

СПЕЦИФИКАЦИЯ МОДЕЛИ ПАРАЛЛЕЛЬНЫХ И РАСПРЕДЕЛЕННЫХ ВЫЧИСЛЕНИЙ GRAPHPLUS НА ОСНОВЕ ЛОГИКИ TLA

© 2006 С.В. Востокин

Самарский государственный аэрокосмический университет

В статье описывается структурная модель вычислительных процессов GraphPlus и ее формальная спецификация на основе темпоральной логики Лампорта TLA. Рассматриваемая модель предназначена для формализации семантики параллельных и распределенных алгоритмов, используемых при реализации сложных численных моделей.

Введение

Параллельные и распределенные вычислительные процессы сложны для понимания. В силу ограниченных возможностей человеческого восприятия трудно предвидеть все особенности, возникающие в процессах, одновременно протекающих в разных точках пространства. Имеется множество специфических ошибочных ситуаций (тупики, гонки, отталкивания), которые выявляются только сложным анализом. Поэтому актуальной является задача построения формальных моделей, облегчающих анализ и конструирование параллельных и распределенных процессов и порождающих их алгоритмов и программ.

В области разработки прикладных параллельных алгоритмов развиваются два направления. Первое направление основано на построении модели по информационно-логической структуре конкретного численного алгоритма на этапе его распараллеливания [1]. Второе – на обобщении свойств вычислительных процессов для различных по назначению алгоритмов численного моделирования [2]. В любом случае требуется однозначное понимание особенностей поведения процесса. В статье описывается семантика вычислительной модели GraphPlus, которая используется в рамках второго подхода к синтезу параллельных процессов вычислительных алгоритмов [3].

Неформальное определение вычислительной модели GraphPlus

Модель GraphPlus является примером гра-

фической объектной модели, представляющей собой граф, в вершинах которого находятся объекты разных типов, а связи передают статические отношения между объектами.

Структура произвольного параллельного алгоритма, пример которой представлен на рис. 1, образована объектами двух типов: “неподвижными” или Р-объектами и “подвижными” или М-объектами.

Р-объект (“неподвижный” объект) в модели вычислений является функциональным аналогом процесса. Все действия, связанные с Р-объектом, выполняются последовательно. Состояние Р-объекта, хранимое в его переменных-членах, обычно соответствует некоторой области данных, за обработку которой он отвечает. Состоянием Р-объекта может быть, например, сегмент из точек сеточной области в термодинамических расчетах.

М-объект (“подвижный” объект) в модели вычислений является функциональным аналогом сообщения, при помощи которого взаимодействуют процессы (Р-объекты). Состояние М-объекта, хранимое в его переменных-членах, например, может соответствовать точкам на границах сегментов сеточной области в термодинамических расчетах.

Вычисления на модели интерпретируются как одновременный обход несколькими М-объектами структуры из Р-объектов. Обход выполняется по связям, обозначенным сплошными линиями. Текущее состояние объектов на рис. 1 обозначается при помощи задания взаимного пространственного расположения и способом прорисовки контура. В процессе вы-

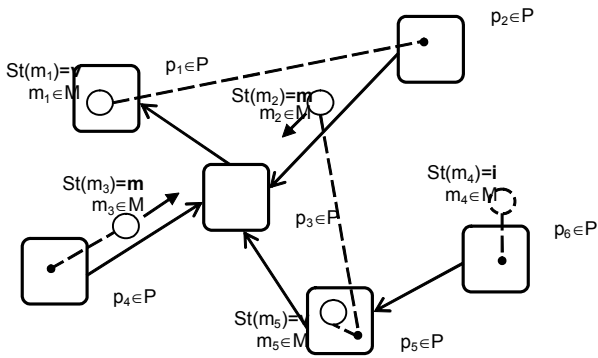


Рис. 1. Мгновенный снимок структуры параллельного вычислительного процесса в модели GraphPlus

числений допустимы следующие состояния объектов модели. М-объект может находиться в неактивном состоянии $St[m] = 'i'$. На рис. 1 это объект m_4 , обозначенный пунктирным контуром. Активные состояния М-объектов $St(m) = 'm' \vee St(m) = 'v'$ на рис. 1 обозначаются сплошным контуром. Находясь в активном состоянии, М-объект может либо перемещаться по связям между Р-объектами (состояние перемещения $St(m) = 'm'$ показано стрелками), либо посещать Р-объект $St(m) = 'v'$ (кружок содержится внутри квадрата – состояние посещения).

В процессе описанного выше взаимодействия Р-объект может находиться в двух состояниях: состоянии посещения и свободном состоянии. Гарантируется исключительно парное взаимодействие Р- и М-объектов. Каждый Р-объект в любой момент времени наблюдения посещается не более чем одним М-объектом.

С каждым объектом типа Р или М дополнительно связаны переменные, которые хранят состояние объекта, определяемое конкретным численным алгоритмом. Это состояние изменяется только в процессе “посещения”. После посещения некоторого объекта p М-объект может направиться к любому из соседних объектов $p' : N(p, p') = 1$, снова посетить только что посещенный объект p или перейти в неактивное состояние $St(m) = 'i'$. В процессе посещения возможна активация любого неактивного М-объекта, связанного с посещаемым Р-объектом x .

Метод описания дискретных систем с использованием темпоральной логики

Для формализации представленного описания вычислительной модели воспользуемся специальной темпоральной логикой TLA (temporal logic of actions), предложенной Лампортом [4,5]. Удобство TLA при формализации модели GraphPlus заключается в использовании переменных для представления состояния вычислений (как в традиционных языках программирования) и возможности однозначного сопоставления модели в терминах TLA со структурным представлением в виде графа рис. 1.

Пусть вычислительный процесс представлен множеством своих историй ss : бесконечных последовательностей дискретных состояний s , причиной изменения которых во времени являются действия A

$$s_0 \xrightarrow{A_1} s_1 \xrightarrow{A_2} s_2 \xrightarrow{A_3} s_3 \xrightarrow{A_4} \dots \quad (1)$$

Моменты наблюдения состояний связаны с отсчетами дискретного времени. Известно, что, не уменьшая выразительной силы описания важных свойств реальных дискретных систем, можно ограничиться рассмотрением только бесконечных историй, порожденных последовательностью неделимых (атомарных) действий из некоторого конечного множества. Тогда модель системы в темпоральной логике — это формула, смысл которой определен в терминах историй (1). Интерпретация формулы модели F (обозначается символом $[[\cdot]]$) состоит в сопоставлении некоторой истории значения истина или ложь:

$$\langle s_0, s_1, s_2, \dots \rangle [[F]] \equiv \sigma [[F]] : St^\infty \rightarrow \{И, Л\}. \quad (2)$$

Множество историй, для которых интерпретация формулы (2) истинна, однозначно определяет дискретную систему в темпоральной логике. Поэтому формула F является математическим описанием дискретной системы, эквивалентным множеству историй (1).

Связь между состоянием, множеством переменных и множеством значений переменных модели передается в виде

$$s_n : Var \rightarrow Val. \quad (3)$$

То есть под состоянием понимается функция (3), отображающая множество переменных Var на множество значений Val . Математические выражения, построенные из переменных, знаков математических операций, знаков отношений, логических связок, кванторов и т.п. называются функциями состояния и интерпретируются согласно выражению

$$s[[f]] \stackrel{\text{def}}{=} f(\forall v'v': s[[v]]/v). \quad (4)$$

Выражение (4) означает, что если вместо каждого вхождения переменной в функцию состояния подставить ее значение, то получится интерпретация для состояния s . Специальным случаем функции состояния является предикат состояния. Предикат состояния – функция состояния, принимающая значения истина или ложь.

Действие – это выражение, аналогичное функции состояния по своей структуре, которое дополнительно включает переменные с символом апострофа ($'$), принимает значения истина или ложь и интерпретируется в виде

$$s[[A]]t \stackrel{\text{def}}{=} A(\forall v'v': s[[v]]/v, t[[v]]/v'). \quad (5)$$

Таким образом, согласно (5) формула для действия дает ответ на вопрос: могут ли два состояния s и t быть соседними в некоторой истории вида (1).

Темпоральные формулы интерпретируются согласно (2) для историй. Поэтому вводится специальный модальный оператор (\square) “всегда”, смысл которого определяется через интерпретацию следующим образом:

$$\sigma[[\square F]] \stackrel{\text{def}}{=} \forall n \in \mathbb{N} : \langle s_n, s_{n+1}, s_{n+2}, \dots \rangle [[F]]. \quad (6)$$

Структуру формулы F составляют функции состояния и действия, поэтому для применения выражения (6) для интерпретации любых темпоральных формул смысл функций состояния и действий расширяют на бесконечные истории следующим образом:

$$\langle s_0, s_1, s_2, \dots \rangle [[A]] \stackrel{\text{def}}{=} s_0[[A]]s_1,$$

$$\langle s_0, s_1, s_2, \dots \rangle [[P]] \equiv s_0[[P]].$$

Тогда описывать бесконечные истории можно на основе следующих интерпретаций формул, построенных из действий и состояний с использованием модального оператора \square “всегда”:

$$\langle s_0, s_1, s_2, \dots \rangle [[\square A]] \stackrel{\text{def}}{=} \forall n \in \mathbb{N} : \langle s_n, s_{n+1}, s_{n+2}, \dots \rangle [[A]] \equiv \forall n \in \mathbb{N} : s_n[[A]]s_{n+1},$$

$$\langle s_0, s_1, s_2, \dots \rangle [[\square P]] \equiv \forall n \in \mathbb{N} : s_n[[P]].$$

Семантика логической основы формулы F передается выражениями

$$\sigma[[\neg F]] \stackrel{\text{def}}{=} \neg \sigma[[F]],$$

$$\sigma[[F \wedge G]] \stackrel{\text{def}}{=} \sigma[[F]] \wedge \sigma[[G]]$$

и распространяется на другие логические связки и кванторы очевидным образом по их определению.

Таким образом, используя традиционную математическую нотацию, расширенную знаками ($'$) и (\square), можно описать множество историй вычислительного процесса.

Специализация темпоральной логики по Лампорту

Для практически важных дискретных систем, в том числе для вычислительных процессов, требуется рассматривать еще более узкий класс множеств всех историй, а именно историй инвариантных по отношению к преобразованию дублирования одного состояния в соседних отсчетах времени (stuttering). Синтаксически это ограничивает рассмотрение всех темпоральных формул только формулами, сводимыми к виду (7), которые и называются формулами TLA.

$$\Phi \stackrel{\text{def}}{=} I \wedge \square[N]_f \wedge F. \quad (7)$$

Формула (7) имеет следующую структуру: I – предикат состояния, описывающий начальное состояние (вычислительного) процесса; N – действие, задающее множество всех атомарных операций системы; множество формул вида WF и SF определенных для подмножества атомарных операций из N . До-

полнительные синтаксические элементы TLA определяются через введенные выше обозначения посредством соотношений (8)-(18).

$$s[[Enabled\ A]] \stackrel{def}{=} \exists t \in St : s[[A]]t \quad (8)$$

$$f \stackrel{def}{=} (v_1, v_2, v_3, \dots) \quad f' \equiv (v_1, v_2, v_3, \dots)' \equiv (v'_1, v'_2, v'_3, \dots) \quad (9)$$

$$unchanged(f) \stackrel{def}{=} f = f' \quad (10)$$

$$[A]_f \stackrel{def}{=} A \vee (f = f') \equiv A \vee unchanged(f) \quad (11)$$

$$\langle A \rangle_f \stackrel{def}{=} A \wedge (f \neq f') \equiv A \wedge \neg unchanged(f) \quad (12)$$

$$\diamond F \stackrel{def}{=} \neg \square \neg F \quad (13)$$

$$\langle s_0, s_1, s_2, \dots \rangle [[\diamond F]] \equiv \exists n \in \mathbb{N} : \langle s_n, s_{n+1}, s_{n+2}, \dots \rangle [[F]] \quad (14)$$

$$\langle s_0, s_1, s_2, \dots \rangle [[\square \diamond A]] \equiv \forall n \in \mathbb{N} : \exists m \in \mathbb{N} : s_{n+m} [[A]]_{s_{n+m+1}} \quad (15)$$

$$\langle s_0, s_1, s_2, \dots \rangle [[\diamond \square A]] \equiv \exists n \in \mathbb{N} : \forall m \in \mathbb{N} : s_{n+m} [[A]]_{s_{n+m+1}} \quad (16)$$

$$WF_f(A) \stackrel{def}{=} (\square \diamond \langle A \rangle_f) \vee (\square \diamond \neg Enabled \langle A \rangle_f) \quad (17)$$

$$SF_f(A) \stackrel{def}{=} (\square \diamond \langle A \rangle_f) \vee (\diamond \square \neg Enabled \langle A \rangle_f) \quad (18)$$

Подробное объяснение определений (8)-(18) и их словесная интерпретация приведены в работе [4]. Для понимания модели GraphPlus существенным является смысл выражений (11), (17) и (18). Определение (11) читается “либо выполнено действие A , либо состояние системы, описываемое переменными f , не изменилось”. Определение (17) называется “справедливостью в слабом смысле” (weak fairness). Оно читается следующим образом: “если, начиная с некоторого момента времени выполнить действие A возможно, и эта возможность сохраняется неопределенно долго, то действие A будет выполнено неопределенно много раз”. Определение (18) называется “справедливостью в сильном смысле” (strong fairness). Оно, в отличие от (17), в качестве условия выполнения требу-

ет, чтобы, начиная с некоторого момента времени, действие можно было бы выполнить неопределенно много раз. Из смысла (17)-(18) очевидно следует назначение условий F в формуле (7) – при сохранении инвариантности к stuttering–преобразованию исключить из описания истории, в которых N может выполняться, но не выполняется бесконечное число отсчетов дискретного времени.

Спецификация модели GraphPlus

В силу того, что модель GraphPlus вводит специальные семантические понятия объектов, обладающих состоянием, потребуется способ связи объектов с множеством переменных состояния. Для этого введем функции, сопоставляющие объект модели GraphPlus переменной:

$$o[] : Obj \rightarrow Var. \quad (19)$$

Будем считать, что все переменные состояния в формулах модели вычислений заданы в виде (19), а вспомогательные переменные (константы TLA) заданы обычным способом.

Определим множества объектов модели GraphPlus следующим образом:

$$\begin{aligned} P \subset Obj \quad M \subset Obj \quad \varepsilon \in Obj; \\ P \cap M = \emptyset, \quad P \cap \{\varepsilon\} = \emptyset, \quad M \cap \{\varepsilon\} = \emptyset; \\ P_+ = P \cup \{\varepsilon\}, \quad M_+ = M \cup \{\varepsilon\}. \end{aligned}$$

Здесь ε – описывает специальный “нуль” объект, а P_+ и M_+ множества P - и M -объектов, расширенные “нуль” объектом. Обозначим домены значений объектов модели, как $D_P \subset Val$ и $D_M \subset Val$. Каждому M -объекту будут соответствовать переменные модели, задаваемые функциями $\tilde{m}[] : M \rightarrow V_{\tilde{M}} \quad m[] : M \rightarrow V_M$, а каждому P -объекту – функциями $\tilde{p}[] : P \rightarrow V_{\tilde{P}} \quad \tilde{n}[] : P \rightarrow V_N \quad p[] : P \rightarrow V_P$. Свойства множества переменных, на которые функционально отображаются множества объектов модели, заданы ниже следующими соотношениями:

$$\begin{aligned} V_{\tilde{M}} \subset \text{Var} \quad V_M \subset \text{Var} \quad V_P \subset \text{Var} \quad V_{\tilde{P}} \subset \text{Var} \\ V_{\tilde{M}} \cap (V_M \cup V_P \cup V_{\tilde{P}}) = \emptyset, V_M \cap (V_{\tilde{M}} \cup V_P \cup V_{\tilde{P}}) = \emptyset; \\ V_P \cap (V_M \cup V_{\tilde{M}} \cup V_{\tilde{P}}) = \emptyset, V_{\tilde{P}} \cap (V_M \cup V_{\tilde{M}} \cup V_P) = \emptyset. \end{aligned}$$

Будем описывать отношения инцидентности (N и L) между объектами модели, показанные на рис. 1 в префиксной форме в виде специальных предикатов

$$N(,): N \subseteq P \times P \text{ и } L(,): M \rightarrow P.$$

Для описания динамики изменения состояния в модели GraphPlus определим четыре функции:

$newP(,): D_P \times D_M \rightarrow D_P$ – новое состояние посещаемого P -объекта;

$newM(,): D_P \times D_M \rightarrow D_M$ – новое состояние посещающего M -объекта;

$activate(,): M \times D_P \times D_M \rightarrow P_+$ – функция активации локальных M -объектов;

$next(,): D_P \times D_M \rightarrow P_+$ – переход M -объекта к следующему P -объекту.

Обозначив множество всех отношений в модели через

$\mathfrak{R} = N \cup L \cup newP \cup newM \cup activate \cup next$, определим модель GraphPlus как

$$\Psi = (P, M, \mathfrak{R}, \Phi), \quad (20)$$

где Φ – формула TLA, описывающая динамику поведения вычислительного процесса. Структура формулы определяется в соответствии с графом модели, пример которого показан на рис. 1 выражениями (21)-(28).

$$\Phi \stackrel{def}{=} I \wedge \square [N]_f \wedge F \quad (21)$$

$$\begin{aligned} I \stackrel{def}{=} \wedge \exists y \in M : (\tilde{m}[y] \neq \varepsilon \wedge \tilde{m}[y] \in P) \wedge \forall x \in P : \tilde{p}[x] = \tilde{n}[x] = \varepsilon \\ \wedge \forall x \in P, y \in M : p[x] \in D_P \wedge m[y] \in D_M. \end{aligned} \quad (22)$$

$$f \stackrel{def}{=} (\tilde{p}[x]: x \in P; \tilde{m}[y]: y \in M; \tilde{n}[x]: x \in P; p[x]: x \in P; m[y]: y \in M) \quad (23)$$

$$\begin{aligned} N \stackrel{def}{=} \vee \exists x \in P, \exists y \in M : A_1(x, y) \\ \vee \exists x \in P, \exists y \in M : A_2(x, y) \end{aligned} \quad (24)$$

$$\begin{aligned} F \stackrel{def}{=} \wedge \forall x \in P, \forall y \in M : SF_f(A_1(x, y)) \\ \wedge \forall x \in P, \forall y \in M : WF_f(A_2(x, y)) \end{aligned} \quad (25)$$

$$\begin{aligned} A_1(x, y) \stackrel{def}{=} \wedge \tilde{p}[x] = \varepsilon \wedge \tilde{m}[y] = x \\ \wedge \tilde{p}'[x] = y \\ \wedge p'[x] = newP(p[x], m[y]) \\ \wedge m'[y] = newM(p[x], m[y]) \\ \wedge \tilde{n}'[x] = next(p[x], m[y]) \\ \wedge Local(x, y) \\ \wedge unchanged(V_\Phi \setminus (\{\tilde{p}[x], p[x], \tilde{n}[x], m[y]\} \cup \{\tilde{m}[k]: L(x, k)\})) \end{aligned} \quad (26)$$

$$\begin{aligned} Local(x, y) \stackrel{def}{=} \forall k \in M, L(x, k) : \\ \vee \wedge \tilde{m}[k] = \varepsilon \\ \wedge \tilde{m}'[k] = activate(k, p[x], m[y]) \\ \vee \wedge \tilde{m}[k] \neq \varepsilon \\ \wedge \tilde{m}'[k] = \tilde{m}[k] \end{aligned} \quad (27)$$

$$\begin{aligned} A_2(x, y) \stackrel{def}{=} \wedge \tilde{p}[x] = y \wedge \tilde{m}[y] = x \\ \wedge \tilde{p}'[x] = \varepsilon \\ \wedge \tilde{m}'[y] = \tilde{n}[x] \\ \wedge N(\tilde{m}[y], \tilde{m}'[y]) \\ \wedge unchanged(V_\Phi \setminus \{\tilde{p}[x], \tilde{m}[y]\}) \end{aligned} \quad (28)$$

Для структурирования конъюнктов и дизъюнктов в формулах (22), (24)-(28) использован прием, облегчающий написание длинных формул TLA [5].

Начальное состояние системы (22) предусматривает наличие хотя бы одного M -объекта в состоянии перемещения и правильно проинициализированных переменных численного метода.

Множество допустимых действий связано с графической структурой модели. Действия (26)-(27) соотносят каждый подвижный M -объект, каждому неподвижному P -объекту, определяя, при каких условиях возможно соответствующее посещение и последующая активация новых M -объектов.

Действия (26)-(27) должны быть “справедливы в строгом смысле” для предотвра-

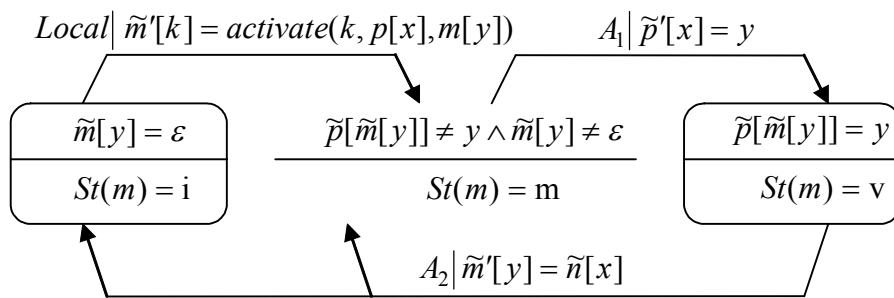


Рис. 2. Схема автомата, описывающего изменения состояния М-объекта

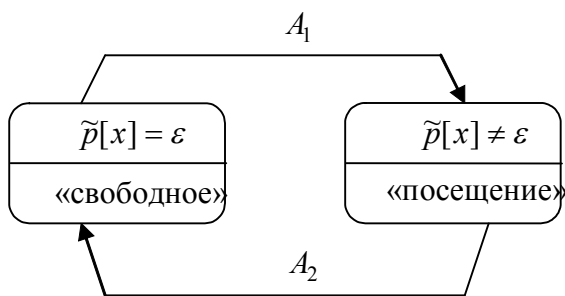


Рис. 3. Схема автомата, описывающего изменения состояния Р-объекта

щения ситуации отталкивания одного М-объекта другим при посещении некоторого Р-объекта.

Действия (28) задают переход Р-объекта в исходное состояние после посещения и управляют переходом М-объекта к следующему Р-объекту или в неактивное состояние после посещения Р-объекта.

Для действий (28) достаточна “справедливость в слабом смысле”, потому что требуется, чтобы они выполнялись, как только станут готовы к исполнению, а выполнение других действий не может изменить условия их готовности. При реализации модели на многопоточной машине условие (26) обеспечивается путем организации очереди запросов на посещение, в которой операция помещения в очередь реализована в критической секции. Условия справедливости для действий (28) гарантируются диспетчером потоков операционной системы [6].

Соответствие между определением состояний в неформальном описании модели со строгим определением (20) поясняется на рис. 2 и рис. 3.

Заключение

Таким образом, в работе построено математически точное определение прикладной вычислительной модели — спецификация модели GraphPlus. Она описывает вычислительные процессы, возникающие при параллельной и распределенной реализации алгоритмов численного моделирования. Модель позволяет изучать свойства конкретных процессов средствами TLA, в частности возможность параллельного и эффективного распределенного исполнения. Предложенная модель определяет способ реализации вычислительного процесса в распределенной и многопоточной вычислительной среде, как описано в работе [6], что обуславливает ее применимость для конструирования комплексов прикладных программ численного моделирования.

СПИСОК ЛИТЕРАТУРЫ

1. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб.: БХВ-Петербург, 2002.
2. Берзигияров П.К. Программирование на типовых алгоритмических структурах с массивным параллелизмом // Вычислительные методы и программирование. 2001. Т. 2. Разд. 2.
3. Востокин С.В. Технология визуального программирования каркасов параллельных и распределенных приложений на основе объектно-ориентированной модели вычислений GraphPlus // Труды III международной конференции “Параллельные

- вычисления и задачи управления”. М.: ИПУ РАН, 2006.
4. *Lamport L.* The temporal logic of actions. ACM Transactions on Programming Languages and Systems, 16(3):872-923, May 1994.
 5. *Lamport L.* Specifying Systems. The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley. 2002.
 6. *Востокин С.В.* Объектно-ориентированный метод структурирования кода метакомпьютерного приложения // Информационные технологии. 2006. №5.

THE SPECIFICATION OF PARALLEL AND DISTRIBUTED COMPUTATIONAL MODEL GRAPHPLUS BASED ON TLA LOGIC

© 2006 S.V. Vostokin

Samara State Aerospace University

The article describes structural model of computational processes named GraphPlus and its formal specification in Lamport's temporal logic TLA. The model targets to formalize semantics of parallel and distributed algorithms used in implementing complex numerical models.