotional purposes

In order to be effective within the changing computation world that is now dominated by parallel hardware, one needs a basic understanding of the mechanisms at work inside parallel computation. The traditional approach to learning parallel computation has simply been to throw the learner into the deep end: "here's some parallel hardware, here's a parallel language, write code!" The books available on the subject simply state a variety of different parallel hardware architectures and a variety of different parallel languages or programming models, and then how to use them, but nothing about what's behind it all. In Parallel Computation: A Fundamental Approach, the reader will gain a conceptual introduction to just what it is that makes parallel computation tick – a blueprint, as it were, a map. In other words, a distillation of the invariant patterns that underlie parallel computation, and hold true no matter what the context.

The basic understanding gained in Parallel Computation: A Fundamental Approach carries across to every parallel language, every parallel program, and every form of parallel hardware. Armed with this blueprint, the reader who wishes to start parallel programming will see the common patterns that are taught in the book within whatever language or programming-model that they decide to learn. The reader who wishes to design parallel hardware will gain insight into classes of application, and how their design choices are going to affect each of the application types. The reader who wishes to design parallel languages will gain an understanding of how their language constructs/programming model affect the programmer, the applications it will be suitable for, the toolchain, and what implications the constructs will have for being mapped onto parallel hardware.

Parallel Computation: A Fundamental Approach is, in effect, the training wheels that guide the reader into the world of parallel computation. It is the rosetta stone that explains "how it all works". The reader comes away with a basic understanding that applies to every parallel computation context.

c. Outline your reasons for proposing a new book in this area:

There don't exist viable texts that give a fundamentals oriented approach to what are the essential mechanisms at work inside parallel computation. Existing texts state "here's what exists", but don't state "here's what's behind what exists". For example, books on parallel programming simply state the syntax and semantics of parallel programming languages and walk the reader through the writing of a parallel program. But they don't give an invariant conceptual model that applies to all parallel programs, and all parallel languages and all parallel hardware. This book will fill in this gap.

d. List several unique features of your book which will attract prospective buyers:

-] Gaining a basic, fundamental understanding of the mechanisms at work inside parallel computation.
-] Conceptual tools to guide the writing of parallel applications
-] Conceptual tools to guide design and implementation of infrastructure to support parallel software
-] Conceptual tools to guide design and implementation of new parallel languages/programming models
-] Conceptual tools to guide design and implementation of new parallel hardware

e. What are the benefits of this book for the reader?

In order to be effective within the changing computation world that is now dominated by parallel hardware, one needs a basic understanding of the mechanisms at work inside parallel computation. The traditional approach to learning parallel computation has simply been to throw the learner into the deep end: “here's some parallel hardware, here's a parallel language, write code!” The books available on the subject simply state a variety of different parallel hardware architectures and a variety of different parallel languages or parallel programming models, and little more. What does not currently exist, in a viable form, is a conceptual introduction to just what it is that makes parallel computation tick – a blueprint, as it were, a map. In other words, a distillation of the invariant patterns that underlie parallel computation, and hold true no matter what the context.

The benefit to the reader of this proposed book will be the gaining of a basic understanding that carries across to every parallel language, every parallel program, and every form of parallel hardware. Armed with this blueprint, the reader who wishes to start parallel programming will see the common patterns that are taught in the book within whatever language or programming-model that they decide to learn. The reader who wishes to design parallel hardware will gain insight into classes of application, and how their design choices are going to affect each of the application types. The reader who wishes to design parallel languages will gain an understanding of how their language constructs/programming model affect the programmer and what implications they will have for being mapped onto parallel hardware.

In the end, this book will be, in effect, the training wheels that guide the reader into the world of parallel computation, starting from the traditional world of sequential computation. It will be the rosetta stone that explains “how it all works”. The benefit from this is the coming away with a basic understanding that applies to every parallel computation context.

f. Attach proposed table of contents/chapter outline, including chapter headings:

Part I: The problem (if you don't understand the problem, you can't understand the solution!)

Chapter 1: So you want to plan a party? (parallel execution in a real world setting)

Chapter 2: Strategies for organizing the chaos (what is a runtime system, anyway?)

Chapter 3: Our party plan in terms of some common parallel programming models

Part II: The Fundamentals

Chapter 4: The three elements at the base of every (parallel) computation (patterns underneath our party plan and how it unfolds)

1. Work Units and Constraints
2. Work Animators and communication consequences
3. Active mapper of Work Units onto Work Animators, given constraints and consequences

Chapter 5: The constraints on execution of work units (what is inherent in the problem and what is artifact)

Chapter 6: The UCC and SCG: visualizing the constraints and how they unfold

Chapter 7: Animators – virtual versus physical, and consequences of assignment of work

Chapter 8: The Runtime system: mapping Application constraints onto hardware consequences

Chapter 9: Organizing it all: classes of application -- according to characteristics that affect the runtime and interact with the hardware

Chapter 10: Organizing it all: classes of hardware – according to consequences for application classes

Chapter 11: Organizing it all: classes of runtime – according to combinations of application class and hardware class

Chapter 12: Advanced topic: Hierarchy of UCC and SCG – how communication hierarchy relates to scheduling hierarchy

Part III: Applying the Fundamentals

Chapter 13: Analyzing a basic problem according to work units and constraints

Chapter 14: Using the problem analysis result – choosing programming models, and consequences of choice

Chapter 15: Choosing hardware for a given problem – how analysis result, programming model, and runtime system characteristics influence which hardware will be most efficient, and tools for making the choice

Chapter 16: Choosing runtime system – managing the inter-dependencies among application characteristics, choice of programming model, hardware characteristics, and which runtime system strategies best map application onto hardware

Chapter 17: Designing parallel languages – what the fundamentals say about the goals of parallel languages and paths to reach those goals (IE, specialize language to domain, intermix languages, supply toolchain needs for automated specialization, toolchain harness, automated distribution)

Chapter 18: Putting it all together – a vision for the future of parallel computation, driven by the fundamentals described herein.

MANUSCRIPT INFORMATION

8. Approximately how many manuscript pages do you expect your book to be?

200 to 300 pages

Manuscript Delivery Date

9. How long do you estimate it will take for completion of the entire manuscript?

1 year, possibly 2

COMPETITION

10. Please list in order of importance any books that compete directly with or are similar to your book. Please supply (if possible) author/editor, publisher, publication date/year, price and any further information you feel is relevant (use separate sheet, if necessary):

Please note: if you are inclined to answer "none" to this question, please tell us where your intended audience currently gets information about this topic…e.g., conferences, tutorials, journal articles, web forums, etc…please be as specific as you can.

There are a great many books on this topic. Unfortunately, they all take the same approach, as epitomized by this blurb on Amazon.com regarding the book "Introduction to Parallel Computing"

“Introducation to Parallel Computing is a complete end-to-end source of information on almost all aspects of parallel computing from introduction to architectures to programming paradigms to algorithms to programming standards. It is the only book to have complete coverage of traditional Computer Science algorithms (sorting, graph and matrix algorithms), scientific computing algorithms (FFT, sparse matrix computations, N-body methods), and data intensive algorithms (search, dynamic programming, data-mining). “

Another example is “Parallel Computing: Principles and Practices” in which they use the word “principles,” but they don't actually state principles, but rather provide a survey of what exists. Here's the blurb (note how it winds up just listing things that the book describes, but nowhere mentions fundamental invariant concepts that cross all those boundaries):

“This book sets out the principles of parallel computing in a way which will be useful to student and potential user alike. It includes coverage of both conventional and neural computers. The content of the book is arranged hierarchically. It explains why, where and how parallel computing is used; the fundamental paradigms employed in the field; how systems are programmed or trained; technical aspects including connectivity and processing element complexity; and how system performance is estimated (and why doing so is difficult). The penultimate chapter of the book comprises a set of case studies of archetypal parallel computers, each study written by an individual closely connected with the system in question. The final chapter correlates the various aspects of parallel computing into a taxonomy of systems.”

There are a great many books on the subject, but all end up doing the same thing – giving a survey that describes concrete details and states what is done in practice. Even when they have “principles” or “fundamentals” in the title, they still end up just providing a survey.

11. Outline in what ways your book is better than and differs from the competitors mentioned in Section 10:

As mentioned in the other questions, this book will provide the fundamental underpinnings, while the others only survey descriptions and experiences.

THANK YOU FOR CONSIDERING SPRINGER

Charles B. Glaser
Senior Acquisitions Editor
Springer
233 Spring Street
New York, NY 10013
(781)535-4719