

12. Kovtun, L. I. “Supervizornaja sistema avtomatizirovannogo upravlenija ierarhicheski-strukturirovannymi organizacionno-tehnicheskimi kompleksami morskoy tehniki s ispolzovaniem imitacionno-vychislitelnyh simuljatorov.” *Trudy CNIИ im. akad. A.N. Krylova* 54 (2010): 5–48.
13. Sharkov, N. A. “Analiz i sintez ierarhicheski-integrirovannyh cheloveko-mashinnyh kompleksov upravlenija morskoy tehniki v avarijnyh uslovijah na baze intellektualnyh tehnologij.” *Trudy CNIИ im. akad. A.N. Krylova* 54(2010): 49–76.
14. Barsegjan, A. A., M. S. Kuprijanov, V. V. Stepanenko, and I. I. Holod. *Tehnologii analiza dannyh: Data Mining, Visual Mining, Text Mining, OLAP*. SPb.: BHV-Peterburg, 2007.
15. Zaharov, I. G. *Obosnovanie vybora. Teorija i praktika*. SPb.: Sudostroenie, 2006.
16. Pospelov, D. A. *Situacionnoe upravlenie: teorija i praktika*. M.: Nauka, 1986.
17. Sharkov, N. A. “Algoritmy imitacionnogo modelirovanija i situacionnogo analiza processov protivooavarijnogo upravlenija slozhnymi cheloveko-mashinnymi sistemami v obuchajushhih sudovyh trenazherah.” *Scientific and Technical Journal of Information Technologies, Mechanics and Optics* 4(38) (2007): 219–229.
18. Russian Federation. GOST RV 1900-003-2008. Korabelnye funkcionalnye komplekсы i avtomatizirovannye sistemy upravlenija. Tipovoj sostav jergonomicheskikh harakteristik: gosudarstvennyj voennyj standart. M.: Standartinform, 2010.
19. Russian Federation. GOST R MJeK 62508 – 2014. Menedzhment riska. Analiz vlijanija na nadezhnost chelovecheskogo faktora. M.: Standartinform, 2015.
20. Proekt trebovanij VMF k sistemam informacionnoj podderzhki po borbe za zhivuchest nadvodnyh korablej, korablej specialnogo naznachenija i morskikh sudov obespechenija VMF. 2014.
21. GOST 27.310-95. Dependability in technics. Failure mode, effects and criticality analysis. Basic principles. Minsk: Mezhdgosudarstvennyj sovet po standartizacii, metrologii i sertifikacii.

ИНФОРМАЦИЯ ОБ АВТОРЕ

Кузьменко Александр Владимирович — преподаватель.
ФГК ВОУ ВПО Военно-морской институт ВУНЦ
ВМФ «Военно-морская академия имени Адмирала
флота Советского Союза Н. Г. Кузнецова»
kuzmenko_alexandr@mail.ru

INFORMATION ABOUT THE AUTHOR

Kuzmenko Alexandr Vladimirovich — lecturer.
Naval Institute of Military Training
and Research Center
of the Navy “Naval Academy”
kuzmenko_alexandr@mail.ru

Статья поступила в редакцию 20 февраля 2016 г.

УДК 004.42

**А. Н. Егоров,
В. А. Кузнецов**

РАСПАРАЛЛЕЛИВАНИЕ ВЫЧИСЛЕНИЙ ДЛЯ ОПТИМИЗАЦИИ ОПЕРАТИВНОГО УПРАВЛЕНИЯ ТРАНСПОРТНЫМ КОНВЕЙЕРОМ ШЛЮЗОВАННОЙ СИСТЕМЫ

В статье рассматриваются вопросы оптимизации оперативного управления транспортным конвейером шлюзованной системы. Обосновывается возможность реализации автоматизированной диспетчеризации судопропуска на основе взаимодействия имитационной модели и алгоритма управления в реальном масштабе времени, конечной целью которой является расписание проводки судов. До настоящего времени подобных систем, работающих в рамках шлюзованного судоходного канала в целом, не создано. Делается вывод, что для этого необходимо использовать системы, обеспечивающие автоматический обмен информацией между судами и береговыми службами, а также параллельные алгоритмы обработки данных. В качестве среды реализации предлагается универсальная программная оболочка распараллеливания, модель и особенности функционирования которой представлены в статье.

Отмечается, что её основным назначением является организация эффективного выполнения прикладного решения пользователя.

Ключевые слова: оптимизация оперативного управления, шлюзованные судоходные каналы, параллельные алгоритмы, технология CUDA.

Введение

На внутренних водных путях России, являющихся частью единой транспортной системы, насчитывается более 150 шлюзов, большинство из которых интегрированы в шлюзованные системы (шлюзованные судоходные каналы — ШСК), крупнейшими представителями которых являются Волго-Балтийский водный путь и Волго-Донской судоходный канал. Исходя из особенностей функционирования ШСК, в работе [1] делается вывод о том, что ШСК представляет собой конвейерную систему с наличием противотока (КСП), т. е. класса транспортных систем типа судоходного канала, отличительными особенностями которых наряду со стохастичностью являются динамичность, дискретность, многомерность, а также влияние неуправляемых и субъективных факторов.

В настоящее время ШСК являются существенными ограничителями дальнейшей интенсификации транспортного процесса. Мероприятия по автоматизации управления судопропуском способны увеличить пропускную способность каналов до 15 – 20 % и поэтому являются предпочтительной альтернативой по сравнению со строительством параллельных «ниток» сооружений, требующим значительно больших инвестиций, трудовых ресурсов и времени. Однако проблема автоматизации и, как следствие, оптимизация оперативного управления транспортными конвейерами, в том числе КСП типа ШСК, решены не полностью. Так, в практике оперативного диспетчерского управления судопропуском преобладают эвристические методы и только в последнее время начали проводиться работы с использованием компьютерных технологий для решения указанных проблем. Например, в информационном сообщении о системе управления судоходными шлюзами компании «Траскон Текнолоджи» [2] приведены краткие сведения о функционале автоматизированного рабочего места (АРМ) диспетчера судоходного канала:

- автоматизированная подготовка распоряжений и указаний по судопропуску для вахтенных дежурных шлюзов и дистанционного управления водным режимом канала;
- специализированное программное обеспечение (ПО) для оптимизации судопропуска через канал с учетом экономии энергоресурсов;
- фиксация всех действий по судопропуску и работе оборудования в базе данных;
- автоматизированная подготовка отчетной и статистической информации.

Заметим, что другой информации (например, математических моделей решения задачи оптимизации судопропуска, а также методов, используемых при разработке специализированного ПО) эти сведения не содержат.

В настоящее время современные информационные технологии интенсивно привлекаются для решения вспомогательных задач, напрямую не связанных с оптимизацией оперативного управления транспортным конвейером типа ШСК: автоматизации управления технологическим процессом шлюзования (системы АСУ ТП) [3], [4], получения навигационной и другой информации о судах (автоматические идентификационные системы АИС) [5].

В работах [5], [6] проблемы автоматизации и на их основе построение оптимального расписания шлюзования судов рассматриваются только в рамках одного шлюза. На самом деле, такое расписание всегда должно являться следствием автоматизированной диспетчеризации судопропуска в ШСК.

Теоретические и методические основы оптимизации оперативного управления транспортным конвейером типа ШСК, представленные в работе [1], дают возможность реализации автоматизированной диспетчеризации судопропуска на основе взаимодействия имитационной модели и алгоритма управления в реальном масштабе времени. Это утверждение основано на том, что имитационная модель КСП, основанная на формализованном описании всех подсистем ШСК

(«Шлюз», «Трасса судопропуска», «Судно», «Внешние условия»), непосредственно связана с алгоритмом управления. С одной стороны, имитация является формальной реализацией функционирования всей системы, отражающей динамику состояния КСП с учетом влияния стохастических факторов, и, таким образом, является исходным моментом для синтеза алгоритмов управления. С другой стороны, имитационная модель КСП отражает принципы, заложенные в стратегию управления, и является совместимой с ними.

Конечной целью автоматизированной диспетчеризации судопропуска на основе математической модели системной оптимизации (например, по критерию минимума совокупных потерь провозной способности при прохождении судами ШСК в целом с учетом естественных и технологических ограничений) является расписание (график проводки судов), формализация которого представлена в работе [1]. Для его построения необходима имитационно-управляющая система (ИУС) ШСК, которая должна содержать в своем составе имитационную модель функционирования ШСК, осуществляющую генерацию допустимых вариантов управления и имитационное моделирование поведения системы при принятии этих вариантов. До настоящего времени подобных систем, работающих в реальном масштабе времени, не создано. Это объясняется следующими обстоятельствами. Во-первых, только в течение последнего времени появились навигационные системы, предназначенные для автоматического обмена информацией между судами и береговыми службами [5], и, во-вторых, вычислительная мощность цифровой техники вносила существенные ограничения до тех пор, пока не появилась возможность использования параллельных алгоритмов обработки информации.

Модель функционирования системы

Рассмотрим структуру и модель функционирования автоматизированной системы оптимизации оперативного управления транспортным конвейером типа ШСК, основными компонентами которой являются:

- шлюзы, каждый из которых оборудован АСУ ТП и системами мониторинга его технического состояния;
- кластер серверов;
- аппаратно-программный комплекс для оптимизации управления судопропуском ШСК;
- АРМ диспетчера;
- технические средства, обеспечивающие мониторинг погодных условий, обмен навигационной и другой информацией между судами и береговыми службами (сведения о рейсе, технические характеристики, тип судна, указания о графике его прохождения ШСК и др.).

Один из возможных вариантов организации системы оптимизации управления судопропуском ШСК и схема взаимодействия ее компонентов приведены на рис. 1. Каждый из компонентов включён в единую корпоративную сеть. На схеме такие связи изображены сплошными линиями, пунктирными линиями показано взаимодействие судов и береговых станций мониторинга. Такая конфигурация позволяет в режиме реального времени осуществлять сбор и передачу информации о судах, показателях технического состояния шлюзов, уровне воды, погодных условиях и др., что в интерактивном виде отображается на АРМ диспетчера. Это, в конечном итоге, обеспечивает постоянный мониторинг ситуации на канале. В случае необходимости или отсутствия каких-либо данных диспетчер с помощью АРМ может вручную добавлять или корректировать их.

На основе актуальной на текущий момент информации выполняется расчёт оптимального расписания проводки судов по ШСК, из которого передаются сведения о дальнейших действиях: судам — график прохождения по ШСК, шлюзам — расписание шлюзования.

Процесс мониторинга также позволяет отслеживать ход выполнения прогнозного расписания, и в случаях отклонения от него или других чрезвычайных ситуаций (например, выход из строя технических средств судна или шлюза, изменение погодных условий, влияющих на судопропуск), сообщение об этом незамедлительно отображается на АРМ диспетчера для принятия оперативного решения.

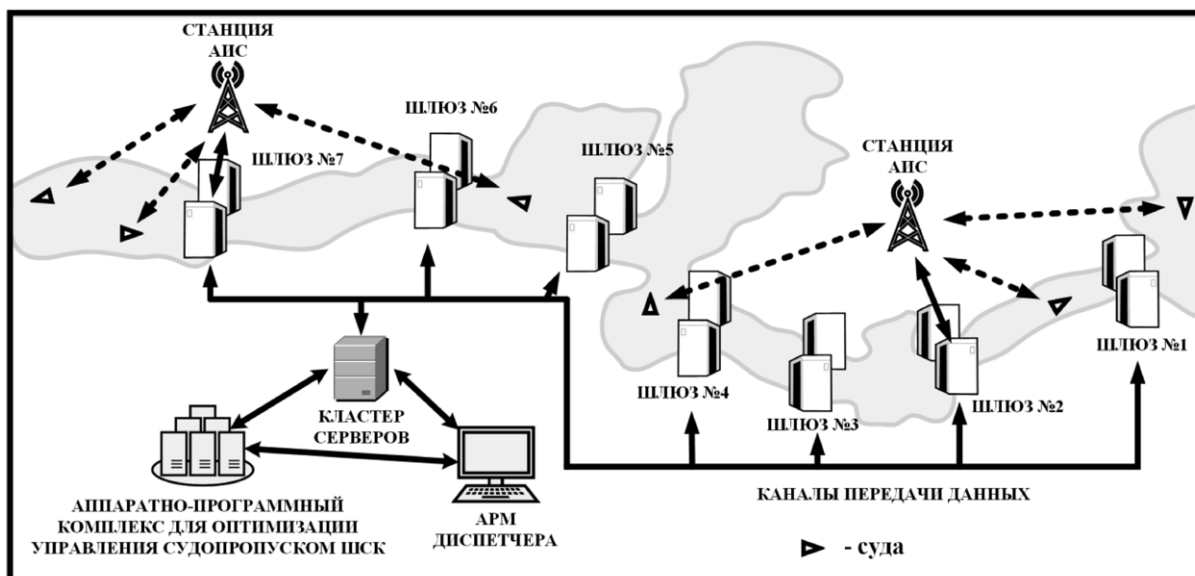


Рис. 1. Пример организации системы оптимизации управления судопропуском ШСК

Распараллеливание вычислений

На данный момент работа системы в режиме реального времени возможна лишь за счёт применения параллельных технологий при построении прогнозного расписания проводки судов через ШСК. Для такой реализации предлагается воспользоваться универсальной программной оболочкой распараллеливания (ПОР) [7], основным назначением которой является организация эффективного исполнения прикладного решения пользователя. Для этого ПОР использует весь потенциал подключенных к компьютеру вычислительных устройств, допускающих распараллеливание, а реализация в дальнейшем возможности объединения компьютеров в единую вычислительную сеть позволит создавать мощные аппаратно-программные комплексы.

Корпоративная сеть в рассматриваемой автоматизированной системе оптимизации оперативного управления транспортным конвейером типа ШСК является незаменимым компонентом. Учитывая тот факт, что ШСК — стратегический объект, информация в сети должна передаваться по защищённому каналу в зашифрованном виде. Однако в процессе функционирования могут возникнуть ситуации, связанные с потерей или искажением секретных ключей, что приведёт к потере данных и обрыве связи [8], а это в условии работы системы в режиме реального времени является катастрофическим. Для восстановления таких ключей с целью обеспечения доступа к зашифрованным данным и возобновлению соединения также можно использовать ПОР [8]. Таким образом, применение ПОР позволяет решить как задачу оптимизации оперативного управления, так и проблему функционирования корпоративных сетей на ШСК.

Модель ПОР

ПОР представляет собой совокупность взаимодействующих модулей, связующим звеном которых является ядро программной оболочки (ЯПО). ЯПО включает базовую функциональность ПОР и имеет самую высокую независимость от других модулей. Вся основная логика работы ПОР заложена во взаимодействии ЯПО с модулями устройств. Модули устройств (МУ) выполняют такую основную работу вычислений ПОР, как решение пользовательских задач (ПЗ). Предполагается также, что МУ имеют одно из двух состояний: busy (устройство занято) и free (устройство свободно).

При формировании требований к параллельным алгоритмам наиболее важным является определение последовательности выполнения определённых событий для каждого вычисления программы. Это позволяет для описания модели ПОР использовать язык темпоральной логики линейного времени (LTL) [9], [10], в котором учтён феномен времени. Каждое событие будет ха-

рактизоваться булевой переменной, которая принимает значение «Истина» тогда и только тогда, когда наступает событие.

В LTL помимо булевых логических связей для описания причинно-следственной зависимости событий во времени используются темпоральные операторы:

- $X\phi$ «в следующий момент времени» — указывает на то, что ϕ выполняется на следующем состоянии пути;
- $F\phi$ «когда-то в будущем» — указывает на то, что ϕ будет соблюдаться в каком-то последующем состоянии пути;
- $G\phi$ «всегда» — указывает на то, что ϕ выполняется в каждом состоянии пути;
- $\phi U \psi$ «до тех пор, пока» — указывает на то, что ϕ выполняется до тех пор, пока не станет верно ψ ;
- $\phi R \psi$ «высвободить, открепить» — указывает на то, что ψ может перестать быть верным только после того, как станет верно ϕ .

Для описания модели введем следующие атомарные высказывания, соответствующие основным событиям вычислений программы:

- try_i — i -й пользователь собирается отправить задачу;
- add_i — i -я ПЗ добавляется в очередь задач;
- del_i — i -я ПЗ удаляется из очереди задач;
- $send_i$ — i -я ПЗ отправляется на решение МУ;
- $create_i$ — i -я подзадача формируется из ПЗ;
- $solve_i$ — i -й МУ решил подзадачу;
- $resv_i$ — i -й результат подзадачи МУ отправляется ЯПО;
- res_i — формируется i -й результат ПЗ;
- $free$ — МУ свободен;
- $busy$ — МУ занят;
- end — получен конечный результат ПЗ;
- $empty$ — очередь задач пуста.

Рассмотрим математическую модель программной оболочки, построенную на основе языка LTL:

Всякий раз, когда пользователь собирается запустить ПЗ, ЯПО добавляет себе ПЗ в очередь задач:

$$G (try_i \rightarrow add_i). \quad (1)$$

Пока очередь задач не окажется пустой ($empty$), ЯПО будет отправлять задачи на решение свободным МУ ($free$):

$$G ((free \rightarrow send_j) U empty). \quad (2)$$

Всякий раз, когда ЯПО отправляет задачу МУ, формируется подзадача для этого модуля:

$$G (send_i \wedge X create_i). \quad (3)$$

Всякий раз, когда у МУ формируется подзадача, он становится занятым ($busy$):

$$G (create_j \rightarrow busy). \quad (4)$$

Всякий раз, когда МУ становится занятым, должна когда-нибудь решиться подзадача:

$$G (busy \rightarrow F solve_i). \quad (5)$$

После решения подзадачи МУ возвращает результат ЯПО, и становится свободным ($free$):

$$G (solve_i \rightarrow X (resv_i \wedge free)). \quad (6)$$

Всякий раз, когда ЯПО получает результат решения подзадачи от МУ, ЯПО формирует результат соответствующей ПЗ:

$$G(\text{res}_i \rightarrow X \text{res}_j). \quad (7)$$

Всякий раз, когда ЯПО формирует результат решения ПЗ, оболочка проверяет, получен или нет конечный результат. Если этот результат получен, то ПЗ удаляется и очереди:

$$G((\text{res}_i \wedge \text{end}) \wedge \text{del}_j). \quad (8)$$

Полученные зависимости темпоральной логики позволяют представить псевдокод основных процессов модели ПОР (листинги 1 и 2). Для каждой операции псевдокода ставится счетчик. Следует заметить, что псевдокод МУ является инвариантным по отношению ко всем вычислительным устройствам:

```

while (true)
{
1.1.   if (try)
1.2.
1.3.   add();
1.4.   while (!empty && status == free)
1.5.
       send();
1.6.   if (recv())
1.7.   {
1.8.
       res();

       if (end)

       del();
       }
}
    
```

Листинг 1. Псевдокод ядра программной оболочки модели ПОР

```

while (true) {
2.1.   if (create()) {
2.2.
2.3.     status = busy;
2.4.
2.5.     while (!solve) {}

       recv();

       status = free;
       }
}
    
```

Листинг 2. Псевдокод модуля устройств модели ПОР

В соответствии с построенным псевдокодом строится размеченная система переходов LTS (Labelled Transition System), где учитываются всевозможные состояния ПОР, основными из которых являются:

- счетчики выполнения команд;
- состояние решения задачи или подзадачи (решено / не решено);
- состояние модуля устройства (занят / свободен).

На рис. 2 приведены схемы LTS каждого отдельного потока: единственного для ЯПО и множества потоков МУ. В узлах через запятую показаны значения основных состояний ПОР. Поскольку программная реализация обеспечивает работу со многими процессами МУ, рассмотрим LTS-схему, демонстрирующую динамику взаимодействия двух процессов МУ и одного процесса ЯПО (рис. 3). ЯПО управляет работой процессов МУ, выполняемых параллельно. Взаимодействие МУ и ЯПО, осуществляемое во временные отметки отправки (блок состояния ЯПО — «1.4, false») и получения (блок состояния ЯПО — «1.5, false») результата решения подзадачи МУ, должно быть синхронизировано. На схеме синхронизация процессов показана жирными линиями.

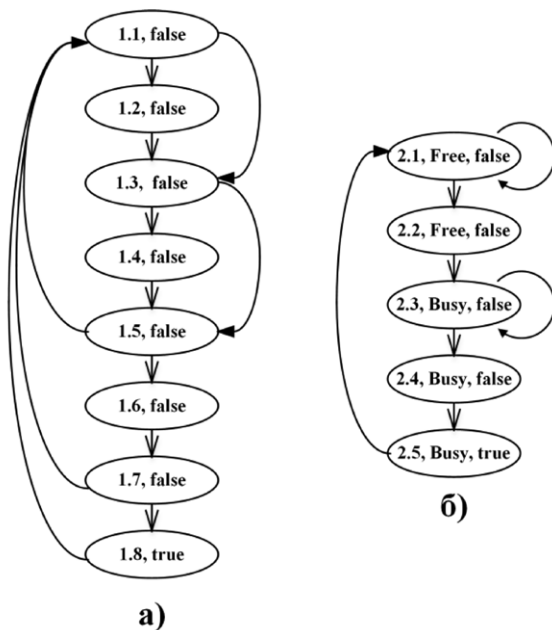


Рис 2. LTS потоков ПОР:
а — ядро программной оболочки; б — модуль устройств

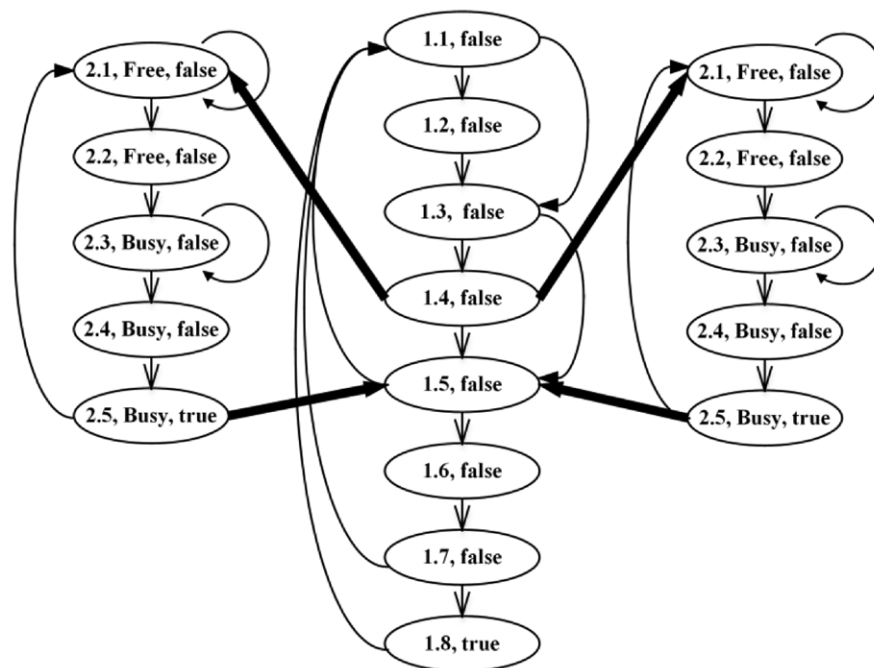


Рис. 3. LTS взаимодействия процесса ядра программной оболочки с двумя процессами модуля устройств

Программная реализация

ПОР реализует универсальный подход к распараллеливанию прикладных решений пользователя, используя весь потенциал подключенных к компьютеру вычислительных устройств, допускающих распараллеливание. Каждому типу устройства соответствует свой обработчик, что даёт возможность для повышения производительности воспользоваться особенностями их архитектуры. Прикладные решения реализуются в виде dll-библиотек, разработанных для каждого типа устройства, работающего с ПОР. Каждая библиотека должна содержать значения параметров, требуемых для настройки среды программной оболочки, таких как размер массива бинарных данных задачи и результата её решения, идентификатор задачи и др. Кроме того, необходимо реализовать ряд экспортных функций, обеспечивающих выполнение следующих операций:

1. Score — оценка производительности устройства.
2. Parser — проверка, обработка и формирование данных ПЗ.
3. Cut — формирование данных подзадачи.
4. SimSim — решение подзадачи.
5. Result и ResultBuild — формирование общего решения ПЗ.

ПОР состоит из модулей, работу которых организует ЯПО. Схема взаимодействия модулей программной реализации приведена на рис. 4, где экспортные функции dll-библиотек представлены в виде отдельных модулей и выделены серым цветом. Сплошными линиями показаны связи между модулями, образующими ЯПО, пунктирными обозначена динамика процесса решения ПЗ. Модули, взаимодействия которых обозначены жирными линиями, выполняются параллельно.

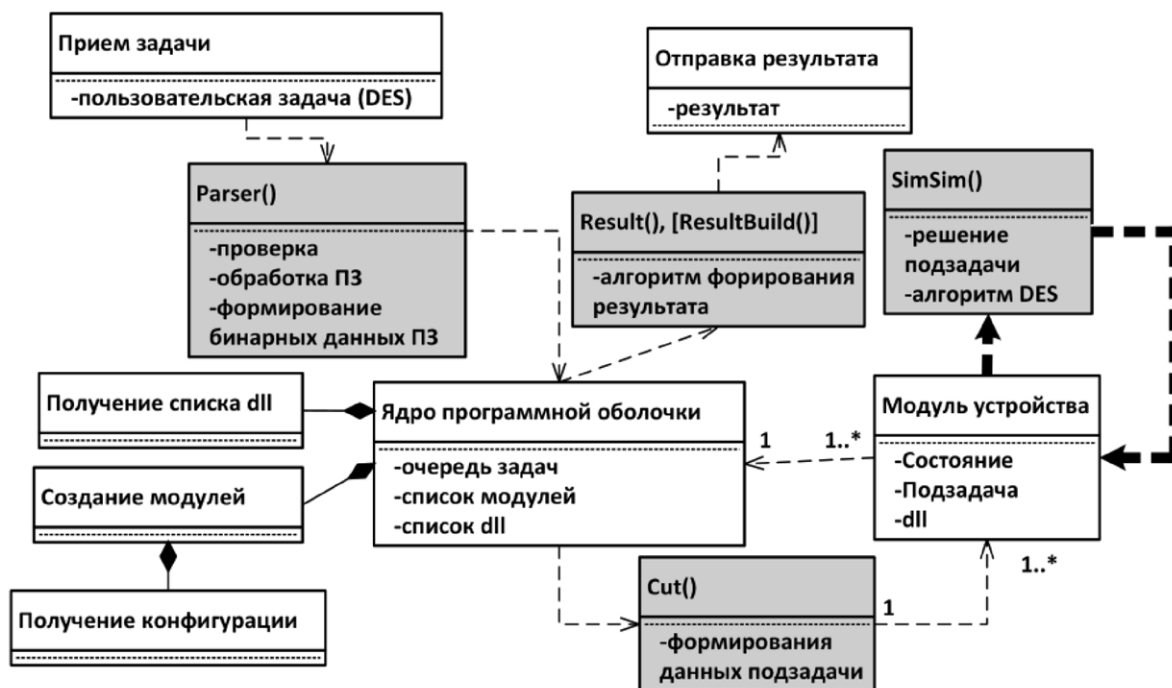


Рис. 4. Схема взаимодействия модулей ПОР

Опишем работу ПОР в виде пошагового алгоритма.

Этап 1. Создание модулей устройств (МУ), отвечающих за взаимодействие с конкретным аппаратным средством, на основе информации, сформированной в результате предварительного анализа вычислительных устройств компьютера, с которыми поддерживает работу ПОР.

Этап 2. Инициализация задачи посредством считывания и обработки данных конфигурационного файла, созданного пользователем, который содержит информацию о типе задачи и бинарные данные задачи, соответствующие выбранному типу.

Этап 3. Решение полученной задачи:

– шаг 1 — с помощью функции «Score», вызванной из соответствующей dll-библиотеки, оценивается скорость решения текущей ПЗ на определенном устройстве;

– шаг 2 — ЯПО отправляет задачу на решение множеству параллельно работающих МУ, при этом, в зависимости от оценки производительности устройства, модуль «Cute» формирует данные подзадачи для каждого МУ;

– шаг 3 — в процессе решения подзадачи модуль «SimSim» осуществляет дополнительное распараллеливание вычислительного процесса, используя особенности аппаратного средства;

– шаг 4 — после решения подзадачи МУ отправляет результат ЯПО.

Этап 4. Формирование общего результата решения задачи, на основе информации полученной от МУ, и отправка его пользователю.

При создании комплекта dll-библиотек для решения определённой задачи реализация функций «Score» и «SimSim» является уникальной по отношению к различным аппаратным средствам. В свою очередь, функции Parser, Cute, Result и ResultBuild, зачастую, являются инвариантными, однако в ряде случаев реализации отличаются. Например, в процессе распараллеливания вычислительного процесса функцией «SimSim» на конкретном устройстве могут потребоваться дополнительные параметры, в связи с чем формирование данных подзадачи функцией «Cute» будет другим.

Архитектура ПОР позволяет практически не ограниченно расширять список поддерживаемых устройств и задач. На текущий момент программная оболочка может использовать центральные процессоры и графические видеокарты, которые поддерживают технологию CUDA компании NVIDIA. CUDA — это платформа параллельных вычислений и модель программирования, позволяющая существенно увеличить вычислительную производительность за счёт максимально эффективного использования ресурсов видеокарты (графических процессоров и памяти) компании NVIDIA.

Выводы

1. Реализация автоматизированной системы оптимизации оперативного управления транспортным конвейером типа ШСК, работающей в режиме реального времени, стала возможной благодаря, во-первых, появлению навигационных систем, предназначенных для автоматического обмена информацией между судами и береговыми службами, и, во-вторых, применению параллельных алгоритмов для повышения эффективности вычислений в процессе составления оптимального расписания проводки судов через шлюзованную систему.

2. В качестве среды реализации рассмотрена ПОР, основной целью которой является организация эффективного исполнения прикладного решения пользователя. Программная оболочка реализует универсальный подход к распараллеливанию вычислений и обеспечивает взаимодействие с пользователем, подключенными к компьютеру вычислительными устройствами, допускающих распараллеливание, и распределение нагрузки между этими устройствами, а реализация в дальнейшем возможности объединения компьютеров в единую вычислительную сеть позволит создавать мощные аппаратно-программные комплексы.

СПИСОК ЛИТЕРАТУРЫ

1. Бутов А. С. Транспортные системы: моделирование и управление / А. С. Бутов, Д. В. Гаскаров, А. Н. Егоров [и др.] / под общ. ред. проф. А. С. Бутова. — СПб.: Судостроение, 2001. — 552 с.
2. Траскон Текнолоджи [Электронный ресурс]. — Режим доступа: <http://tta.ru/projects/-transport/iustbs.html> (дата обращения: 26.03.2016).
3. Морозов В. Н. Комплексная система автоматизации судопропуска Шекснинского гидроузла / В. Н. Морозов, И. Л. Краснощеков, Л. А. Павлович [и др.] // Известия Самарского научного центра Российской академии наук. — 2015. — Т. 17. — № 2–4. — С. 843–850.

4. Бутин В. П. Автоматизация швартовки судов при их шлюзовании / В. П. Бутин // Журнал университета водных коммуникаций. — 2010. — № 3. — С. 31а–42.
5. Белоусов И. В. Применение информационных технологий для управления движением судов через шлюз / И. В. Белоусов, В. Ф. Самосейко, А. В. Саушев // Моделирование, оптимизация и информационные технологии. — 2015. — № 2 (9). — С. 13.
6. Стальмаков В. А. Параллельный генетический алгоритм для решения задачи составления расписания прохождения судов через шлюзованные системы и его верификация / В. А. Стальмаков // Вестник Государственного университета морского и речного флота имени адмирала С. О. Макарова. — 2014. — № 1 (23). — С. 93–102.
7. Св. ГР № 2014619266 РФ. Программная оболочка распараллеливания процесса вычисления прикладных решений / А. Н. Егоров, В. А. Кузнецов, И. А. Назаргулов; правообладатель: Санкт-Петербург, Государственный университет морского и речного флота имени адмирала С. О. Макарова; заявл. 18.07.2014; опубл. 11.09.2014.
8. Егоров А. Н. Модель распараллеливания вычислений для повышения эффективности восстановления доступа к данным в корпоративных сетях / А. Н. Егоров, В. Е. Марлей, В. А. Кузнецов [и др.] // Труды СПИИРАН. — 2015. — № 5 (42). — С. 37–55.
9. Baier C. Principles of Model Checking / C. Baier, J.-P. Katoen. — Cambridge: The MIT Press, 2008. — 994 p.
10. Kröger F. Temporal Logic and State Systems / F. Kröger, S. Merz. — Berlin: Springer, 2008. — 436 p.

PARALLEL COMPUTING FOR OPTIMIZATION OF OPERATIONAL MANAGEMENT OF THE TRANSPORT CONVEYOR OF LOCK SYSTEM

In article the optimization of operational management of the transport conveyor of lock system are considered. Substantiates the possibility of implementing automated dispatching passing of lock system on based interaction simulation model and control algorithm in real time, the purpose of which is the passage of ships schedule. To date such systems operating for the whole lock canal, not created. It is concluded that this requires the use of the system that allows the automatic exchange of information between ships and shore services, as well as parallel data processing algorithms. As the environment of implementing is proposed universal program shell of parallelizing, model and implementation feature of which is presented in the article. It is noted that its main purpose is organization of efficient execution of user application solution.

Keywords: optimization of operational management, lock canal, parallel algorithms, CUDA technology.

REFERENCES

1. Butov, A. S., D. V. Gaskarov, A. N. Yegorov, and N. V. Krupenina. *Transportnye sistemy: modelirovanie i upravlenie*. SPb.: Sudostroenie, 2001.
2. Traskon Teknologzhi. Web. 26 March 2016 <<http://tta.ru/projects/transport/iustbs.html>>.
3. Morozov, V. N., I. L. Krasnoshhekov, L. A. Pavlovich, I. L. Pavlovich, D. B. Krasinskij, I. A. Danilushkin, E. V. Mel'nikov, and S. A. Kolpashhikov. "Crafts passing automation complex system at Sheksninskiy water-engineering system." *Proceedings of the Samara Scientific Center of the Russian Academy of Sciences* 17.2–4 (2015): 843–850.
4. Boutin, V. P. "Automation of the mooring of ships during their locking through." *Zhurnal Universiteta vodnyh kommunikacij*. 3 (2010): 31a–42.
5. Belousov, I. V., V. F. Samosejko, and A. V. Saushev. "Application of information technology to control the movement of ships through a lock." *Modeling, Optimization and Information Technology* 2(9) (2015): 13.
6. Stalmakov, V. A. "Parallel genetic algorithm for solving scheduling passing through the gateways system and its verification." *Vestnik Gosudarstvennogo universiteta morskogo i rechnogo flota imeni admirala S.O. Makarova* 1(23) (2014): 93–102.
7. Yegorov, A. N., V. A. Kuznetsov, and I. A. Nazargulov. Program shell of parallelizing the computation of application solutions. Russian Federation, assignee. Certificate of State registration 2014619266. 11 Sept. 2014.

8. Yegorov, A. N., V. Y. Marley, V. A. Kuznetsov, and I. A. Nazargulov. “The model of parallel computing to effectiveness improvement of the restoration of access to data in corporate networks.” *SPIIRAS Proceedings* 5(42) (2015): 37–55.

9. Baier, C., and J.-P. Katoen. *Principles of Model Checking*. Cambridge: The MIT Press, 2008.

10. Kröger, F., and S. Merz. *Temporal Logic and State Systems*. Berlin: Springer, 2008.

ИНФОРМАЦИЯ ОБ АВТОРАХ

Егоров Александр Николаевич —

кандидат технических наук, доцент.

ФГБОУ ВО «ГУМРФ имени

адмирала С.О. Макарова»

eanspb@rambler.ru, kaf_vsi@gumrf.ru

Кузнецов Виталий Александрович — аспирант.

Научный руководитель:

Егоров Александр Николаевич.

ФГБОУ ВО «ГУМРФ имени

адмирала С. О. Макарова»

kuznecov.v.spb@mail.ru

INFORMATION ABOUT THE AUTHOR

Yegorov Alexander Nikolayevich —

Ph.D., associate professor.

Admiral Makarov State University

of Maritime and Inland Shipping

eanspb@rambler.ru, kaf_vsi@gumrf.ru

Kuznetsov Vitaliy Aleksandrovich — postgraduate.

Supervisor:

Yegorov Alexander Nikolayevich

Admiral Makarov State University

of Maritime and Inland Shipping

kuznecov.v.spb@mail.ru

Статья поступила в редакцию 25 февраля 2016 г.