

относительно информационных вектор-символов согласно правилу максимума апостериорной вероятности:

$$\hat{\mathbf{d}}^{(l)} = \arg \max_j \left\{ \sum_{s^{(l)}=0}^{S-1} \alpha_{s^{(l)}}^{(l)} \cdot \gamma_{s^{(l)},j}^{(l)} \cdot \beta_{s^{(l)},j}^{(l+1)} \right\}, \quad (23)$$

где  $j$  – номер варианта систематического вектор-символа на шаге  $l$ .

На рис. 1 кривая 3 характеризует помехоустойчивость принятия решения в соответствии с правилом при наличии сильно выраженной МСИ. Из сопоставления с кривой 1, которая соответствует процедуре декодирования с ОСР по кодовой решетке в одном из направлений (прямом или обратном) на первом шаге итерации, следует, что декодирование по правилу (23) позволяет несколько улучшить помехоустойчивость

по сравнению со случаем декодирования в одном направлении.

### Литература

1. Robertson P., Worz T. Bandwidth-Efficient Turbo-Trellis-Coded Modulation Using Punctured Component Codes // IEEE Journal on Selected Areas in Communications, Vol. 16, NO 2, February 1998. – P. 206-218.
2. Левин С.В., Петров О.А., Хабаров Е.О., Шалковский С.Ю. Декодирование сверточных кодов на выходе канала с МСИ с использованием выравнивателя с обратной связью по решению и мягкого декодера Витерби // ИКТ. №4, 2008. – С. 28-32.
3. Ван Трис Г.Л. Теория обнаружения, оценок и модуляции. Т.1. М.: Сов. радио, 1972. – 744 с.

## MAP-DECODER OF TTCM-SIGNALS WITH THE BIDIRECTIONAL EQUALIZATION

Khabarov E.O., Fomchenco J.Ad.

**The maximum a posteriori decoder (MAP- decoder) for signal-sequence with turbo-trellis-coded modulation (TTCM) in channels with inter-symbol interference (ICI) are presented. The peculiarity of the decoder is the use of estimation signal-amplitudes, trellis-decision-feed-back with rib-metrics-computing, as well as bidirectional equalization.**

**Keywords:** turbo-trellis-coded modulation (TTCM), intersymbol interference (ICI), bidirectional equalization, maximum a posteriori decoding (MAP-decoding).

Фомченко Яна Эдуардовна, студентка Поволжского государственного университета телекоммуникаций и информатики (ПГУТИ). Тел. (8-846) 228-00-72. E-mail: tors@psati.ru

Хабаров Евгений Оттович, к.т.н. доцент Кафедры ТОРС ПГУТИ. Тел. (8-846) 271-96-23. E-mail: tors@psati.ru

## ТЕХНОЛОГИИ КОМПЬЮТЕРНЫХ СИСТЕМ И СЕТЕЙ

УДК 004.891.3

### АЛГОРИТМЫ АНАЛИЗА ДЕЙСТВИЙ ОПЕРАТОРОВ РАДИОТЕХНИЧЕСКОГО КОМПЛЕКСА В НАЗЕМНОЙ СИСТЕМЕ ОБРАБОТКИ ИНФОРМАЦИИ

Волчихин В.И., Пащенко Д.В.

Приведено сравнение последовательного и параллельных алгоритмов обработки больших массивов информации для повышения оперативности выполнения анализа действий операторов радиотехнического комплекса в наземной системе обработки информации, рассмотрено повышение быстродействия при чтении данных с диска с использованием конвейера.

**Ключевые слова:** авиационный комплекс радиолокационного дозора и наведения, радиотехнический комплекс, объективный контроль, массив данных, менеджер параллельной обработки, многоядерный процессор, контроль действий операторов.

### Введение

Важную роль в обеспечении обороноспособности страны играют авиационные комплексы радиолокационного дозора и наведения (АК РЛДН). Данная система представляет из себя самолет с установленным на борту радиолокационным комплексом. АК РЛДН – сложнейшая радиотехническая, электронная и информационно-вычислительная система, надежная и точная работа которой обеспечивается в том числе и наземной системой обработки информации.

Поэтому в состав РЛК входят средства объективного контроля (СОК), обеспечивающие анализ состояния технических средств АК РЛДН, радиолокационной обстановки и действий операторов. Условия, предъявляемые СОК АК РЛДН, требуют особых подходов при проектировании архитектуры такой системы и алгоритмов обработки данных. В условиях интенсивной летной эксплуатации (обеспечение непрерывного дежурства в воздухе) актуальным является сокращение времени, необходимого для послеполетной обработки зарегистрированной информации. При этом наибольшее время занимает процесс контроля действий операторов.

Современные вычислительные системы, как правило, включают в себя процессоры, имеющие несколько вычислительных ядер. Для использования всей мощности предоставляемых вычислительных ресурсов необходима разработка алгоритмов, осуществляющих многопоточную (параллельную) обработку информации.

Целью данной работы является исследование и разработка наиболее оптимальных алгоритмов, с точки зрения длительности обработки, процедур контроля действий операторов.

### **Объективный контроль действий оператора боевого расчета**

Каждый оператор располагается на своем рабочем месте и выполняет определенные боевые задачи. Суть контроля состоит в том, чтобы проверить правильность последовательности действий по выполнению конкретной задачи. Для этого необходимо из общего массива данных выбрать действия конкретного оператора, определить, какую именно задачу он выполнял, и оценить правильность ее выполнения. Кроме того, иногда требуется проанализировать информацию, зарегистрированную аппаратурой, например состояние каналов передачи [1].

В качестве примера рассмотрим работу оператора боевого расчета АК РЛДН. Оператор боевого расчета наводит истребитель на определенную цель, при этом он постоянно корректирует действия истребителя с учетом изменения местоположения цели либо переводит корректировку в автоматический режим, когда команды наведения на цель выдает аппаратура комплекса [2]. При обнаружении цели истребителю назначаются исходные данные для наведения на цель. После сообщается координата вертикали, на которую следует выйти для поражения цели. Завершение процесса наведения должно подтвердиться приходом команды «Конец атаки».

Каждая запись действия оператора содержит номер рабочего места, номер трассы по бортовой системе управления наведением (БСУН), номер трассы по радиолокационному контролю, идентификатор выполняемого действия и набор служебных параметров (координаты по каждой цели, курс, скорость и т.д.).

Как правило, на борту АК РЛДН выполняют боевую задачу несколько штурманов наведения. Можно сказать, что существует такой момент времени, когда все они будут заняты решением боевых задач.

Следовательно, может существовать определенный момент времени, при котором максимальное количество выполняемых боевых задач зависит от текущего количества штурманов и текущей нагрузки каждого штурмана

Следующим этапом формализации является построение алгоритма контроля действий лиц боевого расчета АК РЛДН. Эффективность данного алгоритма можно сформулировать в задаче разработки такого алгоритма контроля, который позволит за определенное время обрабатывать весь объем информации, полученной при максимальной нагрузке на систему, за весь период эксплуатации АК РЛДН

Задачу анализа действий оператора можно разбить на две подзадачи:

- фильтрация из общего массива данных записей, относящихся к одной и той же задаче;
- контроль порядка действий, выполняемых оператором.

### **Модель анализа действий операторов**

Для анализа правильности действий операторов была разработана модель, основанная на классических продукционных правилах. Для нахождения ошибок оператора будем использовать сценарии, управляемые образцами, которые работают с набором записей по конкретной задаче. На каждом шаге работы такой сценарий анализирует текущую ситуацию и определяет по анализу образцов, какой модуль сценария подходит для обработки этой ситуации. Каждый управляемый образцом модуль состоит из механизмов исследования и модификации одной или нескольких структур данных. Модуль на очередном шаге работы анализирует данные рабочей памяти, проверяя наличие структур, которые сопоставляются с его образцом. Системы, построенные на основе управляемых образцами модулей, называют системами вывода, управляемыми образцами. Функции управления в этих системах осуществляет интерпретатор.

Пусть  $R$  – множество упорядоченных по времени записей, которое можно представить как множество, состоящее из элементов  $r$ :

$$r = \{t, x, p_1, p_2, p_3, \dots, p_n\}, \quad (1)$$

где  $t$  – время записи;  $x$  – команда оператора;  $p_1, p_2, p_3, \dots, p_n$  – дополнительные поля записи, используемые при ее фильтрации и анализе, где  $n$  – количество информационных полей в записи  $r$ .

Для дальнейшего анализа записи введем понятие сценария проверки. Как следует из опыта применения ДРЛО предыдущих поколений, любая боевая задача, выполняемая ими, строго формализована и состоит из конечного набора команд, используемых в различных режимах работы. Для анализа входной последовательности необходимо провести фильтрацию элементов множества  $r$  (1). Для этого вводится понятие фильтра  $F$ , состоящего из элементов множества  $f$ , имеющего поля эквивалентные одноименным полям множества  $r$ :

$$f = \{p_1, p_2, p_3, \dots, p_n\}. \quad (2)$$

Статистическая оценка порядка выполнения команд будет осуществляться с использованием сценариев проверки. Под сценариями проверки будем понимать упорядоченное множество сценариев  $S$ , состоящее из  $S_i$

$$S_i = \left\{ \begin{array}{l} x_{start}, \\ P = \{kort_1, kort_2, \dots, kort_n\}, \\ E = \{kort_{stop1}, \dots, kort_{stopl}\}, \\ f = \{p_1, \dots, p_m\}, \end{array} \right\} \quad (3)$$

где  $P$  – множество, состоящее из  $n$  кортежей промежуточных условий сценария  $S_i$ ;  $x_{start}$  – стартовая команда сценария  $S_i$ ;  $E$  – множество, состоящее из  $l$  кортежей  $kort_{stop}$  условий завершения сценария  $S_i$ ;  $f$  – элемент множества  $F$ , где  $F$  – фильтр входной последовательности, состоящий из  $m$  фильтруемых полей (2), а  $kort$  является кортежем условий

$$kort = \langle x, y, \Delta t \rangle, \quad (4)$$

где  $y$  является элементом множества выполненных команд  $Y$  эквивалентным множеству  $X$  зарегистрированных команд оператора ( $X \sim Y$ );  $\Delta t$  – интервал времени между командами.

Такое описание удобно при заполнении сценариев проверки экспертом для дальнейшего ис-

пользования в работе продукционными правилами.

Так как обработка информации осуществляется сразу по нескольким сценариям, необходимо ввести понятие диагностической последовательности  $D$ , которая будет использоваться при работе продукционной системы (5). Диагностическая последовательность является динамически изменяемым, помеченным множеством  $D$ , состоящим из элементов  $d$ :

$$D = \{d_1, d_2, \dots, d_i\}. \quad (5)$$

Каждая запись из рабочей памяти попадает в блок поиска сценария по фильтру, где осуществляется цикл сравнения фильтра текущей записи с фильтром каждого элемента, содержащегося в списке активных сценариев. Если поля фильтра совпадают, то управление получает блок анализа записи. В блоке анализа записи происходит анализ возможности перехода отслеживаемого сценария на следующий. Если шаг конечный, то сценарий удаляется из списка и информация сбрасывается в отчет. Кроме того, анализируется время, содержащееся в записи, в случае если временной интервал записи превысил время ожидания элемента списка, то данный элемент удаляется из списка.

### Алгоритм параллельной обработки данных с разбивкой общего массива

Крупными этапами, распараллеливание которых друг относительно друга невозможно в силу их взаимозависимости по данным, являются чтение с жесткого диска в оперативную (рабочую) память, поиск соответствующего активного сценария и обработка данных текущей записи. Параллельную обработку можно реализовать за счет декомпозиции представленных крупных этапов на более мелкие, часть из которых может выполняться параллельно.

Самым простым решением является разделение общего массива данных на части по количеству доступных процессов. Данные каждой части должны обрабатываться независимо. При этом если все записи данной части прочитаны, а в списке активных сценариев останутся незавершенные последовательности, то будет прочитан блок данных из следующей части, и новые сценарии перестанут отслеживаться.

Реализация такой схемы представлена на рис. 1. В ней можно выделить менеджер параллельной обработки, который необходим для обеспечения двух важных задач:

- одновременно не должно выполняться потоков больше, чем процессор содержит ядер;
- ни одно ядро не должно простаивать, если хотя бы один поток находится в режиме ожидания выполнения.

Менеджер проверяет наличие свободных критических ресурсов (процессов в системе). Если свободные ресурсы есть, то производится поиск самого загруженного потока из очереди и менеджер разгружает его, передавая часть данных на обработку освободившемуся потоку. Если очередь потоков пуста, то менеджер находится в режиме ожидания поступления новых потоков (см. рис. 2).

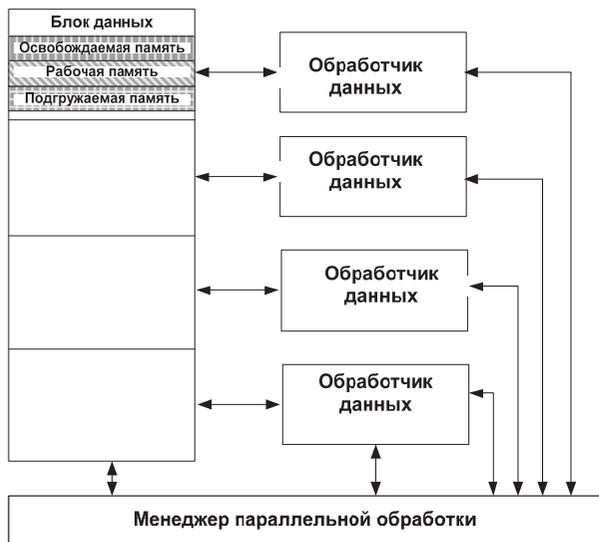


Рис. 1 Схема параллельного алгоритма

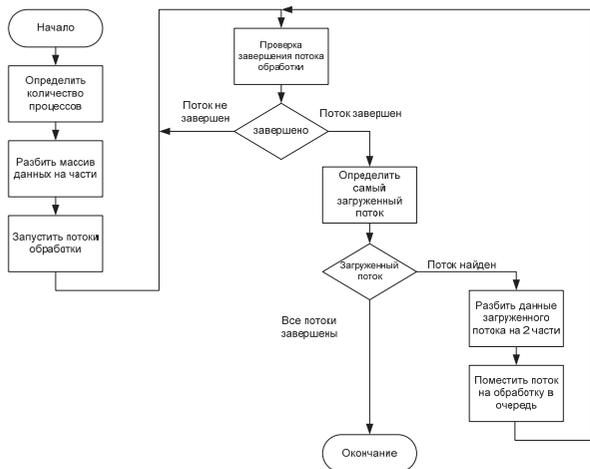


Рис. 2 Алгоритм работы менеджера параллельной обработки

Однако у данного параллельного алгоритма есть недостаток – если начало последовательности окажется в данных одного потока, а окончание в данных другого (см. рис. 3), то алгоритму обработки потока

придется анализировать уже обработанные другими потоками записи. При больших объемах информации этим недостатком можно пренебречь, так как время ожидания новой записи превысит время текущей и сценарий обработки будет удален из списка.

Работу с жестким диском целесообразно организовать разделением области памяти на три части:

- рабочая память (текущие обрабатываемые данные);
- освобождаемая память (обработанные данные, которые необходимо удалить из оперативной памяти);
- подгружаемая память (данные, читаемые с диска в асинхронном режиме).



Рис. 3. Коллизия параллельного алгоритма

Такое разделение позволит избежать простоя всей системы в случае медленного чтения с диска. Чтение диска в асинхронном режиме обеспечивается средствами операционной системы. При этом среднее время работы с диском будет прямо пропорционально обработке данных на самой медленной части конвейера:

$$T_{диск} \sim \max(t_{чтение}, t_{обработка}, t_{освобождение}). \quad (6)$$

В случае медленного чтения с диска значительно (до нескольких раз) сократить время этой операции можно, используя аппаратно-инкрементный подход – быструю систему долговременной памяти (например RAID-массив из высокопроизводительных жестких дисков SAS/SATA с энергонезависимым буфером на запись).

**Алгоритм параллельной обработки данных с динамическим распределением данных по потокам обработки**

Данная реализация снимает нагрузку на операцию чтения с диска (анализ данных в этом случае происходит в один проход), но при этом требует ус-

ложения алгоритма работы менеджера параллельной обработки, однако дает существенный прирост производительности в случае медленной обработки конкретной записи.

Суть данного алгоритма в следующем: менеджер анализирует прочитанную запись. Если запись является условием запуска нового сценария последовательности, то менеджер определяет наименее загруженный поток и передает ему данную запись во входной буфер. При этом в список соответствия добавляется новый элемент, который указывает менеджеру поток для обработки следующей записи последовательности. При этом, если входной буфер потока будет переполнен, менеджер приостановит чтение новых записей, уменьшив вычислительную нагрузку, а при появлении свободных ресурсов продолжит работу.

### Проверка выдвинутых гипотез

Для проверки выдвинутых гипотез необходима разработка модели анализа состояния радиотехнического комплекса, имеющей возможность как последовательной, так и параллельной обработки диагностируемых данных.

В таблицах 1-2 представлены результаты оценки производительности системы, основанной на одноядерном процессоре, и системы, основанной на двух аппаратных платформах: с двух- и четырехъядерным процессором при обработке тестовых массивов малого (11 Мб) и большого (1,1 Гб) объема данных.

Как видно из таблицы 1, производительность последовательного алгоритма не зависит от количества процессоров, а небольшая разница между результатами запусков объясняется оптимизацией функционирования приложений средствами операционной системы в случае многопроцессорной аппаратуры.

Таблица 1. Оценка производительности для объема данных 11 Мб

| Алгоритм   | Среднее время выполнения этапа, мин:С на процессоре с числом ядер |      |      |
|--|---|------|------|
|  | 1   | 2    | 4    |
| Последовательный   | 0:52  | 0:50 | 0:50 |
| Параллельный с разбивкой общего массива данных на части                | 0:52  | 0:48 | 0:46 |
| Параллельный с динамическим распределением данных по потокам обработки | 0:57  | 0:54 | 0:52 |

Для простого параллельного алгоритма с разбивкой массива данных на части имеет место улучшение производительности, обусловленное тем, что в случае небольшого объема информации отдельные процессы работают с данными, фактически находящимися в кэш-памяти дискового массива [3]. Долгое время обработки при использовании параллельного алгоритма с динамическим распределением данных по потокам объясняется большими затратами времени на межпроцессную синхронизацию, которая при небольшом объеме данных занимает значительную часть общего времени обработки. При большом объеме данных (см. таблицу 2) время обработки для последовательного алгоритма возрастает линейно и так же практически не зависит от числа ядер.

Таблица 2. Оценка производительности для объема данных 1,1 Гб

| Алгоритм   | Среднее время выполнения этапа, мин:С на процессоре с числом ядер |       |       |
|--|---|-------|-------|
|  | 1   | 2     | 4     |
| Последовательный   | 87:20   | 86:13 | 85:30 |
| Параллельный с разбивкой общего массива данных на части                | 87:13   | 75:41 | 79:51 |
| Параллельный с динамическим распределением данных по потокам обработки | 87:23   | 69:17 | 63:14 |

При использовании простого параллельного алгоритма с разбивкой массива данных на части происходит деление входного массива между процессами. При этом имеет место большое количество перемещений головок накопителей вследствие того, что объем данных достаточно велик и не помещается в кэш-память дискового контроллера. Эта проблема решается использованием параллельного алгоритма с динамическим распределением данных по потокам, который последовательно считывает данные с диска (что исключает множественные перемещения головок накопителей) и отдает их на обработку параллельным процессам.

Таким образом, были рассмотрены подходы к построению системы анализа данных контроля действий операторов на параллельных и последовательной архитектурах вычислительных систем. В результате проведенного анализа можно сделать

вывод об эффективности использования параллельной обработки с динамическим распределением данных по потокам при больших объемах информации и сложных алгоритмах проверки.

### Литература

1. Верба В.С. Тенденции развития авиационных средств разведки и дозора // Научные технологии. 2004, №8-9, 2004. – С. 53-62.
2. Пащенко Д.В., Васильев А.В., Васильев С.А., Конов Н.Н. Методика контроля показателей

точности сопровождения целей радиолокационным комплексом // Радиотехника. №8, 2009. – С. 116-119.

3. Мачалин В.А., Токарев А.Н., Трокоз Д.А., Пащенко Д.В., Синев М.П. Стратегия параллельной обработки массивов данных в системе объективного контроля радиотехнического комплекса радиолокационного дозора и наведения // Радиотехника. №8, 2010. – С. 51-55.

## ALGORITHMS ANALYSIS OF ACTION OF OPERATORS IN TERRESTRIAL RADIO SET INFORMATION HANDLING SYSTEM

Volchihin V.I., Pashchenko D.V.

**The article compares the serial and parallel algorithms for processing large volumes of information to improve the efficiency of performance analysis of radio operators in a complex system of ground-based information processing, discussed improving performance when reading data from disk using the pipeline.**

*Keywords:* aviation complex radar patrol and guidance systems, radio set, objective monitoring, data file, manager of parallel processing, multi-core processor, the control actions of the operators.

Волчихин Владимир Иванович, д.т.н., профессор, ректор Пензенского Государственного университета (ПГУ). Тел. (8-841) 236-82-27.

Пащенко Дмитрий Владимирович, к.т.н., доцент ПГУ. Тел. 8-927-287-33-32. E-mail: dmitry.pashchenko@gmail.com

УДК 621.391.3

## ТРАНСПОРТНАЯ СЕТЬ ПЕРЕДАЧИ ДАННЫХ ПО ГОРНЫМ ВЫРАБОТКАМ

Кузнецов М.В., Галактионов К.И.

В статье раскрываются альтернативные методики проектирования ведомственных локальных вычислительных сетей (ЛВС), расширяющие область применения технологии Fast Ethernet. В частности, рассмотрен случай создания транспортной сети передачи данных в условиях функционирующей горной выработки (рудника) с учетом местных особенностей.

**Ключевые слова:** ЛВС, Fast Ethernet, ведомственные сети, КСПП 1×4×0,9; частотные характеристики, Power over Ethernet, рудничное оборудование.

В настоящее время функционирование любого предприятия и уровень безопасности труда на нем во многом определяются возможностями и состоянием используемых систем связи и передачи данных. Остро стоит задача создания систем горно-подземной связи на предприятиях горнодобывающих отраслей промышленности. Специфика производства требует внедрения в подземной части шахт (рудников) современных технологий, характеризующихся возможностью

высококачественной передачи значительных объемов оперативной информации при высокой надежности системы. Основой такой системы связи является транспортная сеть передачи данных, к которой предъявляются следующие требования:

- возможность создания на базе обслуживающих шахту информационных систем общешахтной информационной сети, совместимой с существующими системами связи и сигнализации (п.п.540, 545 «Правил безопасности в угольных шахтах» ПБ 05-618-03);

- приспособленность оборудования к эксплуатации в подземных условиях, в том числе при быстро меняющейся топологии горных выработок (возможность быстрой смены мест установки базовых станций, оперативной прокладки новых линий связи и кабелей электропитания);

- соответствие характеристик оборудования требованиям, предъявляемым к рудничному взрывозащищенному электрооборудованию (ГОСТ P51330-99). [1].