

Академія наук України
Інститут кібернетики імені В. М. Глушкова

На правах рукопису

СПАСИТЕЇЛЄВА Світлана Олексіївна

УДК 681.3.06

**МЕТОДИ ТА ЗАСОБИ РОЗПОДІЛУ РЕСУРСІВ
В ОБЧИСЛЮВАЛЬНИХ СИСТЕМАХ
З ЖОРСТКИМИ ЧАСОВИМИ ОБМЕЖЕННЯМИ**

05.13.11 — математичне та програмне забезпечення обчислю-
вальних машин, комплексів, систем та мереж

Автореферат дисертації на здобуття ученого ступеня
кандидата фізико-математичних наук

Київ — 1993

ДВ 20.400

Робота виконана в Інституті кібернетики імені В. М. Глушкова АН України.

ЛНБ України ім. В. Стефаника



00810677 (T)

Науковий керівник: доктор технічних наук, професор
НІКІТИН А. І.

Оційні опоненти: доктор фізико-математичних наук,
професор ЛАВРИЩЕВА К. М.,
кандидат фізико-математичних наук
ДМИТРУК Ю. В.

Провідна установа: Науково-виробниче об'єднання
«Міськсистемотехніка».

Захист відбудеться «26» листопада 1993 р. о 14
год. на засіданні спеціалізованої ради Д 016.45.01 при Інституті кібернетики імені В. М. Глушкова АН України за адресою:

252207 Київ 207, проспект Академіка Глушкова, 40.

З дисертацією можна ознайомитися в науково-технічному архіві інституту.

Автореферат розісланий «20» жовтня 1993 р.

Учений секретар
спеціалізованої ради

СИНЯВСЬКИЙ В. Ф.

ЛНБ ім. В. Стефаника
АН України

7B-28.400 - 1 -

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми. Для широкого кола систем реального часу розробка продуктивного програмного забезпечення стримується відсутністю систематизованого науково обґрунтованого підходу до промислового створення систем, які автоматично налагоджуються на заданий час обчислень.

Розв'язок порушеної проблеми має широкий спектр важливих застосувань, тому будь-які результати теоретичного та практичного характеру становлять певну цінність. Ряд окремих питань, які доцільно використовувати при дослідженні даного напрямку, вирішені в роботах Ліпавца В. В., Шнейдера Б. М., Головкина Б. А., Гуленка В. П., Ермольєва Ю. М., Чичканя І. В., Cohen J., Gabrielian A., McNamee L., Trawick D., Shaw A. C., Metayer D. Проте достатньо узагальнюючих публікацій у напрямку комплексного дослідження згаданих задач немає. Відсутні також інструментальні засоби часового аналізу програм для керуючих систем (КС) з часовими обмеженнями

З розвитком різних складних технічних систем різко збільшується попит на згаданий інструментарій і одночасно підвищується його логічна та структурна складність. Основні зусилля в розробці даного напрямку можна зосередити на розвитку функціональних можливостей систем конкретних галузей застосувань, проте програмне забезпечення автоматизації даного процесу може стати непридатним при зміні часових вимог. Тому необхідне цілісне дослідження проблеми з урахуванням реальних характеристик обчислювальних систем та математичного забезпечення, яке при цьому використовується.

Актуальність підтверджується також тим, що дисертаційні дослідження проведені в межах НДР "Горизонт" та "Електромережа-УН", які виконувались Інститутом кібернетики імені В. М. Глушкова АН України за тематичним планом

Метод дисертаційної роботи є вивчення та формування підходу до промислового створення систем, які автоматично адаптуються до заданого часу обчислень.

Поставлена мета вимагає розв'язку ряду основних задач:

1. Розробка моделі автоматичної адаптації КС до жорстких часових обмежень.
2. Оптимізація розподілу часових ресурсів між функціональними блоками системи.
3. Розробка методів, алгоритмів та інструментальних засобів

часового аналізу програмних комплексів.

Досягнення поставленої мети породжує необхідність розв'язку ряду інших більш часткових задач.

Наукова новизна проведених досліджень та отриманих результатів полягає у такому.

Вперше досліджена алгоритмічна схема адаптації КС до часових обмежень, що задаються; отримано співвідношення точності до часу обчислень у таких системах.

Запропонована методика раціонального використання алгоритмів та засобів часового аналізу програм у поєднанні з кваліфікаційними функціями Макнеї¹.

Запропоновані модель та алгоритми побудови аналітичних часових виразів послідовних та паралельних Ада-програм.

Методика досліджень. Методами прикладної інформатики, на яких в основному ґрунтуються дослідження в цій роботі, є часовий аналіз програм та систем, а основний математичний апарат дослідження - теорія оптимальних керувань.

Практична цінність. На основі розроблених у дисертації моделей і методів автором створено інструментальний програмний комплекс для побудови часових формул Ада-програм, які досить просто включаються в пакети прикладних програм.

Результати даної роботи можуть бути використані при проектуванні програмного забезпечення критичних систем, а отримані залежності точності від часу обчислень та кроку стробування дозволять оптимально вибрати необхідні метод та характеристики.

Апробація роботи. Головні положення дисертаційної роботи доповідалися та обговорювалися на 1-й Всесоюзній конференції "Проблеми создания суперЭВМ, суперсистем и эффективность их применения" (Мінськ, 1987); на 1-й Всесоюзній науково-технічній конференції "Методы анализа надежности программного обеспечения вычислительных систем реального времени на основе моделей нечеткой логики и качественных описаний" (Київ, 1987); на 2-й Всесоюзній науково-технічній конференції "Практическое применение современных технологий программирования, пакетов прикладных программ в вычислительных системах и сетях ЭВМ" (Дніпропетровськ, 1990); на конференції "Математические и программные методы проектирования информационных и управляющих систем" (Ленінград, 1990); на Республіканській науково-технічній конференції "Функционально ориентированные вычислительные системы" (Алушта, 1990); на Всесоюзному

науково-технічному семінарі "Программное обеспечение СМ ЭВМ" (Москва, 1990); на конференції "Радиофизическая информатика" (Москва, 1990); на 5-й Всесоюзній нараді "Надежность, живучесть и безопасность автоматизированных комплексов" (Москва, 1991); на Всесоюзному семінарі "Программное обеспечение новых информационных технологий" (Твер, 1991); на 3-му Всесоюзному семінарі "Качество ПО" (Дагомис, 1991); на семінарах Наукової ради АН України з проблеми "Кібернетика" (Інститут кібернетики ім. В.М.Глушкова АН України, 1986-1993).

Публікації. Головні результати дисертаційної роботи опубліковані в 14 роботах. Крім того, з суміжних питань пошукувач має 4 наукові роботи. Значна частина результатів увійшла до звітів НДР, які виконує Інститут кібернетики ім. В.М.Глушкова АН України.

Структура і об'єм роботи. Дисертація складається із вступу, чотирьох розділів, висновків, списку літератури (90 джерел) та додатку. За обсягом робота складає 124 сторінки, включаючи 14 рисунків, 3 таблиці.

ЗМІСТ ДИСЕРТАЦІЙНОЇ РОБОТИ

У вступі обґрунтовується актуальність проблеми створення систем, які автоматично адаптуються до ресурсних обмежень, обговорюється специфіка таких систем. Характеризується наукова новизна та практична цінність отриманих результатів, обґрунтовується необхідність використання математичних методів та методів прикладної інформатики.

Перший розділ присвячений опису і математичному обґрунтуванню моделі КС, що автоматично адаптується до жорстких часових обмежень. Розділ носить загальний для всієї роботи характер, дослідження в наступних розділах ґрунтуються на отриманих тут математичних залежностях та оцінках.

У підрозділі 1.1 для проектування систем з жорсткими обмеженнями реального часу пропонується використовувати багатoversійний підхід, який полягає в розміщенні у тілі КС великої кількості альтернативних складових елементів з різними показниками. Обґрунтовується переваги вибраного підходу для розв'язання поставленої задачі, можливості його реалізації та склад інструментальних засобів.

У підрозділі 1.2 аналізуються та викладаються критерії розподілу інформаційних ресурсів. На основі їх аналізу пропонуються алгоритми прийняття рішень про розподіл часових ресурсів.

Підрозділ 1.3 включає увесь спектр питань, пов'язаних з оптимізацією розподілу часових ресурсів, починаючи від математичної постановки задачі до отримання всіх необхідних залежностей.

Розглядається процес обробки даних в КС, який оперативно будується із представників N блок-функцій (див.рис.1), послідовність виконання яких відома. Увесь ланцюг обчислень повинен виконуватися за обмежений час T_0 . Кожна i -та блок-функція включає завчасно відоме достатньо велике число альтернативних процедур p_{ij} , які функціонально виконують одні й ті ж операції і відрізняються одна від одної часом обчислень t_{ij} та їх точністю δ_{ij} , де i - номер блок-функції у ланцюгу обчислень, j - номер алгоритму в блок-функції. Іншими словами, в задачі розподілу часових ресурсів при виборі "бажаних" процедур $p_{ij}^{\circ(i)}$ враховується вплив тільки значень t_{ij} та δ_{ij} .

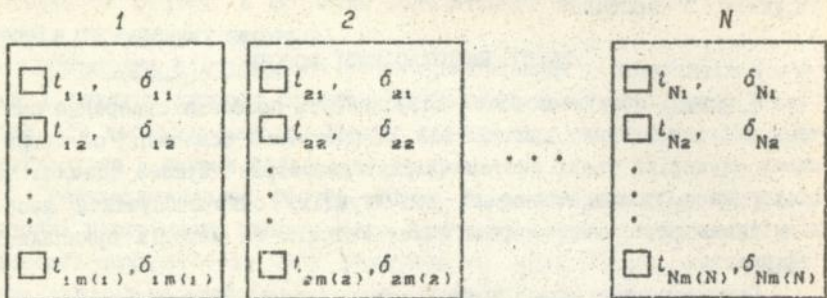


Рис. 1

При відомих значеннях t_{ij} і δ_{ij} ($i = \overline{1, N}$, $j = \overline{1, m(N)}$) необхідно побудувати такий ланцюг обчислень $p_{1j}^{\circ(1)} \rightarrow p_{2j}^{\circ(2)} \rightarrow \dots \rightarrow p_{Nj}^{\circ(N)}$, що складається із представників N блок-функцій, сумарний час виконання якого не перевищує T_0 . При цьому із кожної блок-функції обов'язково вибирається одна і тільки одна із допустимих альтернативних процедур p_{ij}° . Із двох ланцюгів обчислень більш високий пріоритет має той, який забезпечує більш точний кінцевий результат.

У підрозділі 1.3.1 задача розподілу часових ресурсів зведена до задачі оптимального керування.

Лема 1.1 показує, яким чином задачу розподілу часових ресурсів у системах з жорсткими обмеженнями можна завжди звести до задачі оптимального керування в дискретно-різницевої формі:

мінімізувати

$$\Delta(N) \quad (1)$$

при обмеженнях

$$T(k+1) = T(k) + t(\delta(k)), \quad k = \overline{0, N};$$

$$\Delta(k+1) = \Delta(k) + \delta(k),$$

$$T(0) = 0, \quad \Delta(0) = 0; \quad (2)$$

$$\delta(k) \in U(k), \quad k = \overline{0, N-1}; \quad (3)$$

$$T(N) \leq T_0, \quad (4)$$

де $\delta(k) \in U(k)$ - скалярна дискретна змінна значень похибки обчислень альтернативних процедур на k -му кроці;

$$U(k) = \{ \delta_{k+1,1}, \dots, \delta_{k+1,m(k+1)} \}, \quad k = \overline{0, N-1};$$

$$\Delta(k) - \text{змінна стану похибки обчислень на } k\text{-му кроці: } \Delta(k) = \sum_{i=0}^{k-1} \delta(i);$$

$t(\delta(k))$ - дискретна змінна значень часу виконання альтернативних процедур у k -й блок-функції;

$$T(k) - \text{змінна станів часу на } k\text{-му кроці: } T(k) = \sum_{i=0}^{k-1} t(\delta(i)).$$

У підрозділі 1.3.2 для розв'язку задачі оптимального розподілу часових ресурсів застосовується метод "стробування" фазового простору. Він ґрунтується на дискретизації фазової координати точності результатів Δ з кроком h . На кожному кроці із усіх станів, що потрапляють до зони розгляду h , вибирається тільки один.

При розв'язанні задачі розподілу часових ресурсів з використанням методу "стробування" одна із ключових задач полягає у виборі кроку квантування h залежно від точності обчислень, яка задається користувачем. Розв'язок цієї задачі ґрунтується на теоремі 1.1.

Теорема 1.1. Для забезпечення необхідної точності ε розв'язку задачі розподілу часових ресурсів за функцією мети (1) крок квантування за точністю досить взяти рівним $h = \varepsilon/n$.

При досягненні необхідної точності за функцією мети (1) час, отриманий на останньому кроці за методом "стробування", може відрізнитися від оптимального на величину, яка не задовольняє користувача. Тому в роботі отримана така оцінка.

Теорема 1.2. Сума розподілених часових ресурсів $\bar{T}_h(N)$ відхиляється від абсолютно оптимального значення часу $T(N)$ на величину, яка не перебільшує ε : $|\bar{T}_h(N) - T(N)| \leq \varepsilon$.

де L - максимальна із констант Ліпшиця функцій $t(\delta(k))$, $k = \overline{0, N-1}$.

Висновок. Для отримання потрібної точності ϵ розв'язку задачі 1.3 за функцією мети (1) фазове обмеження (4) повинне при відсіванні недопустимих варіантів мати вигляд

$$T(N) \leq T_0 + L\epsilon.$$

При цьому $h = \frac{\epsilon}{N}$.

Теорема 1.3. Для забезпечення потрібної точності ϵ розв'язку задачі за функцією мети (1) та потрібної точності ϵ_1 уловлювання фазового обмеження $T(N) \leq T_0$ крок квантування за

$$\min \left\{ \frac{\epsilon_1}{L}, \epsilon \right\}$$

точність досить взяти рівним $h = \frac{\min \left\{ \frac{\epsilon_1}{L}, \epsilon \right\}}{N}$.

З метою можливості оцінки зваженого сумарного критерію з урахуванням порушення фазових обмежень пропонується застосувати метод штрафних функцій.

У підрозділі 1.3.3 розглядається при попередніх обмеженнях (2), (3) нова функція мети вигляду

$$\Phi(N) = \Delta(N) + \frac{\Delta^*(N) - \Delta_{\min}(N)}{\epsilon_1} f(T(N) - T_0), \quad (5)$$

де не диференційована функція $f(T(N) - T_0) = \begin{cases} 0, & \text{якщо } T(N) - T_0 \leq 0 \\ T(N) - T_0, & \text{у протилежному випадку,} \end{cases}$

$\Delta^*(N)$ - значення функції мети (1) для допустимого розв'язку задачі (1) - (4), $\Delta_{\min}(N)$ - оцінка знизу мінімуму функції (1) при відсутності обмеження (4); ϵ_1 - точність виконання обмеження (4). Значення $\Delta^*(N)$ і $\Delta_{\min}(N)$ знаходяться таким чином:

$$\Delta^*(N) = \sum_{t=0}^{N-1} \max_{1 \leq j \leq m(t)} \delta_{tj}, \quad \Delta_{\min}(N) = \sum_{t=0}^{N-1} \min_{1 \leq j \leq m(t)} \delta_{tj}.$$

Оптимальний розв'язок задачі (2), (3), (4) позначасмо $\bar{\Phi}_{\epsilon_1}(N)$, $\bar{\Delta}_{\epsilon_1}(N)$, а отримане при цьому сумарне значення часу - $\bar{T}_{\epsilon_1}(N)$. Мабуть місце такі твердження.

Теорема 1.4. Отриманий оптимальний розв'язок задачі (2), (3), (5) не гірше за функцією мети розв'язку початкової задачі (2) - (4), тобто $\bar{\Phi}_{\epsilon_1}(N) \leq \bar{\Delta}(N)$, а фазове обмеження (4) буде

виконано з точністю до ε_1 , тобто $T_{\varepsilon_1}(N) - T_0 \leq \varepsilon_1$.

Теорема 1.5. Якщо в розв'язку задачі задана ціна \mathcal{C} одиниці простроченого часу, то оптимальні кроки квантування за фазовими змінними $\Delta(k)$ і $T(k)$, $k = \overline{1, N}$, є відповідно

$$h_{\Delta} = \frac{\mathcal{C}}{2N}, \quad h_T = \frac{\mathcal{C}}{2\mathcal{C}N}. \quad (6)$$

Висновок 1. Похибка ε_1 задовольняння нерівності $T(N) \leq T_0$

при цьому оцінюється так: $\varepsilon_1 \leq \frac{\Delta^*(N) - \Delta_{\min}(N)}{\mathcal{C}}$.

Висновок 2. Діапазонами фазових змінних можна взяти

$$D_{\Delta} = \Delta^*(N), \quad D_T = T_0 + \varepsilon_1.$$

Автоматична адаптація КС до жорстких часових обмежень передбачає наявність засобів автоматизації часового аналізу довільних програм, які написані мовами високого рівня. Тому в другому розділі досліджуються питання, пов'язані з часовими характеристиками програм.

У підрозділі 2.1 узагальнюється досвід визначення часових характеристик і властивостей програм, вивчається можливість та досліджуються підходи до автоматизації часового аналізу програм. Формується систематизована методологія отримання часових характеристик програм з метою автоматизації розподілу часового ресурсу між блок-функціями при жорстких обмеженнях.

У підрозділі 2.1.1 проведено аналіз відомих методів та автоматизованих засобів прогнозування часових характеристик виконання програм. Аналізуються системи METRIC, OPAJ, ACE, ARGUS, SDS, SADAT, що містять засоби опису часової поведінки програм.

Існують два шляхи часового аналізу програм - експериментальний і аналітичний. Експериментальний аналіз полягає у прогоні програм на ЕОМ з різними допустимими наборами вхідних даних і встановленні контрольних точок (часової розмітки) часу її виконання. Аналітичний підхід можна поділити на дві категорії: мікроаналіз і макроаналіз.

При порівнянні методів макро- і мікроаналізу програм, відмічається, що мікроаналіз забезпечує більш точний часовий прогноз, ніж макроаналіз. Макроаналіз дає тільки залежність (порядок росту) часу виконання від об'єму вхідних даних, а мікроаналіз дозволяє отримати числові значення часу виконання програм у вигляді параметрів: максимального, мінімального, середнього часу виконання

програми та середньоквадратичного відхилення від середньої величини. Причому на першому етапі аналізу при побудові часових формул можна не враховувати об'єм вхідних даних і тип EOM, на якій буде виконуватися програма, що піддається аналізу. І тільки на другому етапі часові змінні пов'язуються з дійсними числовими величинами.

Автоматизація мікроаналізу з отриманням усіх чотирьох часових характеристик є громіздкою і часто не може бути реалізованою, тому що обчислення середнього значення часу виконання програм і середньоквадратичного відхилення від середньої величини вимагають знання функції "типового" розподілу вхідних даних і обчислення відповідних імовірних значень. До того ж невдалий вибір розподілу може суттєво викривити результат аналізу.

У зв'язку з вищевикладеним реалізація часового аналізу вдавалася тільки для простих послідовних програм. На програми, які аналізувалися, накладалося ряд обмежень відносно форми зображення і структури програм. Існуючі підходи до реалізації систем часового аналізу програм передбачають участь спеціаліста (краще всього автора програми), який добре знає програмні модулі, до якому необхідно задавати множину параметрів блоків програми, їх взаємозв'язки, імовірності виконання гілок програми і т.п. Якість отриманих оцінок розрахунку часу виконання програм пропорційно залежить від компетентності спеціаліста.

У підрозділі 2.1.2 автор виходячи з результатів дослідження у галузі часового аналізу програм пропонує шляхи подальшого розвитку методів прогнозування часу виконання програмних систем та створення на їх основі часових алгоритмів для роботи у системах з жорсткими часовими обмеженнями. Обмежене використання у таких системах відомих методів та автоматизованих засобів прогнозування часових характеристик виконання програм пояснюється тим, що до теперішнього часу не вирішено ряд проблем, зокрема пов'язаних з розрахунком часу функціонування програмних систем, які містять як послідовно, так і паралельно виконувані модулі. До програм з паралельними процесами ставляться вимоги, пов'язані з прогнозуванням часу виконання не тільки їх функціональних компонент, але й процесів синхронізації та обміну інформацією. У зв'язку з цим виникла необхідність аналізу операторів і алгоритмів, які включають семафори, рандеву, засоби блокування, передачу повідомлень. Для досить широкого класу застосувань, включаючи КС з жорсткими

часовими обмеженнями, необхідно розробити засоби для формального опису часової поведінки програмних систем, написаних мовами високого рівня, і побудови часових формул їх виконання.

У підрозділі 2.2 описана систематизована методологія отримання часових характеристик програм, запропоновані алгоритми побудови часових формул програм, написаних мовами високого рівня. В основу розроблених алгоритмів покладено поєднання методів часового мікроаналізу по визначенню граничних часових значень виконання програм і формальних методів аналізу поведінки систем реального часу, які використовують кваліфікаційні функції Макнеймі. Для зручного зображення часу виконання оператора S та псев'язаних з ним перетворень вектора змінних \bar{X} програми вводиться формальний механізм, який ґрунтується на трансформаційних і часових функціях - спеціальному вигляді кваліфікаційних функцій. Часова функція визначається таким чином.

Означення. Нехай S - оператор програми, I - скінченна множина, \bar{X} - вектор змінних до виконання оператора S . Тоді $T(S; \bar{X})$ - часова функція виконання оператора S з відніми змінними \bar{X} .

$$T(S; \bar{X}) = \left\{ \left(\mathcal{E}_i, T(S_i; \bar{X}) \right) \right\}_{i \in I}$$

де $\{\mathcal{E}_i\}$ - множина логічних умов, які визначають наступні гілки у графі виконання складного оператора S (рис.2), $T(S_i; \bar{X})$ - часова функція, пов'язана з виконанням оператора $S_i \in S$. Спосіб обчислення $T(S; \bar{X})$ для отримання результату $\tilde{T}(S_i; \bar{X})$

задається формулою $\tilde{T}(S; \bar{X}) = \sum_{\mathcal{E}_i = \text{істина}} \tilde{T}(S_i; \bar{X})$,

де $\tilde{T}(S_i; \bar{X}) = T(S_i; \bar{X})$, якщо $T(S_i; \bar{X})$ - арифметичний вираз.

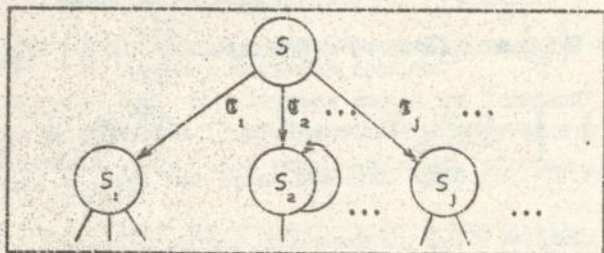


Рис.2. Граф виконання складного оператора S

У підрозділі 2.2.1 запропоновані методи визначення часу виконання послідовних програм. Описана побудова часових функцій і граничних значень часу виконання лінійної програми, яка містить оператори присвоєння, виклики процедур, а також програми, що містять типові умовні оператори (*if*, *case*), оператори циклів (*for*, *while*). Як приклад наведемо побудову часової функції умовного оператора *if* з припущенням, що S_1, \dots, S_{n+1} - прості оператори. \bar{X} - вектор змінних:

```

S_if ≡ if B_1 then S_1(X);
      . . .
      elsif B_n then S_n(X);
      elsif S_{n+1}(X);
      endif;
    
```

$$T(S_{if}; \bar{X}) = \left\{ \left(\text{Істина}, T(B_1) + T(\text{elsif}) \right) \cup \left\{ \left(B_1, T(S_1; \bar{X}) \right), \right. \right. \\ \left. \left(\neg B_1, T(B_2) + T(\text{elsif}) \right), \left(\neg B_1 \wedge B_2, T(S_2; \bar{X}) \right), \right. \\ \left. \left(\neg B_1 \wedge \neg B_2, T(B_3) + T(\text{elsif}) \right), \dots, \left(\neg B_1 \wedge \dots \wedge \neg B_n, T(S_{n+1}; \bar{X}) \right) \right\},$$

де $T(\text{elsif})$ - накладні часові витрати, пов'язані з переходами всередині оператора, $T(B_i)$ - часова функція для обчислення та перевірки булевої умови B_i .

При мінімальному об'ємі інформації про вхідні дані, контекст програми, компілятор, версія машинного часу не завжди можливо апіорі визначити час виконання програми S . Проте можна отримати граничні значення часу її виконання: $t_{\min}(S)$ і $t_{\max}(S)$. У цьому випадку $T(S) = [t_{\min}(S), t_{\max}(S)]$. Граничні часові значення при виконанні умовних операторів знаходяться таким чином:

$$T_k(S_{if}) = T(S_k) + \min(k+1, n) * T(\text{elsif}) + \sum_{i=1}^n T(B_i) = [t_{k_1}, t_{k_2}].$$

$$\text{То } T(S_{if}) = \left[\min \left(\min(t_{k_1}), n * T(\text{elsif}) + \sum_{i=1}^n T(B_i) \right), \max_k(t_{k_2}) \right].$$

де $T(\text{elsif})$ - граничні значення часу, пов'язані з переходами в операторі.

У підрозділі 2.2.2 запропоновані методи отримання оцінок часу виконання паралельних обчислень. Проведення часовий аналіз

операторів обліку часу, операторів затримки *delay*, періодичних процесів. Розглядаються способи отримання граничних значень часу встановлення рандеву для паралельних процесів.

Третій розділ присвячений інструментальним засобам та моделям часового прогнозу роботи програмних систем.

У підрозділі 3.1 запропонована структура інструментальних засобів (див.рис.3), яка розроблена з урахуванням принципу розмежування процесу побудови часових формул програм і отримання часових оцінок їх виконання. Інструментальні засоби забезпечуть для довільно взятої програмної системи, написаної мовою високого рівня типу Ада, автоматичну генерацію відповідних їй часових формул. Така генерація ґрунтується на аналізі графа керування і застосування розглянутих у 2-му розділі алгоритмів часового аналізу програм. Після вводу і аналізу вхідних даних визначаються оцінки часу роботи даної програмної системи, виключачи модулі, які містяться у ній, і видаються відповідні часові значення.

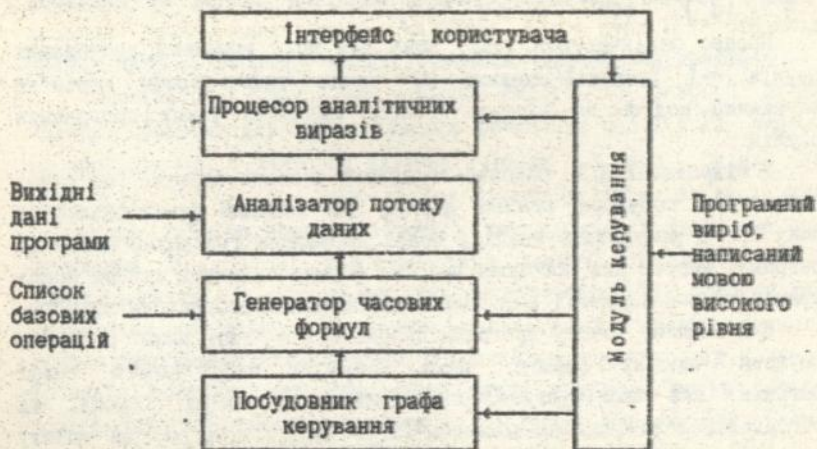


Рис. 3. Структура програмних засобів часового прогнозу виконання програм

У підрозділі 3.2 пропонується модель та "спадний" алгоритм "оберненого проектування" для отримання прямо із тексту програми його структурного графа, який розпізнає модулі програми і показує їх керувчі взаємодії.

Алгоритм побудови структурного графа є ітераційним. Для кожної ітерації будується ітераційна множина $S_i = \{a_1, \dots, a_k\}$, де

i - номер ітерації, q_1, \dots, q_k - корінні модулі. На першій ітерації корінними програмними модулями є компілювчі модулі, які можуть бути або процедуров без параметрів або пакетом.

Модель MI програми на i -й ітерації окреслюється такими значеннями $(S_i, Q, MQ^i, MT^i, f_{MQ}, f_{EV}, f_{ET})$, де Q - множина програмних модулів, які описані у програмі; MQ^i - множина імен програмних модулів, які викликаються корінними модулями на i -й ітерації; MT^i - множина активних задач, які описані у корінному модулі; f_{MQ} - функція, яка відображає кожний корінний програмний модуль на i -й ітерації у множину MQ^i ; f_{EV} - функція, яка відображає кожний пакет $q_j \in Q$ у множину описаних у ньому входів EV ; f_{ET} - функція, яка відображає кожну задачу $q_j \in Q$ у множину описаних в її входів. MI можна представити графом $G(MI) = (N, A)$, де N - множина вузлів, $N = \{S_i \cup MQ^i \cup MT^i\}$; A - множина всіх можливих пар ребер (q_k, q_j) , $q_k \in S_i$ і $q_j \in f_{MQ}(q_k)$ або $q_j \in f_{EV}(q_k)$ або $q_j \in f_{ET}(q_k)$.

Процес закінчується тоді, коли для всіх корінних програмних модулів k -ї ітерації множина MQ^k пуста, тобто жоден корінний програмний модуль не містить у собі викликів інших програмних модулів.

У підрозділі 3.3 описано генератор часових формул, що призначений для побудови часових формул для кожного функціонального блоку КС. В результаті аналізу графа керування генератор визначає програмні засоби для побудови часових формул складних мовних конструкцій та послідовно і паралельно виконуваних програмних модулів.

Для аналізу тексту програми пропонується "висхідний" алгоритм побудови часових формул. Якщо побудова структурного графа програми, яка аналізується, закінчилась на k -му кроці, то розглядається ітераційна множина $S_k = \{q_1, \dots, q_m\}$. Із тексту програми послідовно вибираються програмні модулі q_1, \dots, q_m і для кожного із них будуються часові формули у відповідності з алгоритмами часового аналізу. При цьому програмним блокам "begin ... end;" і складним операторам мови присвоюються порядкові номери, які дозволять згодом при необхідності звертатися до кожного із них за ім'ям програмного модуля і відповідним номером для моделювання їх часової поведінки. Таким чином виконувча частина кожного i -го, $i=1, m$, програмного модуля переписується у вигляді

ді часових формул і запам'ятовується як процедура з відповідним ім'ям і вхідними параметрами. Кожний програмний модуль також містить інформацію про змінні, їх тип і зону дії, а також список блоків і складних програмних конструкцій.

На наступному $(k-1)$ -му кроці розглядаються ітераційна множина $S_{k-1} = \{q_1, \dots, q_n\}$ і функція f_{MO} . Так само як і на попередньому кроці, будуються часові формули для кожного програмного модуля $q_i \in S_{k-1}$, $i = \overline{1, n}$. Якщо програмний модуль $q_i \in S_{k-1}$ містить виклик програмного модуля $q_j \in S_k$, викликається процедура отримання часової формули q_j , у якій формальні параметри замінюються на фактичні. Таким чином відбувається побудова часових формул, починаючи з k -го і закінчуючи 1 -м кроком, на якому отримуємо часову формулу для всієї програми.

У підрозділі 3.4 описано процесор аналітичних виразів, який дозволяє на основі вхідних даних генератора часових формул і вхідних даних аналізуючої програми (блок-функції) отримувати оцінку часу її роботи у вигляді числових значень чи графіків.

Аналізатор потоку даних у відповідності із списком вхідних змінних програми дозволяє процесору обчислити час $T(S; \bar{X})$ виконання програми S з вектором вхідних змінних \bar{X} або оцінку часу $T(S) = [t_{\min}, t_{\max}]$, якщо відомі діапазони зміни вхідних даних. Ці оцінки пізніше використовуються при розподілі часу між блок-функціями КС. Часові характеристики різних алгоритмів обробки даних залежать від типу даних, які обробляються. Час обробки даних динамічних процесів може суттєво залежати від одного параметра, який в подальшому називатимемо головним, наприклад від кількості об'єктів обробки, від обмеженого числа типів об'єктів обробки, розмірності обчислювальних операцій. У такому випадку при заданому діапазоні зміни головного параметра $p_i^{SOA} = [p_{\min}^{SOA}, p_{\max}^{SOA}]$ і-ї блок-функції (програми) можна автоматично отримати граничні значення часу її виконання $T_i = [t_{\min}, t_{\max}]$. Також можна побудувати графік залежності часу від головного параметра $T_i = f(p_i^{SOA})$. Якщо неможливо виділити головний параметр, то можна визначити "кращий" і "гірший" випадки за часом обчислень. Це відноситься до змонтованих систем зняття і обробки інформації з апаратури і прийняття керуючих рішень.

У четвертому розділі розглянуті питання реалізації запропонованих у роботі алгоритмів побудови часових формул Ада-програм!

промодельовано процес розподілу часових ресурсів між функціональними блоками системи.

Для автоматичного отримання часових формул Ада-програм у середовищі BORLAND PASCAL реалізовано інструментальний комплекс, що являє собою набір програм для отримання граничних значень часу виконання і побудови часових функцій програм. Розроблений комплекс досить просто може включатися в пакети прикладних програм, тому що є реентерабельним. На вхід повинен подаватися файл в текстовій формі програми, на виході видається результат у вигляді файлу часових формул, який може в подальшому використовуватися будь-яким чином. Побудова часових формул здійснюється на основі тексту програми і списку базових операцій.

Викладено процес отримання граничних значень часу виконання програм. Описані питання використання розробленого інструментального комплексу за допомогою програмного посередника TIME T.

У підрозділі 4.1 на прикладі програмної системи, що визначає час виробництва готових виробів із партії заготовок, показано процес отримання часових формул. При цьому використовуються "спадний" алгоритм отримання структурного графа програмної системи і "висхідний" алгоритм побудови часових формул системи на основі розглянутих вище засобів часового аналізу програм.

У підрозділі 4.2 на прикладі промодельована робота алгоритму розподілу часу між функціональними блоками системи.

Результати моделювання підтвердили правильність теоретично отриманих у роботі оцінок результатів обчислень за точністю і отримання найкращого із можливих розподілів часового ресурсу.

У висновку наведені головні результати роботи.

У додатку подані тексти програм, результати експериментів і моделювання та документи про використання результатів дисертаційної роботи.

ГОЛОВНІ РЕЗУЛЬТАТИ РОБОТИ

1. Розроблена алгоритмічна схема адаптації керуваних систем до часових обмежень, які задаються ззовні. Запропонована схема ґрунтується на багатoversійній архітектурі системи та на можливості оперативного вибору відповідних альтернативних програмних одиниць з кращими із можливих показників точності залежно від часу, що виділяється користувачем.

2. Запропоновано підхід до розв'язку задачі розподілу часових ресурсів між функціональними блоками системи як дискретної задачі оптимального керування. При цьому встановлені крок квантування

$$\min \left\{ \frac{\epsilon_1}{L}, \epsilon \right\}$$

фазового простору $h = \frac{\epsilon_1}{N}$ та оцінка відхилення розподіленого часу $T(N)$ від заданого значення T_0 : $|T(N) - T_0| \leq L\epsilon$, де N - кількість функціональних блоків у ланцугу обчислень; ϵ - допустима похибка уловлювання сумарної точності обчислень; ϵ_1 - точність виконання фазового обмеження $T(N) \leq T_0$; L - максимальна із констант Ліпшица функції часу виконання альтернативних процедур $t(\delta(k))$, $k = 0, N-1$. При заданій ціні ϵ одиниці простроченого часу запропоновано застосовувати метод штрафних функцій і встановити кроки квантування фазових змінних

$$h_\Delta = \frac{\epsilon}{2N}, \quad h_T = \frac{\epsilon}{2\epsilon N}$$

3. Проведений аналіз відомих методів прогнозування часових характеристик виконання програм показав їх обмежене застосування у системах з жорсткими часовими обмеженнями. Запропоновані методика та алгоритми раціонального використання методів, яка поєднує застосування кваліфікаційних функцій Макнеймі з апаратом часового аналізу. Описані умови та широкий клас задач, для якого можливе застосування методики.

4. Розроблені "спадний" алгоритм отримання із тексту програми її структурного графа, який визначає програмні модулі та показує їх керульчі взаємодії, і "висхідний" алгоритм побудови часових формул програм.

5. Розроблені алгоритми часової оцінки послідовних і паралельних програм, написаних мовами високого рівня. Запропоновані методи побудови часових формул програм.

6. Розроблені модель генератора часових формул та модель процесора аналітичних виразів, які забезпечують отримання часових характеристик програм, написаних мовами високого рівня.

7. На основі отриманих результатів у середовищі BORLAND PASCAL на IBM PC реалізовано інструментальний програмний комплекс для побудови часових формул Ада-програм.

* * *

Головні положення дисертації опубліковані в таких роботах:

1. Чичкань Н.Б., Спасителева С.А. О прогнозе работоспособности ЭВМ на заданное время вперед // Проблемы создания супер-ЭВМ.

- супер-систем и эффективность их применения: Тез. докл. 1-й Всесоюз. конф. - Минск: ИМ АН БССР. - 1987. - Ч.2. - С.198-199.
2. Чичкань И. В., Спасителева С. А. Оценка качества функционирования программного обеспечения в условиях временных ограничений // Методы анализа надежности программного обеспечения вычислительных систем реального времени на основе моделей нечеткой логики и качественных описаний: Тез. докл. 1-й Всесоюз. науч.-техн. конф. - Киев: КИИГА, 1987. - С. 73.
3. Спасителева С. А., Чичкань И. В. Разработка систем моделирования внешней обстановки и критерии эффективности работы комплекса // Исследование вопросов оперативного управления комплексом и взаимодействие средств комплекса в условиях противодействия. - Киев: Ин-т кибернетики им. В.М.Глушкова АН УССР. - 1988. - С. 48-79.
4. Спасителева С. А., Чичкань И. В. Обработка особых ситуаций в реальном масштабе времени // Практическое применение современных технологий программирования, пакетов прикладных программ в вычислительных системах и сетях ЭВМ: Тез. докл. 2-й Всесоюз. науч.-техн. конф. - Днепропетровск: НПО "Орбита", 1990. - С. 20-22.
5. Спасителева С. А., Чичкань И. В. Средства оценки по времени выполнения программ заданного типа // Математические и программные методы проектирования информационных и управляющих систем. - Пенза: ПДНТП, 1990. - С. 64-69.
6. Спасителева С. А., Чичкань И. В. Алгоритмы распределения времени и временного контроля в информационно-управляющих системах радиофизических комплексов // Функционально ориентированные вычислительные системы: Тез. докл. респ. науч.-техн. конф. - Харьков: ХПИ, 1990.-Ч.1. Теория и программные средства. - С. 123-124.
7. Спасителева С. А., Чичкань И. В. Компромиссные решения при распределенной обработке данных динамического процесса в условиях временных ограничений // Программное обеспечение СМ ЭВМ: Тез. докл. Всесоюз. науч.-техн. семинара. - М.: ИНЭУМ, 1990. - С. 97-98.
8. Спасителева С. А., Чичкань И. В. Оперативное планирование структур системы обработки данных динамического процесса при ограниченном времени // Радиофизическая информатика: Тез. докл. конф. - М.: РТИ АН СССР, 1990. - С. 46-47.
9. Спасителева С. А. Критерии распределения ресурсов в системах с жесткими временными ограничениями // Практическое применение современных технологий программирования, пакетов прикладных прог-

рам в вычислительных системах и сетях ЭВМ: Тез. докл. 2-й Всесоюз. науч.-техн. конф. — Днепропетровск : НПО «Орбита», 1990. — С. 127—130.

10. Чичкань И. В., Спасителева С. А. Проблема временной адаптации автоматизированных комплексов для обеспечения надежности обработки данных // Надежность, живучесть и безопасность автоматизированных комплексов: Тез. докл. V Всесоюз. совещ. — М. : ИПУ АН СССР, 1991. — С. 69—71.

11. Чичкань И. В., Спасителева С. А. Инструментальные средства для автоматической адаптации сложных систем к жестким временным ограничениям // Программное обеспечение новых информационных технологий: Тез. докл. Всесоюз. семинара. — Тверь : НТО «Центрпрограмм-систем», 1991. — С. 165—166.

12. Спасителева С. А. Методы прогнозирования времени вычислений в инструментальном комплексе Технопрос-Т // Качество ПО: Тр. Всесоюз. семинара. — М. : СНПО «Алгоритм», 1991. — С. 38—41.

13. Чичкань И. В., Спасителева С. А. Автоматическая адаптация информационно-управляющих систем к жестким временным ограничениям // УСиМ. — 1992. — № 7/8. — С. 41—49.

14. Чичкань И. В., Спасителева С. А. Программно-технические средства для ввода информации из УПДМЛ в графопостроитель // УСиМ. — 1987. — № 3. — С. 118—121.

Підп. до друку 10.06.93. Формат 60×84/16. Папір друк. № 2. Офс. друк. Ум. друк. арк. 0,93. Ум. фарбо-відб. 1,05. Обл.-вид. арк. 1,0. Тираж 100 прим. Зам. 1018.

Редакційно-видавничий відділ з поліграфічною дільницею
Інституту кібернетики імені В. М. Глушкова АН України
252207 Київ 207, проспект Академіка Глушкова, 40

463910

AB 28.400

AB 28.400