

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



Наукові праці
Донецького національного технічного університету
Серія: «Проблеми моделювання
та автоматизації проектування»

№ 1 (15), 2020

Покровськ – 2020

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАУКОВІ ПРАЦІ

**ДОНЕЦЬКОГО НАЦІОНАЛЬНОГО ТЕХНІЧНОГО
УНІВЕРСИТЕТУ**

**Серія «Проблеми моделювання та автоматизації
проектування»**

Всеукраїнський науковий збірник

Заснований у червні 1999 року

Виходить 2 рази на рік

№ 1 (15), 2020

Покровськ – 2020

Публікується згідно з рішенням Вченої ради державного вищого навчального закладу «Донецький національний технічний університет», протокол № 2 від 27.02.2020

У збірнику опубліковано статті співробітників факультету комп'ютерних наук та технологій ДонНТУ, а також деяких інших університетів, які є науковими партнерами ДонНТУ і активно співпрацюють у спільних дослідженнях. В публікаціях наведено результати наукових досліджень та розробок, виконаних у 2019-2020 роках за наступними напрямками: теоретичні аспекти моделювання та автоматизації проектування, методи та засоби моделювання, методи та засоби автоматизації проектування, спеціалізовані обчислювальні системи моделювання, контроль процесів та керування динамічними системами. Матеріали збірки призначені для викладачів, наукових співробітників, інженерно-технічних працівників, аспірантів та студентів, що займаються питаннями дослідження, моделювання та розробки складних динамічних систем.

Засновник та видавець - Донецький національний технічний університет

РЕДАКЦІЙНА КОЛЕГІЯ

Професор, д. т. н., В.А. Святний (головний редактор, відп. за випуск), доцент, к. т. н., С.О. Цололо (заст. головного редактора), доцент, к. т. н., В.В. Шамаєв (заст. головного редактора), Prof. Dr. of Sc., A. Barkalov, Prof. Dr.-Ing. habil. Dr. h.c., S. Heinrich, Prof. Dr.- Ing. habil., A. Kienle, Prof. Dr.- Ing. habil., W. Ehlers, Prof. Dr. rer. nat. Dr. h. c., K. Rothermel, доцент, к. т. н., С.О. Ковальов, професор, д. т. н., О.А. Дмитрієва, професор, д. т. н., Н.М. Куцуль, доцент, к. т. н., Ю.Л. Дікова, ст. викладач, О.Г. Шевченко, аспірант, К.С. Гайдук.

Збірник зареєстрований в Державному комітеті інформаційної політики, телебачення та радіомовлення України. Свідоцтво КВ № 7377 від 03.06.2003.

Збірник «Наукові праці Донецького національного технічного університету» заснований в червні 1999 р. Випуск 212.

ЗМІСТ

Svyatnyy V., Kushnarenko V. Ein Ansatz zur gleichmäßigen Lastverteilung zwischen Prozessoren des MIMD-Simulators der dynamischen Netzobjekte mit verteilten Parametern	5
Баркалов А.А., Титаренко Л.А., Ковалев С.А. Реализация схем автоматов Мили на счётчиках	15
Shcherbakov O., Polsterer K., Svyatnyj V. Integration of Distributed Parallel Simulation Environment with Cloud-Infrastructures	23
Кучерявенко А. Розробка веб-додатку для моделювання сипких матеріалів методом дискретних елементів	29
Зайцев К.С., Шамаєв В.В. Аналіз та розробка альтернативних рішень інфрачервоної пайки	33
Новіков В.Г. Використання одноплатних комп'ютерів для моделювання процесів шахтних вентиляційних мереж	40

EIN ANSATZ ZUR GLEICHMÄSSIGEN LASTVERTEILUNG ZWISCHEN PROZESSOREN DES MIMD-SIMULATORS DER DYNAMISCHEN NETZOBJEKTE MIT VERTEILTEN PARAMETERN

Svyatnyy V. A., Kushnarenko V. G.

Dynamische Netzobjekte mit verteilten Parametern (DNOVP) sind in verschiedenen Gegenstandsgebiete die Objekte der Untersuchung, Projektierung, Automatisierung, Überwachung, Qualitätssicherung, optimalen Prozessführung, Sicherheitsanalyse und -vorhersage, Vermeidung der sicherheitskritischen Betriebszustände. Die Methoden und Mittel der Modellierung und Simulation dieser Objektklasse haben sowohl bei der Projektierung als auch während des Betriebes eine zunehmende theoretische und praktische Bedeutung. Bei der Entwicklung der parallelen DNOVP-MIMD-Simulatoren soll aktuelle Aufgabe der gleichmäßigen Lastverteilung zwischen beteiligten Prozessoren gelöst werden. Im Beitrag wird mögliche Parallelitätsebene des DNOVP „Eine Kante des Netzgraphen – ein MIMD-Prozess/Prozessor“ hinsichtlich der Lastverteilung betrachtet und Ansatz zur Balancierung der Belastungen vorgeschlagen. Idee der Balancierung besteht in der Zuordnung „Die minimallänge Kante – ein Prozess“ und Formalisierung des DNOVP-Simulationsmodells entsprechend darauffolgender sekundären Topologie sowie der Anwendung des blockartigen numerischen Verfahrens (BAV).

Schlüsselwörter: *Dynamisches Netzobjekt mit verteilten Parametern (DNOVP), virtuelle Parallelitätsebene, Lastverteilung, paralleler MIMD-Simulator, blockartiges numerisches Verfahren (BAV).*

1. Modell und Simulationsmodell des DNOVP, die Parallelitätsebenen und Aufgabestellung. Modell der dynamischen Prozesse in der j-Kante des DNOVP wird mit den Gleichungen

$$\begin{cases} -\frac{\partial P_j}{\partial \xi} = r_j Q_j^2 + \frac{\rho}{F_j} \frac{\partial Q_j}{\partial t} + r_j(\xi, t) Q_j^2 \\ -\frac{\partial P_j}{\partial t} = \frac{\rho a^2}{F_j} \frac{\partial Q_j}{\partial \xi}, \end{cases} \quad (1)$$

beschrieben [1]. Hier sind: P_j , Q_j – Druck und Luftstrom der Koordinate ξ entlang, die von Anfangs (AKI)- bis zum End(EKK)-Knoten errechnet wird; r_j – spezifischer aerodynamischer Widerstand; F_j – die Querschnittsfläche der Kante (Luftwegstrecke); ρ – Luftdichte; a – die Schallgeschwindigkeit im Luft; $r_j(\xi, t)$ – regelbarer Widerstand; ξ – die Ortskoordinate des regelbaren Widerstands (z. B., ein Schieber). Die Randbedingungen für (1) sind die Druckfunktionen P_{AKI} , P_{EKK} in den Knoten der j-Kante. Es sind drei Kanten- und Knotenarten nach Randbedingungen in DNOVP zu unterscheiden:

- die Kanten, die den inneren n_1 DNO-Knoten inzident sind; hier werden die

Druckwerten während des Lösens des DNOVP-Gleichungssystems entsprechend den dynamischen Knotenbedingungen

$$-\frac{\partial P_{wi}}{\partial t} = \frac{\rho a^2}{F_{wi}} \frac{\partial Q_{wi}}{\partial \xi} \quad (2)$$

berechnet; hier sind P_{wi} – Druck in wi -Knoten; Q_{wi} – Gesamtluftstrom durch wi -Knoten; F_{wi} – Querschnittsfläche des Knotenraums;

- die Kanten, die den n_2 Knoten der Aktivelementanschlüsse (AEJ) inzident sind; z. B., hier wird Druck als die Ventilatorcharakteristik vorgegeben

$$P_{wi} = P_{AEJ}(Q_j); \quad (3)$$

- die Kanten, die den n_3 Knoten der Atmosphäreanschlüsse inzident sind:

$$P_{wi} = P_{ATM} = const. \quad (4)$$

DNOVP hat insgesamt

$$n = n_1 + n_2 + n_3 \quad (5)$$

Knoten und entsprechend die n Randbedingungen. Die Anfangsbedingungen sind

$$P_j(\xi, 0), Q_j(\xi, 0) \quad (j = 1, 2, \dots, m) \quad (6)$$

Durch die Approximation der Gleichungen (1) nach Linienverfahren mit der Ortsschrittweite $\Delta \xi$ und bei $\beta_{rj}=0$ erhalten wir für v -Element der j -Kante das Simulationsmodell:

$$\begin{cases} \dot{Q}_{jv} = \alpha_j (P_{jv} - P_{jv+1}) - \beta_j Q_{jv} |Q_{jv}| \\ \dot{P}_{jv+1} = g_j (Q_{jv} - Q_{jv+1}) \end{cases} \quad (7)$$

Die Koeffizienten α_j , β_j , g_j sind von aerodynamischen j -Kanteparametern abhängig. Für DNOVP sind $j=1, 2, \dots, m; v=lj/\Delta \xi = Mj$. Wir werden zwischen den Luftstrom(Q)- und Druck(P)-Gleichungen unterscheiden und die folgenden Bezeichnungen einführen:

$$\begin{cases} \dot{Q}_{jv} = F_{jv}; \\ F_{jv} = \alpha_j (P_{jv} - P_{jv+1}) - \beta_j Q_{jv} |Q_{jv}| - \beta_{rj} Q_{jv} |Q_{jv}|; \\ \dot{P}_{jv+1} = D_{jv+1}; \\ D_{jv+1} = g_j (Q_{jv} - Q_{jv+1}) \end{cases} \quad (8)$$

Bei der Anwendung des blockartigen Lösungsverfahrens (BAV) werden wir aus (7) und (8) das diskrete Simulationsmodell (BAV-DSM) bekommen. Fürs Einschnitt-Vierpunkt-Verfahren sehen die BAV-DSM-Gleichungen bei $\beta_{rj}=0$ so aus [3,4]:

$$\begin{aligned} Q_{jv,1} &= \frac{\tau}{720} (251F_{jv,0} + 646F_{jv,1} - 264F_{jv,2} + 106F_{jv,3} - 19F_{jv,4}) + Q_{jv,0}; \\ Q_{jv,2} &= \frac{\tau}{90} (29F_{jv,0} + 124F_{jv,1} + 24F_{jv,2} + 4F_{jv,3} - F_{jv,4}) + Q_{jv,0}; \\ Q_{jv,3} &= \frac{3\tau}{80} (9F_{jv,0} + 34F_{jv,1} + 24F_{jv,2} + 14F_{jv,3} - F_{jv,4}) + Q_{jv,0}; \\ Q_{jv,4} &= \frac{2\tau}{45} (7F_{jv,0} + 32F_{jv,1} + 12F_{jv,2} + 32F_{jv,3} + 7F_{jv,4}) + Q_{jv,0}; \\ F_{jv,0} &= \alpha_j (P_{jv,0} - P_{jv+1,n,0}) - \beta_j Q_{jv,n,0} |Q_{jv,n,0}|; \quad (9) \\ F_{jv,1} &= \alpha_j (P_{jv,0} - P_{jv+1,n,1}) - \beta_j Q_{jv,n,1} |Q_{jv,n,1}|; \\ F_{jv,2} &= \alpha_j (P_{jv,0} - P_{jv+1,n,2}) - \beta_j Q_{jv,n,2} |Q_{jv,n,2}|; \\ F_{jv,3} &= \alpha_j (P_{jv,0} - P_{jv+1,n,3}) - \beta_j Q_{jv,n,3} |Q_{jv,n,3}|; \\ F_{jv,4} &= \alpha_j (P_{jv,0} - P_{jv+1,n,4}) - \beta_j Q_{jv,n,4} |Q_{jv,n,4}| \end{aligned}$$

Die Werte $Q_{v,0}$ und $P_{v+1,n,0}$ im Nullpunkt des Blocks n werden als die Werte im letzten Punkt des vorigen Blocks für $n > 1$ ($t_{n-1,k} = t_{n,0}$) definiert. Für jede Kante des DNOVP werden die Gleichungssysteme (9), (10) mit den entsprechenden Indexe $j = 1, 2, \dots, m$ und $v = 1, 2, \dots, Mj$ generiert. Die Randbedingungen $P_{w1} \dots P_{wn1}$ werden mit den approximierten Knotengleichungen (2) beschrieben.

Entsprechend dem System (10) wird jede Knotengleichung in der BAV-Form umgewandelt und ihre gesuchte Variable P_{wi} ($i=1, 2, \dots, n1$) wird im gesamten Rechenprozess berechnet.

$$\begin{aligned} P_{jv+1,n,1} &= \frac{\tau}{720} (251D_{jv+1,n,0} + 646D_{jv+1,n,1} - 264D_{jv+1,n,2} + 106D_{jv+1,n,3} - 19D_{jv+1,n,4}) + P_{jv+1,n,0}; \\ P_{jv+1,n,2} &= \frac{\tau}{90} (29D_{jv+1,n,0} + 124D_{jv+1,n,1} + 24D_{jv+1,n,2} + 4D_{jv+1,n,3} - D_{jv+1,n,4}) + P_{jv+1,n,0}; \\ P_{jv+1,n,3} &= \frac{3\tau}{80} (9D_{jv+1,n,0} + 34D_{jv+1,n,1} + 24D_{jv+1,n,2} + 14D_{jv+1,n,3} - D_{jv+1,n,4}) + P_{jv+1,n,0}; \\ P_{jv+1,n,4} &= \frac{2\tau}{45} (7D_{jv+1,n,0} + 32D_{jv+1,n,1} + 12D_{jv+1,n,2} + 32D_{jv+1,n,3} + 7D_{jv+1,n,4}) + P_{jv+1,n,0}. \end{aligned}$$

$$D_{jv+1,n,0} = g_j (Q_{jv,n,0} - Q_{jv+1,n,0}); \quad (10)$$

$$D_{jv+1,n,1} = g_j (Q_{jv,n,1} - Q_{jv+1,n,1});$$

$$D_{jv+1,n,2} = g_j (Q_{jv,n,2} - Q_{jv+1,n,2});$$

$$D_{jv+1,n,4} = g_j (Q_{jv,n,4} - Q_{jv+1,n,4}).$$

In [1, 2] wurden vier möglichen Parallelitätsebene (PE) bei der Entwicklung der DNOVP-MIMD-Simulatoren betrachtet, ohne Problem der ebenenrelevanten gleichmäßigen Lastverteilung mit der BAV-Anwendung zu behandeln. Hinsichtlich der Komplexität der industriellen DNOVP und der technischen Möglichkeiten der Zuordnungen von ihren Simulationsmodellen auf vorhandenen parallelen Zielrechnersysteme finden wir aktuell, die Fragen der gleichmäßigen Auslastung der Prozesse/Prozessoren sowie Entwicklung der parallelen BAV-orientierten Algorithmen bezüglich der perspektiven PE-3 „eine Kante – ein Prozess“ zu beantworten.

2. Struktur virtueller QP-Prozesse mit der Zuordnung „Kantengleichungssystem – Prozess“. Die j -Kante kann in DNOVP verschiedene funktionelle Bedeutung haben und mit den Gleichungen beschrieben werden, die nach einigen Gliedern voneinander sich unterscheiden. Bei weiteren Untersuchungen werden wir auf den Simulationsmodellen (7) uns konzentrieren. Hier werden folgenden Bezeichnungen eingeführt: M_j – Menge der Kantenelemente, $j=1, 2, \dots, m$ (m – Kantenumenge im DNOVP); $v = 1, 2, \dots, M_j$ – laufende Nummerierung von Kantenelementen; $Q_{j1}, Q_{j2}, \dots, Q_{jv-1}, Q_{jv}, Q_{jv+1}, \dots, Q_{jMj}$ – laufende Nummerierung von Luftströmen in den Kantenelementen; $P_{j1}, P_{j2}, \dots, P_{jv-1}, P_{jv}, P_{jv+1}, \dots, P_{jMj}$ – laufende Nummerierung von Druckwerten in den Kantenelementen; dabei ist P_{j1} Druck am Anfang und P_{jMj+1} – Druck am

Ende der j- Kante. Dabei entsprechen die Druckdifferenzen

$$(P_{j1}-P_{j2}), (P_{j2}-P_{j3}), \dots, (P_{jv-1}-P_{jv}), (P_{jv}-P_{jv+1}), \\ (P_{jv+1}-P_{jv+2}), \dots, (P_{jMj}-P_{jMj+1})$$

den Luftströme

$$Q_{j1}, Q_{j2}, \dots, Q_{jv-1}, Q_{jv}, Q_{jv+1}, \dots, Q_{jMj}$$

Die Auslastung von Prozessen L ist von der Menge der im Prozess gelösten Gleichungen abhängig und beträgt pro Kante

$$L_j = 2M_j = \frac{2l_j}{\Delta\xi} \quad (11)$$

Gleichungen. Wegen unterschiedlicher Länge der Kanten wird sich die Auslastung in dem Wertebereich $L_{min} \leq L_j \leq L_{max}$ ändern. Hier sind L_{min} , L_{max} die Auslastungen bezüglich der Kanten der minimalen und maximalen Längen. Die Ungleichmäßigkeit beträgt $\Delta L = L_{max} - L_{min} = 2(M_{jmax} - M_{jmin})$ und kann aus der Kodierungstabelle der DNOVP-Parameter (die Spalte l_j) berechnet werden. Nach der Ausführungszeit T_{QP} des QP-Prozesses kann man die maximale Wartezeit T_{Wmax} für DNOVP $T_{Wmax} = \Delta L \cdot T_{QP}$ und die Wartezeiten T_{Wj} für die Prozesse aller Kanten bezüglich des Prozesses der l_{max} -Kante $T_{Wj} = \Delta L_j \cdot T_{QP}$ abschätzen. Hier ist $\Delta L_j = L_{max} - L_j$. Die Kantenlänge des DNOVP haben die Abweichungen zueinander in mehrere Male, deshalb wird die Ungleichmäßigkeit der Prozessauslastungen wesentlich groß sein. Es wird den Ansatz zur gleichmäßigen Auslastung durch eine Modifikation der Parallelitätsebene PE-3 als „ l_{min} -Kante – 1 Prozess“ vorgeschlagen. Bei den weiteren Betrachtungen werden wir die PE-3-Versionen als **PE-3-1** und **PE-3-2** bezeichnen. In Abb.1a wird j-Kante des Testnetzes (Abb.2) mit dem Anfangsknoten „Atmosphäre“ und dem Endknoten $w1(u1)$ dargestellt. Die Kante hat $M_j = l_j / \Delta\xi$ Elemente mit den obigen Nummerierung von gesuchten Variablen $Q_{j1}, \dots, Q_{jMj-1}, Q_{jMj}$ und $P_{j1}, P_{j2}, \dots, P_{jMj}, P_{jMj+1}$. Dabei sind die Randwerte $P_{j1} = P_A, P_{jMj+1} = P_{u1} = P_{w1}$.

Die Parameter des Netzobjektes (Abb. 2) sind in der Tabelle 1 zusammengefasst. In der Version PE-3-1 „1 Kante – 1 BAV-Prozess“ (Abb.1-b) besteht allgemeiner j-Kante-Prozess aus der $(QP)_j$ - und $(QP)_{jw}$ -BAV-Prozesse. $(QP)_j$ -BAV-Prozess realisiert die Kette von BAV-Berechnungsformeln von Type (9)-(10) für gegebene j und durchlaufende

$v=1,2,\dots, M_j$ (in Abb.1 $j=1$ für Testnetz Abb. 2). Letztes M_j -Element der j-Kante hat im M_{j-1} -Element berechnetes „Vorrandwert“ P_{jMj} und braucht für die Berechnung der Variable Q_{jMj} den Randwert $P_{jMj+1} = P_{u1} = P_{w1}$, welcher $(QP)_{jw}$ -BAV-Knotenprozess berechnen soll. Für diese Berechnung braucht $(QP)_{jw}$ -BAV-Knotenprozess eine Eingabe von Werten der inzidenten Luftströme (in Abb.1a, 1b wurden Q_{jMj} und Q_{21}, Q_{51}, Q_{61} gezeigt). Auf dieser Etappe der BAV-Löser-Entwicklung ist methodisch sinnvoll, die Knotenprozesse einfügen, um die BAV-Knotengleichungen für ganzes DNOVP zu generieren und zu lösen. Wegen enger Zusammenwirkung des letzten Elements der Kante und des zugehörigen Knotenprozesses bei der Berechnungen von Q_{jMj} und $P_{jMj+1} = P_{jw} = P_{u1}$ ist die Zweckmäßigkeit der Vereinigung beider Prozessen zu untersuchen.

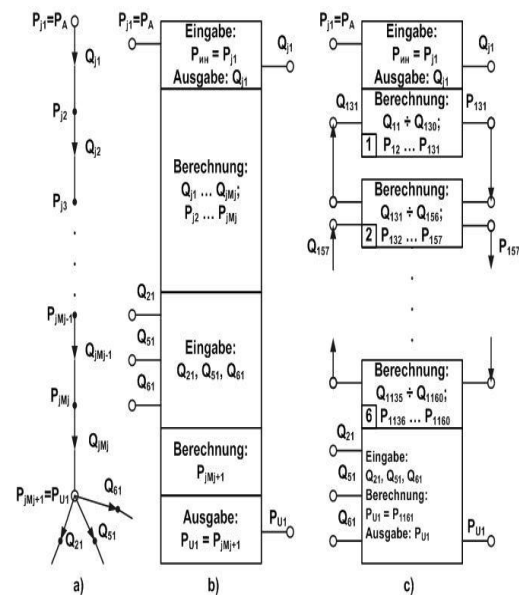


Abbildung 1. Zur Struktur virtueller QP-Prozesse mit der Zuordnung „1 Kante – 1 Prozess“: **a** – j-Kante des DNOVP (Knoten $P_{11}=P_A$ und $U1=W1$ bei $j=1$ aus Testnetz Abb.2); **b** – Struktur der j-kanteangehörigen $(QP)_j$ - und $(QP)_{jw}$ -BAV-Prozesse; **c** – Struktur der j-kanteangehörigen $(QP)_j$ - und $(QP)_{jw}$ -BAV-Prozesse bei der Zuordnung „ l_{min} -Teilkante – Prozess“ (Q_1 -Kante, $j=1$, Testnetz Abb.2 wird in den 6 l_{min} -Teilkanten zerlegt).

Abb.1c zeigt Struktur der BAV-Prozesse für die Version der PE-3-2 mit der Zuordnung „Kante mit minimaler Länge l_{min} – ein BAV-Prozess“. Hier wird j-Kante in $l_j/l_{min} = 6$ Teilkanten zerlegt.

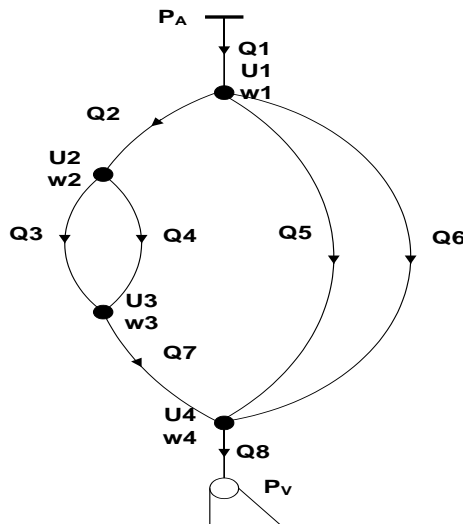


Abbildung 2. Testnetz.

Jede Teilkante (TK) soll eigenen (QP)_{JKTK}-BAV-Prozess haben, der für das Lösen von $M_j/6$ Gleichungssystemen der Elementen nach BAV-Berechnungsformeln zuständig wäre. Dabei soll im letzten Element der 6. Teilkante die Berechnung der Randbedingung realisiert werden. Wenn wir betrachten die j -Kante im DNOVP, dann gehört P_{ji} zum Anfangsknoten und $P_{jM_{j+1}}$ – zum Endknoten der j -Kante. Die auf Abb. 1c gezeigte Struktur der PE-3-2-BAV-Prozesse für j -Kante entsteht als Ergebnis der folgenden Berechnungen:

$l_j=800M$ (Kante Q1, Tab.1) und Ortsschrittweite $\Delta\xi=5M$, $M_j=l_j/5=160$; $l_{min}=150M$ (Kanten Q2, Q7, Tab.1); die Menge der Teilkanten ist $5 < N_{jTK} = l_j/l_{min} < 6$, nehmen wir $N_{jTK} = 6$; ausgehend aus gleichmäßigen Verteilung der Prozessauslastung machen wir die folgende Zuordnung von TK-Elementen zu den BAV-Prozesse:

TK_{10j} – BAV-(QP)_{j1}-Prozess – 30
Elementen 1-30 mit den Variablen Q_{j11}- Q_{j130},
P_{j12}-P_{j131};

TK_{20j} – BAV-(QP)_{j2}-Prozess – 26
Elementen 31-56 mit den Variablen Q_{j231}- Q_{j256},
P_{j232}-P_{j257};

TK_{30j} – BAV-(QP)_{j3}-Prozess – 26
Elementen 57-82 mit den Variablen Q_{j357}- Q_{j382},
P_{j358}-P_{j383};

TK_{40j} – BAV-(QP)_{j4}-Prozess – 26
Elementen 83-108 mit den Variablen Q_{j483}-
Q_{j4108}, P_{j484}-P_{j4109};

TK_{50j} – BAV-(QP)_{j5}-Prozess – 26
Elementen 109-134 mit den Variable Q_{j5109}-
Q_{j5134}, P_{j5110}-P_{j5135};

TK_{60j} –BAV-(QP)_{j6}-Prozess– 26
 Elementen 135-160 mit den Variablen Q_{j6135}-
 Q_{j6160}, P_{j6136}-P_{j6160};

TK_{6wQj} – BAV-(QP)_{jw1}-Prozess – w1-Knotenprozess mit den Eingangsvariablen Q_{j6160} sowie Q₂₁₁, Q₅₁₁, Q₆₁₁ (Abb. 1a, c) und der berechneten Randbedingung P_{j6161}=P_{w1}

Also, die Strukturen an Abb.1 zeigen, dass die Parallelitätsebene PE-3-1 «1 Kante – 1 Prozess» soll m (QP)-BAV-Prozesse haben, die die Gleichungssysteme aller j-Kanten ($j=1,2,...,m$) lösen:

$$\begin{cases} \dot{Q}_{I1} = \alpha_I(P_{I1} - P_{I2}) - \beta_I Q_{I1} |Q_{I1}| \\ \dot{P}_{I2} = g_I(Q_{I1} - Q_{I2}) \end{cases}$$

$$\begin{cases} \dot{Q}_{1Ml} = \alpha_l(P_{1Ml} - P_{1Ml+1}) - \beta_l Q_{1Ml} |Q_{1Ml}| \\ \dot{P}_{1Ml+1} = g_l(Q_{1Ml} - Q_{1Ml+1}) \end{cases}$$

$$j=1, v=1, \dots, M_l; (QP)_l\text{-BAV-Prozess}; \quad (12)$$

$$\begin{cases} \dot{Q}_{m1} = \alpha_m(P_{m1} - P_{m,2}) - \beta_m Q_{m1} |Q_{m1}| \\ \dot{P}_{m,2} = g_m(Q_{m1} - Q_{m,2}) \end{cases}$$

$$\begin{cases} \dot{Q}_{mv} = \alpha_m (P_{mv} - P_{m,v+1}) - \beta_m Q_{mv} |Q_{mv}| \\ \dot{P}_{m,v+1} = g_m (Q_{mv} - Q_{m,v+1}) \end{cases}$$

$$j=m, v=1, \dots, M_m; (QP)_m\text{-BAV-Prozess.}$$

In dem $(QP)_{jwi}$ -BAV-Teilprozess werden dabei die Gleichungen

$$\begin{cases} \dot{Q}_{jMj} = \alpha_j(P_{jMj} - P_{jwi}) - \beta_j Q_{jMj} |Q_{jMj}| \\ \dot{P}_{jwi} = g_{jwi}(Q_{jMj} - \Sigma Q_{mz}) \end{cases} \quad (13)$$

mit den BAV-Berechnungsformeln für letzten Element $v=M_j$ der j-Kante gelöst. Wie Abb.1b zeigt, die algebraische Summe von w_1 -inzenten Luftströme ist $\sum Q_{inz} = Q_{21} + Q_{51} + Q_{61}$ (die Luftströme in den Anfangselementen der zum Knoten $u_1=w_1$ inzidenten Kanten Q_2, Q_5, Q_6). Auf PE-3-2 brauchen wir für jede Kante ganzzahlige Menge $N_{ITK} = l/l_{min}$ und für ganzes

$$\text{DNOVP} \sum_{j=1}^m N_{jTK} \quad \text{BAV-Teilkanten-Prozesse.}$$

Alle DNOVP-Kanten sind zu den \mathbf{n}_I inneren Knoten inzident, wo sollen die Knotendruckwerte während des Lösen des DNOVP-Gleichungssystems entsprechend der dynamischen Knotenbedingungen (2) und (13) berechnet werden. Also, beide PE-3-Versionen brauchen die $\mathbf{n}_I(QP)_{jwi}$ -BAV-Teilprozesse ($i =$

$1, 2, \dots, n_1$) für das Lösen von n_1 Gleichungssystemen (13). Dabei besteht allgemein Summe ($Q_{jMj} - \sum Q_{inz}$) aus zufließenden, in j-Kante berechneten Luftstrom

Q_{jMj} und aus in der anderen Kanten berechneten zufließenden und ausfließenden Luftströme, die zur j_{wi} -Knoten inzident sind.

Tabelle 1

\Par Kante\	Lj	Fj	rj	$\Delta \xi_m$ Δx	Mj	α_j	β_j	γ_j	Test PA	Test PV	Test Q
Q1	800	6,5	0,000484	40	20	0,13	0,00252	523,6	0	0	71,6
Q2	150	6,5	0,000484	40	4	0,13	0,00252	523,6	0	0	28,9
Q7	150	6,5	0,000484	40	4	0,13	0,00252	523,6	0	0	28,9
Q8	650	6,5	0,000484	40	16	0,13	0,00252	523,6	0	-4100	71,6
Q3	1925	4,7	0,00105	40	48	0,09	0,004	728,6	0	0	14,4
Q4	1410	6	0,0019	40	35	0,12	0,009	569,1	0	0	14,9
Q5	1720	4,5	0,00112	40	43	0,09	0,004	756,2	0	0	19,2
Q6	934	5	0,00137	40	23	0,1	0,0055	684,7	0	0	23,7

3. Die Operationen von BAV-Prozessen auf PE-3-1 und PE-3-2. Aus Abb.1 und der zu lösenden Gleichungen (12), (13) folgt, dass vereinigte (QP)_j-BAV-Prozesse die folgenden Operationen auf PE-3-1 erfüllen sollen:

- 1) Eingabe der Druckwerte $P_{jwi} = P_{jl}$ in den Anfangsknoten der jeweiligen j-Kanten;
- 2) Berechnung der Q_{jv} - und P_{jv+1} -Werte für v-Elementen ($v=1,2,\dots,M_j$) der j-Kanten;
- 3) Berechnung der Knotendruckwerte $P_{jMj+1}=P_{jwi}$ für die Knoten, die zur letzten Elemente der j-Kanten inzident sind, entsprechend den Gleichungssysteme (13);
- 4) Eingabe der Variablen Q_{inz} , die fürs Lösen von Systemen (13) notwendig sind;
- 5) Ausgabe der berechneten Q_{jMj} - und ($P_{jMj+1}=P_{jwi}$)-Werte den BAV-Prozesse, die den zur j-Kante-Endknoten inzidenten weiteren Kanten entsprechen und die Q_{inz} -Ströme berechnen.

Auf der PE-3-2 wird jede j-Kante nach der Approximation mit der Ortschaftweite $\Delta \xi$ und Verteilung auf $l_j/l_{min} = N_{jTK}$ Teilkanten (TK) die folgenden gesuchten Variablen beinhalten:

Hier sind P_{winzAK} und P_{winzEK} die Druckwerte entsprechend in den für j-Kante inzidenten Anfangs- und Endknoten ($winzAK$, $winzEK$). Die j-Kante wird in eine Kette von N_{jTK} Teilkanten zerlegt, für welchen die in den vorigen Nachbarsteilkanten berechneten Druckwerte als die Randbedingungen dienen. Also, auf der PE-3-2 sind drei folgenden Arten

von Teilkanten (TK) und zugeordneten TK-(QP)-BAV-Prozesse zu unterscheiden.

$Q_{j11}, Q_{j12}, \dots, Q_{j1MjTK}; P_{j11}, P_{j12}, \dots, P_{j1MjTK}, P_{j1MjTK+1}$ – in 1. Teilkante, $P_{j11}=P_{winzAK}$;

$Q_{j21}, Q_{j22}, \dots, Q_{j2MjTK}; P_{j21}, P_{j22}, \dots, P_{j2MjTK}, P_{j2MjTK+1}$ – in 2. Teilkante, $P_{j21}=P_{j1MjTK+1}$;

(14)

$Q_{jNjTK-1,1}, Q_{jNjTK-1,2}, \dots, Q_{jNjTK-1,MjTK}; P_{jNjTK-1,1}, P_{jNjTK-1,2}, \dots, P_{jNjTK-1,MjTK}, P_{jNjTK-1,MjTK+1}$ – in der vorletzten Teilkante $N_{jTK}-1$, dabei ist $P_{jNjTK-1,1}=P_{jNjTK-2,MjTK+1}$;

$Q_{jNjTK1}, Q_{jNjTK2}, \dots, Q_{jNjTK,MjTK}; P_{jNjTK,1}, P_{jNjTK,2}, \dots, P_{jNjTK,MjTK}, P_{jNjTK,MjTK+1}$ – in der letzte Teilkante N_{jTK} , dabei sind $P_{jNjTK,1}=P_{jNjTK-1,MjTK+1}$ und $P_{jNjTK,MjTK+1} = P_{winzEK}$.

Teilkanten TK1. Teilkanten, die den Anfangsknoten (AK) wi der j-Kanten inzident sind (AK_i in der Kodierungstabelle des DNOVP-Graphen, $1 \leq i \leq n_1$) und haben die gesuchten Variablen

$$Q_{j11}, Q_{j12}, \dots, Q_{j1MjTK}; \quad (13)$$

$$P_{j11}, P_{j12}, \dots, P_{j1MjTK}, P_{j1MjTK+1} \quad (j=1, 2, \dots, m)$$

Die **TK1-(QP)-Prozesse** sollen die folgenden Operationen erfüllen:

- 1) Eingabe der Druckwerte $P_{jwi} = P_{jl}$ in den Anfangsknoten der jeweiligen j-Kanten (ähnlich wie auf PE-3-1);

- 2) Berechnung der Q_{jlv} - und P_{jlv+1} -Werte für v-Elementen im Bereich der Teilkanten TK1_j, welche die Länge $l_{min} + \Delta l_j$ haben. Hier ist Δl_j eine Längendifferenz, die bei der

ganzzahligen Zerlegung von j-Kanten nach N_{jTK} l_{min} -Teile entstehen kann: $\Delta l_j = l_j - l_{min} * N_{jTK}$. Die Approximation der j-Teilkanten nach Linienverfahren mit der Ortsschrittweite $\Delta \xi$ gibt die Menge der Elemente in Teilkanten $M_{jTK} = (l_{min} + \Delta l_j) / \Delta \xi$. Deshalb erfolgen die Berechnungen für $v=1,2,...,M_{jTK}$.

3) Berechnung der Randbedingungen für die erste Elementen von Teilkanten **TK2**, die nach **TK1** folgen. Diese Operationen sollen entsprechend den Gleichungen von Type (13) in den Elemente $v = M_{jTK}$ erfüllt werden. Das Gleichungssystem (13) für das Grenzelement der j-Teilkante **TK1** wird so aussehen (Abb.1, a, c):

$$\begin{cases} \dot{Q}_{j1M_{jTK}} = \alpha_j (P_{j1M_{jTK}} - P_{j21}) - \beta_j Q_{j1M_{jTK}} |Q_{j1M_{jTK}}| \\ \dot{P}_{j21} = g_j (Q_{j1M_{jTK}} - Q_{j21}) \end{cases} \quad (13')$$

Das Element $v=M_{jTK}+1$ in (13') gehört der Nachbarsteilkante **TK2**, der Druck $P_{jM_{jTK}+1} = P_{j21}$ ist die Randbedingung für diese Teilkante.

4) Eingabe der Variablen $Q_{jM_{jTK}+1} = Q_{j21}$ aus BAV-Prozesse von Teilkanten **TK2**;

5) Ausgabe der berechneten Variablen P_{j21} als Randbedingungen für die ersten Elemente der zugehörigen Teilkanten **TK2**.

Die BAV-(QP)-Prozesse erfüllen alle obigen Operationen mit der Nummerierung der gesuchten Variablen entsprechend der Bezeichnungen (14).

Teilkanten TK2. Teilkanten, die mit den **TK1** verbunden sind und die folgenden gesuchten Variablen haben:

$$\begin{aligned} &Q_{j21}, Q_{j22}, \dots, Q_{j2M_{jTK}}; P_{j21}, P_{j22}, \dots, P_{j2M_{jTK}}, \\ &P_{j2M_{jTK}+1} (P_{j21} = P_{j1M_{jTK}+1}); \\ &Q_{j31}, Q_{j32}, \dots, Q_{j3M_{jTK}}; P_{j31}, P_{j32}, \dots, P_{j3M_{jTK}}, \\ &P_{j3M_{jTK}+1} (P_{j31} = P_{j2M_{jTK}+1}); \end{aligned} \quad (14')$$

$$\begin{aligned} &Q_{j_{N_{jTK}-1,1}}, Q_{j_{N_{jTK}-1,2}}, \dots, Q_{j_{N_{jTK}-1,M_{jTK}}}; P_{j_{N_{jTK}-1,1}}, P_{j_{N_{jTK}-1,2}}, \dots, \\ &P_{j_{N_{jTK}-1,M_{jTK}}}, P_{j_{N_{jTK}-1,M_{jTK}+1}} (P_{j_{N_{jTK}-1,1}} = P_{j_{N_{jTK}-2,M_{jTK}+1}}). \end{aligned}$$

In Abb.1c ist beispielweise $N_{jTK} = 6$ – die Menge der Teilkanten, auf welchen die j-Kante zerlegt wurde. Die **TK2**-BAV-(QP)-Prozesse sollen die folgenden Operationen erfüllen:

1) Eingabe der Werte $P_{j1M_{jTK}+1} = P_{j21}$ aus **TK1** als die Randbedingungen für die ersten Elemente der ersten Teilkanten **TK2**;

2) Berechnung der Q_{jv} - und P_{jv+1} -Werte für v-Elementen im Bereich der Teilkanten **TK2**. Wie oben in **TK1**, erfolgen die Berechnungen für Indexe $v=1,2,...,M_{jTK}$. Dabei werden die Randbedingungsdruckwerten

$$P_{j2M_{jTK}+1} = P_{j31}, P_{j3M_{jTK}+1} = P_{j41}, \dots,$$

$$P_{j_{N_{jTK}-2,M_{jTK}+1}} = P_{j_{N_{jTK}-1,1}},$$

$$P_{j_{N_{jTK}-1,M_{jTK}+1}} = P_{j_{N_{jTK},1}}$$

für die ersten Elementen von nachfolgenden Teilkanten nach ähnliche wie (13') Gleichungen berechnet. Die Teilkante $j_{N_{jTK}}$ ist letzte in der j-Kante, deshalb gehört zur **TK3**. Berechnung der Randbedingungen $P_{j_{N_{jTK}-1,M_{jTK}+1}} = P_{j_{N_{jTK},1}}$ für die ersten Elementen von Teilkanten **TK3** soll in den Grenzelementen $v = M_{jTK}$ der letzten Teilkanten **TK2** entsprechend (13'') erfüllt werden:

$$\begin{cases} \dot{Q}_{j_{N_{jTK}-1,M_{jTK}}} = \alpha_j (P_{j_{N_{jTK}-1,M_{jTK}}} - P_{j_{N_{jTK},1}}) - \beta_j Q_{j_{N_{jTK}-1,M_{jTK}}} |Q_{j_{N_{jTK}-1,M_{jTK}}}| \\ \dot{P}_{j_{N_{jTK},1}} = g_j (Q_{j_{N_{jTK}-1,M_{jTK}}} - Q_{j_{N_{jTK},1}}) \end{cases} \quad (13'')$$

3) Eingabe der obigen in vorigen Teilkantenprozesse berechneten Druckwerten

$$P_{j2M_{jTK}+1} = P_{j31}, P_{j3M_{jTK}+1} = P_{j41}, \dots,$$

$$P_{j_{N_{jTK}-2,M_{jTK}+1}} = P_{j_{N_{jTK}-1,1}},$$

$P_{j_{N_{jTK}-1,M_{jTK}+1}} = P_{j_{N_{jTK},1}}$, die für die ersten Elemente der nachfolgenden Teilkanten $j_{31}, j_{41}, \dots, j_{N_{jTK}-1,1}, j_{N_{jTK},1}$ als die Randbedingungen sind;

4) Eingabe der Variablen $Q_{j_{N_{jTK},1}}$ aus den BAV-Prozesse von Teilkanten **TK3** für die Berechnungen nach (13'');

5) Ausgabe der berechneten Variablen $P_{j_{N_{jTK},1}}$ als Randbedingungen für die ersten Elemente der Teilkanten **TK3**.

Die BAV-(QP)-Prozesse erfüllen alle obigen Operationen mit der Nummerierung der gesuchten Variablen entsprechend der Bezeichnungen (14').

Teilkanten TK3. Teilkanten, die nach den **TK2** folgen, als letzte in den j-Kanten sind und die folgenden gesuchten Variablen haben:

$$\begin{aligned} &Q_{j_{N_{jTK},1}}, Q_{j_{N_{jTK},2}}, \dots, Q_{j_{N_{jTK},M_{jTK}}}; P_{j_{N_{jTK},1}}, P_{j_{N_{jTK},2}}, \dots, \\ &P_{j_{N_{jTK},M_{jTK}}}, P_{j_{N_{jTK},M_{jTK}+1}} \end{aligned}$$

$$(P_{j_{N_{jTK},1}} = P_{j_{N_{jTK}-1,M_{jTK}+1}}). \quad (14'')$$

Die Druckwerte $P_{j_{N_{jTK},M_{jTK}+1}}$ fallen mit den Druckwerten in den Knoten, die zu den j-Kanten inzident sind, zusammen. Die **TK3**-BAV-(QP)-Prozesse sollen die folgenden Operationen erfüllen:

1) Eingabe der in **TK2** berechneten Druckwerte $P_{j_{N_{jTK},1}}$ als Randbedingungen für die ersten Elemente der Teilkanten **TK3**;

2) Berechnung der Q_{jv} - und P_{jv+1} -Werte für v -Elementen im Bereich der Teilkanten **TK3**. Wie oben in **TK2**, erfolgen die Berechnungen für Indexe $v=1,2,...,M_{jTK}$, $j=1,2,...,m$.

3) Berechnung der Variablen $P_{jNjTK,MjTK+1} = P_{jwi}$ in den letzten Elementen von Teilkanten als inzidenten Knotendruckwerte mit der Berücksichtigung der Luftströme in den inzidenten Kanten. Diesen Berechnungen sollen in den Grenzelementen $v = M_{jTK}$ der Teilkanten **TK3** entsprechend der (13) erfüllt werden. Das Gleichungssystem (13) wird für das Grenzelement der Teilkante **TK3** so aussehen:

$$\begin{cases} \dot{Q}_{jNjTK,MjTK} = \alpha_j(P_{jNjTK,MjTK} - P_{jwi}) - \beta_j Q_{jNjTK,MjTK} |Q_{jNjTK,MjTK}| & (13'') \\ \dot{P}_{jwi} = g_j(Q_{jNjTK,MjTK} - \Sigma Q_{inz}) \end{cases}$$

4) Eingabe der Variablen Q_{inz} aus der BAV-Prozesse von zur wi -Knoten inzidenten Endteilkanten **TK3** für die Berechnungen nach (13'');

5) Ausgabe der berechneten Variablen P_{jwi} als Randbedingungen für die ersten Elemente der wi -inzidenten Teilkanten.

Die **TK3**-BAV-(QP)-Prozesse erfüllen alle obigen Operationen mit der Nummerierung der gesuchten Variablen nach (14').

4. Entwicklung des parallelen BAV-Algorithmus für den DNOVP-Löser auf PE-3-2.

Die Ungleichmäßigkeit der Prozessauslastungen in PE-3-1 ist wesentlich groß. Aus dieser Tatsache wird in der Parallelitätsebene PE-3-2 folgender Algorithmus der gleichmäßigen Lastverteilung vorgeschlagen:

1. Aus der DNOVP-Kodierungstabelle (die Spalten Q_j , l_j) wird eine Kante mit minimaler Länge l_{jmin} ausgewählt. Allgemein können mehrere solche Kanten vorhanden sein.

2. Es werden die Prozessmenge für alle Kanten unter der Bedingung „ l_{jmin} -Kante – 1Prozess“ (mit der Rundung bis Ganzzahlen) berechnet:

$$N_{pj} = l_j / l_{min}$$

3. Es werden die Elementenmenge M_j in den Kanten bei der vorgegebenen Schrittweite $\Delta\zeta$ der Diskretisierung nach Ortskoordinate berechnet:

$$M_j = l_j / \Delta\zeta.$$

4) Es werden die Elementenmenge M_{pj} der j -Kante, die mit den N_{pj} Prozesse „bedeckt

werden“ und die durch die Rundung N_{pj} verursachte Elementendifferenz ΔM_j gesucht:

$$M_{pj} = M_{min} N_{pj} \quad (M_{min} = l_{min} / \Delta\zeta),$$

$$\Delta M_j = M_j - M_{pj}.$$

5) Es wird ein Parameter „Last“ L_{pj} des Prozesses von j -Kante als die Elementenmenge pro Prozess eingeführt:

$$L_{pj} = \frac{M_{pj}}{N_{pj}} = M_{min}$$

6) Die Elementendifferenz ΔM_j wird zwischen den Prozesse der j -Kante verteilt, ergibt sich damit ein Last

$$L_j = \frac{M_{pj} + \Delta M_j}{N_{pj}} = \frac{M_j}{N_{pj}}$$

7) Um alle Kantenelemente in den Prozessen der j -Kante zu berücksichtigen, wird in den Kanten mit $\Delta M_j > 0$ ein zusätzlicher Prozess eingefügt. Dann werden zwei Varianten der Lastverteilung analysiert:

7.1. Für alle N_{pj} Prozesse wird ein Last $L_{pj} = M_{min}$ zugeordnet. Die übrige ΔM_j Elemente werden dem zusätzlichen Prozess zugeordnet, d. h. für j -Kante stehen $N_{pj}+1$ Prozesse zur Verfügung. In diesem Variant werden alle Prozesse N_{pj} in den Kanten $j = 1...m$ gleichmäßig ausgelastet; die zusätzliche Prozesse werden nicht voll ausgelastet sein, ihre Wartezeiten sind von der Differenzen ΔM_j abhängig.

7.2. Alle $N_{pj}+1$ Prozesse der j -Kante werden so ausgelastet, dass wenigstens ein Prozess die „nominale“ Auslastung M_{min} hätte; dann werden alle andere Prozesse gleichmäßige, kleinere als M_{min} Auslastung haben.

8) Die obige Operationen werden für $j = 1,2,...,m$ im Vorbereitungsmodul des Simulators erfüllt und Ergebnisse in der Lastbalancierungstabelle (TABLAST) zusammengefasst.

Struktur des Algorithmus wird in Abbildung 3 dargestellt.

Kurze Beschreibung des Algorithmus:

1. Es wird HOST-Prozess eingeführt, der die parallel erfüllten BAV-(QP) $_j\omega$ -Prozesse für $1 \leq j \leq m$, $1 \leq \omega \leq N_{jTK}$ (mit den inneren Kantenelementen $1 \leq v \leq M_{jTK}$) verwaltet. Vorbereitungsmodul VM-1 wird durch Realisation des obigen Algorithmus der Lastverteilung und TABLAST-Erstellung

erweitert. Nach TABLAST-Information wird im Block 2 eine Definition der Struktur von DNOVP-BAV-Prozessen erledigt: jedem j -Kante-Prozess werden entsprechende $j\omega$ -Teilkante-Prozesse zugeordnet.

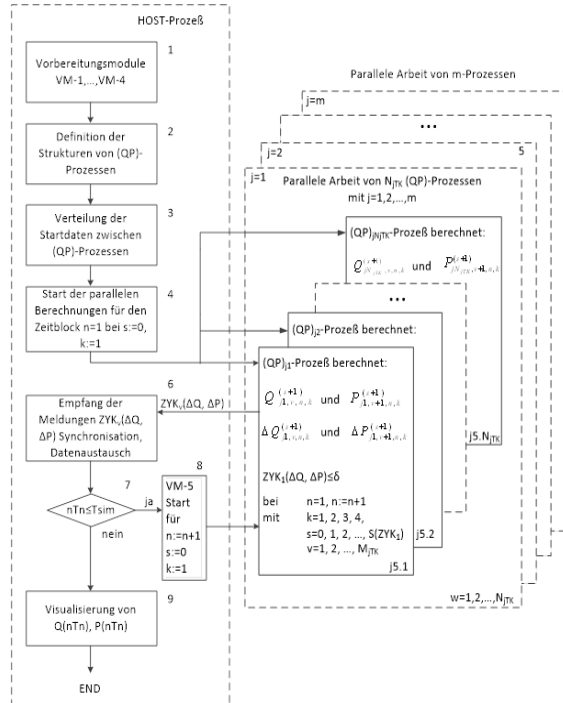


Abbildung 3. Struktur des parallelen BAV-Algorithmus für den PE-3-2-Simulator

2. Nach dem Start von Block 4 realisiert jeder $j\omega$ -Teilkante-Prozess den sequentiellen Algorithmus fürs Lösen der den Teilkanten entsprechenden BAV-Gleichungsketten. Block 5 der Abb.3 wird in m j Kanten-Blöcke parallelisiert: für den Zeitschritt n werden alle $j\omega$ -Teilkante-(QP)-Prozesse gleichzeitig die iterativen Berechnungen der zugeordneten BAV-Variablen $Q_{jv,n,k}^{(s+1)}$ nach (9) mit $k = 1, 2, 3, 4$ und $s = 0, 1, 2, \dots, S(ZYK)$ durchführen, bis die vorgegebene Genauigkeit δ erreicht wird. Hier werden auch für den Zeitschritt n alle P_{jv+1} sowie P_w -Prozesse gleichzeitig die iterativen Berechnungen der zugeordneten BAV-Variablen $P_{jv+1,n,k}^{(s+1)}$ nach (10) und $P_{wi,n,k}^{(s+1)}$ nach BAV-Form von (13) mit $k = 1, 2, 3, 4$ und $s = 0, 1, 2, \dots, S(ZYK)$ durchführen, bis die vorgegebene Genauigkeit δ erreicht wird.

3. Im Block 6 prüft HOST-Prozess die Meldungen über Variablen $ZYK_{jv}(\Delta Q, \Delta P)$, synchronisiert die Prozesse und fixiert Ende des n -Zeitschrittes.

4. Dann prüft HOST Bedingung $nT_n \leq T_{sim}$ und erlaubt Start der parallelen Prozesse für nächsten Zeitschritt $n:=n+1$.

5. Die Visualisierung der Simulationsergebnisse erfolgt nach Angaben der Prozesse über gespeicherte Datenfile $Q_j\omega_v(nT)$, $P_j\omega_{v+1}(nT)$ mit $1 \leq j \leq m$, $1 \leq \omega \leq N_{jTK}$ und $1 \leq v \leq M_{jTK}$.

PE-3-1 braucht folgende Mengen von Prozessen:

$$N_{PE-3-1} = m + n1 + n2 + n3 \quad (16)$$

Auf PE-3-2 wird jede DNOVP-Kante in N_{jTK} l_{min} -Teilkanten zerlegt, deshalb braucht dieser Ansatz insgesamt

$$N_{PE-3-2} = \sum_{j=1}^m N_{jTK} + n1 + n2 + n3 \quad (17)$$

Prozesse. Die Zuordnung auf parallele virtuelle Systeme sieht das Verhältnis „1Prozess – 1Prozessor“ als üblich aus und ist realisierbar.

5. Zusammenfassung und Ausblick. Es wurden die diskrete virtuelle parallele Simulationsmodelle (DVPSM) der dynamischen Netzobjekten mit verteilten Parametern (DNOVP) und detaillierte Entwicklungen der BAV-Lösern für die **Parallelitätsebene 3** aufgrund der vorigen Implementierungserfahrung der sequentiellen BAV-Löser betrachtet. Dabei sind folgende für die Implementierung wichtige Entwicklungsaspekte bearbeitet: Struktur der virtuellen (QP)-Prozesse mit der Zuordnung „1 Kante – 1 Prozess“, „1 Teilkante – 1 Prozess“; Die BAV-Darstellung von Rechenaufgaben für (QP)-Prozesse der j -Kante und des DNOVP; Entwicklung des parallelen BAV-Algorithmus für den j -Kante-Löser auf PE-3-2; Entwicklung des parallelen BAV-Algorithmus für den DNOVP-Löser auf PE-3-1 und PE-3-2.

Die weiteren Arbeiten in Kooperation mit dem HLRS der Universität Stuttgart werden nach Lösen folgenden Aufgaben konzentriert: Implementierung der parallelen BAV-Simulatoren für j -Kante und DNOVP auf PE-3 aufgrund der erhaltenen Erfahrung; Effizienzanalyse der parallelen BAV-Simulatoren; Anwendung der obigen Ergebnisse im Programm der Entwicklung und Implementierung des ukrainischen Forschungs- und Schulungszentrums der parallelen Simulationstechnik.

Literaturverzeichnis

- [1] Gogolenko S.Y., Svjatnyj V.A. Architecture aware parallelization of solvers for PDE systems on geometrical graphs. In: Computer science – Research and Development. – 2009, Vol. 23, no. 3-4, P.225-230, - ISSN 1865-2034.
- [2] Svyatnyy V.A., Kushnarenko V.G., Resch M., Miroshkin O.M., Wesner S.: Problematik der parallelen Simulationstechnik. Наукові праці ДонНТУ №2 (23), 2016 Серія "Інформатика, кібернетика та обчислювальна техніка", ISSN 1996-1588, Покровськ, 2016, с.5-20.
- [3] Фельдман Л. П. Параллельные алгоритмы моделирования динамических систем, описываемых обыкновенными дифференциальными уравнениями. // Электронное моделирование, том 26, № 1, 2004.- С. 19-30.
- [4] Фельдман Л. П., Назарова И. А. Параллельные алгоритмы численного решения задачи Коши для систем обыкновенных дифференциальных уравнений. // Математическое моделирование, том 18, №6, 2006.- С.17-31.
- [5] Kushnarenko, V., Resch, M., Svjatnyj, V., Wesner, S.: Zur Entwicklung des Gleichungslösersubsystems der verteilten parallelen Simulationsumgebung.ASIM'2014 in Berlin, Tagungsband 2, ARGESIM Report 43, Wien 2014, S. 357-363.

References

- [1] Gogolenko S.Y., Svjatnyj V.A. Architecture aware parallelization of solvers for PDE systems on geometrical graphs. In: Computer science – Research and Development. – 2009, Vol. 23, no. 3-4, pp.225-230, - ISSN 1865-2034.
- [2] Svyatnyy V.A., Kushnarenko V.G., Resch M., Miroshkin O.M., Wesner S.: Problematik der parallelen Simulationstechnik. Scientific works of DonNTU №2 (23), 2016 Series "Informatics, Cybernetics and Computer Engineering", ISSN 1996-1588, Pokrovsk, 2016, pp.5-20.
- [3] Feldman L. P. Parallel dynamic modeling algorithms systems described by ordinary differential equations. // Electronic modeling, vol. 26, № 1, 2004.- pp. 19-30.
- [4] Feldman L. P., Nazarova Y. A. Parallel algorithms for the numerical solution of the Cauchy problem for systems of ordinary differential equations. // Mathematical modeling, vol. 18, №6, 2006.- pp.17-31.
- [5] Kushnarenko, V., Resch, M., Svjatnyj, V., Wesner, S.: Zur Entwicklung des Gleichungslösersubsystems der verteilten parallelen Simulationsumgebung.ASIM'2014 in Berlin, Tagungsband 2, ARGESIM Report 43, Wien 2014, pp. 357-363.

Надійшла до редакції 10.01.2020 р.

Svyatnyy Volodymyr Andriiovych – Doctor of technical sciences, Professor, Head of Computer Engineering Department, SHEE "Donetsk National Technical University" (Shibankova sq., 2, Pokrovsk, 85302, Ukraine).

E-mail: vsvjatnyj@gmail.com

Kushnarenko Volodymyr Hennadiiovych – Academic employee, Institute of Information Resource Management, Ulm University (Albert-Einstein-Allee, 43, 89081 Ulm, Germany).

E-mail: volodymyr.kushnarenko@uni-ulm.de

ПІДХІД ДО РІВНОМІРНОГО РОЗПОДІЛУ НАВАНТАЖЕННЯ МІЖ ПРОЦЕСОРАМИ MIMD-СИМУЛЯТОРА МЕРЕЖЕВИХ ДИНАМІЧНИХ ОБ'ЄКТІВ З РОЗПОДІЛЕНИМИ ПАРАМЕТРАМИ

Мережеві динамічні об'єкти з розподіленими параметрами (МДОРП) широко представлені в різних предметних областях і є об'єктами досліджень, проектування, автоматизації, спостереження й контролю, забезпечення якісних показників, оптимального ведення процесів, аналізу й прогнозу станів безпеки, запобігання аварійним ситуаціям тощо. Методи та засоби моделювання цього класу об'єктів мають суттєве теоретичне й практичне значення як на стадії проектування, так і в процесі експлуатації. При розробці паралельних МДОРП-MIMD-симуляторів має розв'язуватись актуальна задача рівномірного розподілу

завантаження процесорів, що задіяні в вирішенні системи рівнянь об'єкта. В статті розглядається можливий рівень розпаралелювання моделі МДОРП „Гілка графу мережі – MIMD-процес/процесор“ з позиції розподілу завантаження обчислювальних ресурсів і пропонується підхід до його балансування. Ідея вирівнювання завантажень полягає в підпорядкуванні типу „Гілка мінімальної довжини – один процес/процесор“ і формалізації МДОРП-Simulation-моделі відповідно до вторинної топології графу мережі, яка є наслідком цього підпорядкування та застосування блокового обчислювального методу (БОМ).

Ключові слова: мережеві динамічні об'єкти з розподіленими параметрами (МДОРП); віртуальні рівні розпаралелювання; розподіл завантаження; паралельний MIMD-симулятор; блоковий обчислювальний метод (БОМ).

Святний Володимир Андрійович – доктор технічних наук, професор, завідувач кафедри, кафедра комп'ютерної інженерії, ДВНЗ “Донецький національний технічний університет” (пл. Шибанкова, 2, Покровськ, 85302, Україна).

E-mail: vsvjatnyj@gmail.com

Кушнарєнко Володимир Геннадійович – науковий співробітник, Інститут організації та управління інформаційними системами, Університет Ульма (Алея Альберта Ейнштейна, 43, 89081 Ульм, Німеччина).

E-mail: volodymyr.kushnarenko@uni-ulm.de

AN APPROACH TO EVEN LOAD DISTRIBUTION BETWEEN PROCESSORS OF THE MIMD SIMULATOR OF DYNAMIC NETWORK OBJECTS WITH DISTRIBUTED PARAMETERS

Dynamic network objects with distributed parameters (DNODPs) are widely represented in various subject areas and are the objects of research, design, automation, observation and control, quality assurance, optimal process management, analysis and forecasting of safety states, accident prevention, etc. Modeling methods and tools for this class of objects are of significant theoretical and practical importance both at the design stage and during operation. When developing parallel DNODP-MIMD simulators, the urgent task of evenly distributing the CPU load involved in solving the object equation system must be solved. The article looks at the possible level of parallelization of the DNODP model "Network Graph Branch - MIMD Process / Processor" from the point of view of computing resource load distribution and proposes an approach to balancing it. The idea behind load balancing is to subordinate the "Minimum length branch - one process / processor" type and formalize the DNODP-Simulation model according to the secondary topology of the network graph, which is the result of this subordination and the application of the block computing method (BCM).

Keywords: *dynamic network objects with distributed parameters (DNODP); virtual levels of parallelization; download distribution; parallel MIMD simulator; block computing method (BCM).*

Svyatnyy Volodymyr Andriiovych – Doctor of Technical Sciences, Professor, State Higher Education Establishment “Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: vsvjatnyj@gmail.com

Kushnarenko Volodymyr Hennadiiovych – Academic employee, Institute of Information Resource Management, Ulm University (Albert-Einstein-Allee, 43, 89081 Ulm, Germany).

E-mail: volodymyr.kushnarenko@uni-ulm.de

РЕАЛИЗАЦИЯ СХЕМ АВТОМАТОВ МИЛИ НА СЧЁТЧИКАХ

Баркалов А. А., Титаренко Л. А., Ковалев С. А.

Предложен метод уменьшения числа элементов табличного типа (LUT) в схеме автомата Мили. В работе рассматривается метод кодирования стабильных пар логических условий. Этот подход позволяет уменьшить число аргументов в функциях возбуждения памяти. При этом уменьшается число входов блока замены логических условий дополнительными переменными. Рассмотрен пример синтеза и приведены результаты исследований.

Ключевые слова: микропрограммный автомат Мили, синтез, FPGA, LUT, пары логических условий, счётчик состояний.

Введение. Одной из наиболее важных проблем, связанной реализацией цифровых схем, является уменьшение аппаратных затрат [1]. Для решения этой проблемы необходимо учитывать особенности цифровой схемы и элементного базиса [2]. Одной из наиболее трудных проблем в этой области является оптимизация схем устройств управления. Это связано с их нерегулярной структурой [2]. В этой статье предлагается метод оптимизации схемы микропрограммного автомата (МПА) Мили, реализующего некоторый алгоритм управления. Мы сосредотачиваем усилия на уменьшении площади кристалла, занимаемой схемой МПА. Как известно, решение этой задачи позволяет уменьшить время задержки сигналов и потребляемую мощность [2-4].

Как правило, в схемах МПА Мили используются регистры, хранящие коды состояний [2]. В этой статье мы рассматриваем случай, когда регистр заменен счетчиком [5]. В качестве элементного базиса мы выбрали СБИС типа FPGA (field-programmable gate arrays) [6-7]. Этот базис является основным при реализации сложных цифровых систем [3]. При этом схема реализуется на элементах табличного типа, называемых LUT (look-up table) [6-7]. Для реализации схем счётчиков используются триггеры, связанные через мультиплексоры с выходами элементов LUT.

Основной особенностью элементов LUT является ограниченное число входов (S_L). Например, для семейства Virtex-5 $S_L=5$, для Virtex-7 $S_L=6$ [7]. Считается, что $S_L=6$ является оптимальным [8] с точки зрения площади, занимаемой элементом, и потребляемой им энергии.

Основной особенностью МПА Мили является зависимость его выходных сигналов (микроопераций) от входных сигналов (логических условий) и состояний [2]. Системы булевых функций (СБФ), представляющие микрооперации (M_0), могут зависеть от десятков аргументов [2]. Этот дисбаланс приводит к многоуровневым схемам МПА [9-10]. Для таких схем характерно значительное число межсоединений и, как следствие, значительная паразитная емкость [2]. Как известно, чем больше паразитная емкость, тем больше задержка сигнала и потребляемая мощность [11]. Для уменьшения числа аргументов в СБФ, мы предлагаем заменить логические условия (ЛУ) некоторыми дополнительными переменными. Такой подход является развитием идей из работы [12].

Синтез автоматов на счётчиках. Поведение автоматов может быть задано при помощи таблиц либо графов [2-3]. В данной работе мы используем язык граф-схем алгоритмов (ГСА) [2], обладающий значительной наглядностью для иллюстрации методов синтеза. Рассмотрим в качестве примера ГСА Γ_1 (рис.1).

Для отметки состояний на ГСА Γ_1 использован метод из работы [2]. Анализ ГСА Γ_1 позволяет найти множество состояний $A=\{a_1, \dots, a_8\}$, логических условий $X=\{x_1, \dots, x_4\}$ и микроопераций $Y=\{y_1, \dots, y_6\}$. Пусть $|A|=M$, $|X|=L$ и $|Y|=N$. В рассматриваемом примере $M=8$, $L=4$ и $N=6$. Кроме того, пути вида $\langle a_m, x_h, y_h, a_s \rangle$ определяют переходы между состояниями $a_m, a_s \in A$. При этом a_m – исходное (текущее) состояние; a_s – состояние перехода; x_h – входной сигнал, определяющий переход $\langle a_m, a_s \rangle$ и равный конъюнкции некоторых ЛУ $x_i \in$

X (или их отрицаний); y_h – выходной сигнал, формируемый на переходе номер h ($h=1, N$).

Для автоматов практического объема [2], $L \approx 50$, $N \approx 100$, $M \approx 200$, $H \approx 2000$. Этот факт определяет значительную сложность СБФ, задающих схему МПА.

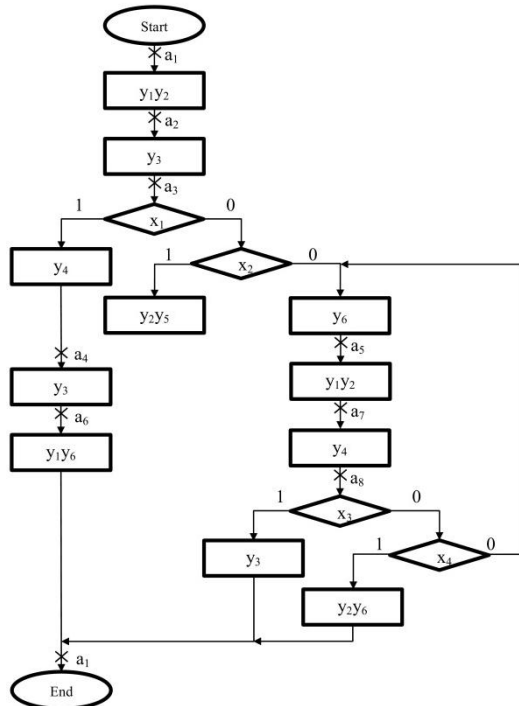


Рис.1. Отмеченная ГСА Γ_1

Для реализации схемы МПА необходимо закодировать состояние $a_m \in A$ двоичными кодами $K(a_m)$ [2]. Для кодирования состояний используются элементы множества внутренних переменных $T = \{T_1, \dots, T_R\}$. Коды $K(a_m)$ хранятся в специальном регистре состояний (RGS), состоящем из R двухтактных триггеров типа D [9]. Для смены содержимого RGS используются функции возбуждения памяти (ФВП), образующие множество $\Phi = \{D_1, \dots, D_R\}$.

Схема МПА задается СБФ

$$\Phi = \Phi(T, X); \quad (1)$$

$$Y = Y(T, X); \quad (2)$$

Каждая функция $D_r \in \Phi$ и $y_h \in Y$ реализуется как совокупность элементов LUT, называемой нами LUTer. Регистр RGS распределен между LUTer D_1, \dots, D_R . Для обнуления RGS используется импульс Start. Импульс Clock разрешает смену состояния (содержимого RGS). Всё это определяет структурную схему МПА Мили U_1 (рис.2).

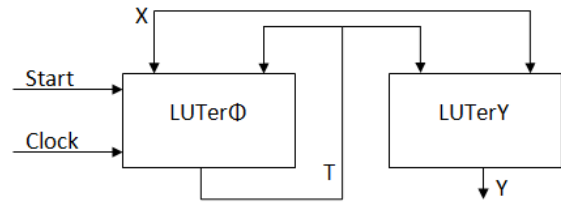


Рис.2. Структурная схема МПА U_1

Символ LUTerΦ соответствует блоку, включающему подсхемы LUTer D_1, \dots, D_R и распределенный регистр RGS. Символ LUTerY соответствует блоку, состоящему из подсхем LUTerY₁, ..., LUTerY_n. При этом LUTerY_n реализует схему для МО $y_n \in Y$.

Для уменьшения числа термов в СБФ (1), регистр RGS можно заменить счетчиком состояний (CTS). При этом некоторые переходы осуществляются путем увеличения содержимого счётчика. Как правило, этот подход используется для автоматов Мура [5]. Однако, в [12] предлагалось заменить RGS на счётчик для МПА Мили. Это ведет к МПА Мили U_2 (рис.3).

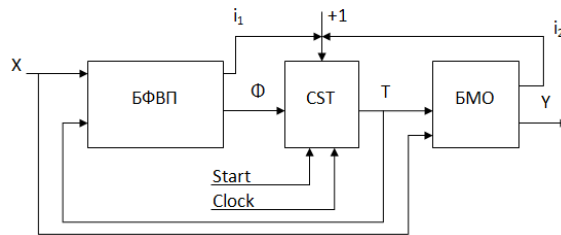


Рис.3. Структурная схема МПА U_2

В автомате U_2 , блок БФВП формирует СБФ (1) и сигнал $i_1 \# CT := CT + 1$. Блок БМО формирует СБФ (2) и сигнал $i_2 \# CT := CT + 1$. Блок БФВП формирует i_1 для условных переходов, а БМО – i_2 для безусловных [12].

Однако в случае FPGA, уменьшение числа термов не влияет на число LUT элементов в схеме МПА [3,9]. Для решения этой задачи необходимо уменьшать число аргументов в реализуемых СБФ. В настоящей работе мы предлагаем один из возможных подходов к решению этой задачи.

Основная идея предложенного метода. Для реализации функции СБФ f_1 , имеющей $NL(f_1)$ литералов, достаточно одного элемента LUT, если следующие условия имеют место:

$$NL(f_1) \leq S_L. \quad (3)$$

При нарушении условия (3) для реализации схемы требуется несколько

элементов. В этом случае схема имеет не менее двух уровней.

Если (3) нарушается, то мы предлагаем два метода, одновременное применение которых приводит к оптимизации схемы МПА Мили на счётчике. Это следующие методы:

1. Замена ЛУ $x_i \in X$ дополнительными переменными $P_g \in P = \{P_1, \dots, P_G\}$, где $G \ll L$.

2. Кодирование стабильных пар ЛУ, определяющих переходы $\langle a_m, a_s \rangle$, элементами множества $Z = \{z_1, \dots, z_{RL}\}$. Пусть N_P - число стабильных пар ЛУ. Тогда для их кодирования достаточно RL переменных:

$$R_L = \lceil \log_2 N_P \rceil. \quad (4)$$

Пусть функции $z_r \in Z$ формируются блоком БМО. Тогда МПА Мили на CTS может быть представлен в виде U_3 (рис.4).

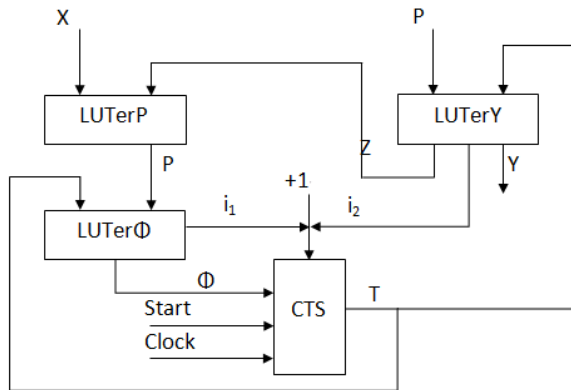


Рис.4. Структурная схема МПА Мили U_3

В схеме на рис.4 блок LUTerP реализует СБФ

$$P = P(Z, X). \quad (5)$$

Функции (5) используются как аргументы функции

$$\Phi = \Phi(T, P); \quad (6)$$

$$i_1 = i_1(T, P). \quad (7)$$

Функции (6)-(7) реализуется блоком LUTerPhi, соответствующем БФВП (рис.3). Блок LUTerY формирует СБФ

$$Y = Y(T, P); \quad (8)$$

$$Z = Z(T, P); \quad (9)$$

$$i_2 = i_2(T, P). \quad (10)$$

Блок LUTerY соответствует блоку БМО (рис.3).

Переменные $T_r \in T$ кодируют состояния $a_m \in A$. Как правило, для МПА, реализуемых на элементах LUT, число R определяется как

$$R = \lceil \log_2 M \rceil. \quad (11)$$

Мы будем использовать (11) в нашем методе.

В настоящей статье предлагается метод синтеза МПА U_3 , который включает следующие этапы:

1. Отметка ГСА Γ состояниями автомата Мили.

2. Создание множества линейных цепей состояний.

3. Кодирование состояний МПА.

4. Формирование множества стабильных пар логических условий.

5. Кодирование пар ЛУ и замена ЛУ.

6. Формирование таблиц переходов МПА U_3 .

7. Формирование СБФ (5) – (10).

8. Реализация схемы МПА в заданном базисе.

Пусть символ $U_i(\Gamma_j)$ означает, что модель U_i используется для синтеза схемы, начиная с ГСА Γ_j . Рассмотрим пример синтеза МПА $U_3(\Gamma_1)$.

Примеры синтеза МПА U_3 . Центральное место в предлагаемом методе занимает второй этап. Определим линейную цепь состояний (ЛЦС) как вектор d_i ($i = \overline{1, I_C}$), элементами которого являются состояния. При этом для любой пары соседних компонент существует условный или безусловный переход из компонента с меньшим индексом в компонент с большим на единицу индексом. В результате выполнения этого этапа, формируется разбиение Π_C множества состояний A на I_C цепей. Для оптимизации схемы, необходимо найти разбиение $\Pi_C = \{d_1, \dots, d_{I_C}\}$ с минимально возможным числом ЛЦС [5].

Для автомата $U_3(\Gamma_1)$ множество A включает $M=8$ состояний. Используя метод [5], найдём $\Pi_C = \{d_1, d_2\}$, где $d_1 = \langle a_1, a_2, a_3, a_4, a_6 \rangle$, $d_2 = \langle a_5, a_7, a_8 \rangle$.

Состояния $a_m \in A$ необходимо закодировать так, чтобы для любой пары компонент выполнялось условие

$$K(a_i) = K(a_{i-1}) + 1. \quad (12)$$

Назовём такое кодирование натуральным.

Для нашего примера из (11) следует, что $R=3$, $T = \{T_1, T_2, T_3\}$ и $\Phi = \{D_1, D_2, D_3\}$. Закодируем состояние так, как показано на рис.5.

$T_1 T_2 T_3$	000	001	010	011	100	101	110	111
	a_1	a_2	a_3	a_4	a_6	a_5	a_7	a_8

Рис.5. Кодирование состояний автомата

В рассматриваемом примере существует две пары логических условий: $B_1 = \langle x_1, x_2 \rangle$ и $B_2 = \langle x_3, x_4 \rangle$. Это следует из анализа ГСА Γ_1 . Итак, $N_p=2$ и из (4) следует, что $R_L = 1$, $Z = \{z_1\}$. Закодируем пары произвольным образом: $K(B_1) = 0$ и $K(B_2) = 1$.

Таблица замены ЛУ строится тривиальным образом. Её столбцы отмечены кодами $K(B_n)$, а строки переменными $p_g \in P$. Для рассматриваемого примера $G=2$, $P=\{p_1, p_2\}$. Один из вариантов замены показан в табл.1.

Из табл.1 можно найти СБФ (5). Для нашего примера имеем:

$$\begin{cases} p_1 = \bar{z}_1 x_1 \vee z_1 x_3; \\ p_2 = \bar{z}_1 x_2 \vee z_1 x_4. \end{cases} \quad (13)$$

Система (13) определяет схему блока LUTerP.

Таблица 1. Таблица замены ЛУ для $U_3(\Gamma_1)$

$\begin{matrix} Z \\ P \end{matrix}$	0	1
P_1	x_1	x_3
P_2	x_2	x_4

Таблица переходов МПА U_3 содержит следующие столбцы: a_m , $K(a_m)$, a_s , $K(a_s)$, P_h , Y_h , Φ_h , Z_h , h . При этом в столбце Y_h записываются не только МО $y_n \in Y$ формируемые при переходе из a_m в a_s , но и переменные i_1 (для безусловных переходов) и i_2 (для безусловных переходов). Столбец P_1 содержит конъюнкцию дополнительных переменных $P_g \in P$ (или их отрицаний), определяющую переход номер h ($h = \overline{1, N}$). Столбец Φ_h содержит ФВП, принимающее значение 1 для переходов, которые не удовлетворяют условию (12).

Для нашего примера $N=12$. Таблица переходов МПА $U_3(\Gamma_1)$ приведена в табл.2.

Таблица 2. Таблица переходов МПА $U_3(\Gamma_1)$

a_m	$k(a_m)$	a_s	$k(a_s)$	P_h	Y_h	Φ_h	Z_h	h
a_1	000	a_2	001	1	$y_1 y_2 i_1$	-	-	1
a_2	001	a_3	010	1	$y_3 i_1$	-	-	2
a_3	010	a_4	011	p_1	$y_4 i_2$	-	-	3
		a_4	011	$\bar{p}_1 p_2$	$y_2 y_5 i_2$	-	-	4
		a_5	101	$\bar{p}_1 \bar{p}_2$	y_6	$D_1 D_2$	-	5
a_4	011	a_6	100	1	$y_3 i_1$	-	-	6
a_5	101	a_7	110	1	$y_1 y_2 i_1$	-	-	7
a_6	100	a_1	000	1	$y_1 y_6$	-	-	8
a_7	110	a_8	111	1	$y_4 i_1$	-	-	9
a_8	111	a_1	000	p_L	y_3	-	z_1	10
		a_1	000	$\bar{p}_1 p_2$	$y_2 y_6$	-	z_1	11
		a_5	101	$\bar{p}_1 \bar{p}_2$	y_6	$D_1 D_3$	z_1	12

Столбец Z_h включает переменные $z_r \in Z_1$, определяющие пару ЛУ. Очевидно $z_1 = 1$ для состояния a_8 , переходы из которого зависят от ЛУ из пары B_2 . Это дает систему (9), которая для нашего примера определяется как

$$Z_1 = T_1 T_2 T_3. \quad (14)$$

Функции для блоков LUTerФ и LUTerY формируются из таблицы переходов. Блок LUTerФ определяется системами

$$D_1 = D_3 = \bar{T}_1 T_2 \bar{T}_3 \bar{p}_1 \bar{p}_2 \vee T_1 T_2 T_3 \bar{p}_1 \bar{p}_2. \quad (15)$$

$$i_1 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \vee \bar{T}_1 \bar{T}_2 T_3 \vee \dots \vee T_1 T_2 \bar{T}_3. \quad (16)$$

Блок LUTerY определяется системой (14) и системами (17-18):

$$\begin{cases} y_1 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \vee T_1 \bar{T}_2 T_3 \vee T_1 \bar{T}_2 \bar{T}_3; \\ \vdots \\ y_6 = \bar{T}_1 T_2 \bar{T}_3 \bar{p}_1 \bar{p}_2 \vee T_1 \bar{T}_2 \bar{T}_3 \vee T_1 T_2 T_3 \bar{p}_1. \end{cases} \quad (17)$$

$$i_2 = \bar{T}_1 T_2 \bar{T}_3 p_1 \vee \bar{T}_1 T_2 \bar{T}_3 \bar{p}_1 p_2. \quad (18)$$

Последний этап синтеза связан с использованием стандартных пакетов фирм-производителей FPGA [6,7] и VHDL - Verilog - моделей автоматов [11, 13]. Мы не рассматриваем этот этап в нашей статье.

Заключение. Замена регистров состояний счётчиками позволяет уменьшить число элементов LUT в схемах автоматов. Это возможно при условии уменьшения числа аргументов в реализуемых СБФ. Для подобного уменьшения, мы предлагаем модифицировать известный метод замены логических условий. Вместо отдельных условий мы предлагаем кодировать их пары. Такой подход имеет смысл, если число кодирующих переменных остаётся меньше числа внутренних переменных, кодирующих состояния автомата.

Предложенный метод был исследован, используя стандартные тестовые примеры из библиотеки [14]. Исследования проводились с использованием пакета Vivado фирмы

Xilinx и нашего программного пакета ABSYNTH [5]. Пакет [5] генерирует VHDL модели автоматов на уровне языка регистровых передач. В качестве элементного базиса использованы микросхема Virtex - 7 фирмы Xilinx.

Микросхемы семейства Virtex - 7 включают элементы LUT, имеющие $S_L = 6$. Исследования показали, что замена регистров счётчиками и кодирования пар ЛУ позволяет уменьшить аппаратные затраты для схем 35 автоматов (из 48). При этом максимальный выигрыш достигал 19%, а средний - 12%.

Дальнейшее направление наших исследований связано с применением различных методов структурной декомпозиции [5,9] для реализации схем МПА Мили на счётчиках. Кроме того, мы планируем применить эти методы для схем, реализуемых на отличных от FPGA базисах.

Литература

- [1] Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб:БХВ. – Петербург, 2002. – 608 с.
- [2] S. Baranov. Logic synthesis for control automata. – Boston: Kluwer Academic Publishers, 1994. - 257 pp.
- [3] I. Skliarova, V. Sklyarov, A. Sudnitson. Design of FPGA-based circuits using hierarchical finite state machines. – Tallinn: TUT Press, 2012. – 240 pp.
- [4] R. Czerwinski, D. Kania. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. – Berlin: Springer, 2013. – 182 pp.
- [5] A. Barkalov, L. Titarenko, J. Bieganski. Logic synthesis for finite state Machines Based on Linear Chains of States. – Berlin: Springer, 2018. – 218 pp.
- [6] Altera: www.altera.com
- [7] Xilinx: www.xilinx.com
- [8] C. Maxfield. The Design Warriors Guide to FPGAs – Orlando: Academic Press, 2004. – 542 pp.
- [9] V. Sklyarov, I. Skliarova, A. Barkalov, L. Titarenko. Synthesis and Optimization of FPGA-based Systems. - Berlin: Springer, 2014. – 432 pp.
- [10] G. DeMicheli. Synthesis and Optimization of Digital Circuits. – N.Y.: McGraw-Hill, 1994. – 636 pp.
- [11] J. Grout. Digital System Design with FPGAs and CPLDs. – Amsterdam: Elsevier, 2008. – 784 pp.
- [12] A. Barkalov, L. Titarenko, S. Chmielewski. Design of CPLD-based Mealy FSMs with counters. – proceeding of MOCAST 18, 2018. – pp. 1-4.
- [13] Соловьев В. В. Проектирование цифровых систем на основе ПЛИС. – М.: Горячая линия – ТЕЛЕКОМ, 2001. – 636 стр.
- [14] S. Yang. Logic Synthesis and Optimization benchmarks user guide. MCNC. – North Carolina, 1991. – 43pp.

References

- [1] Grushvitskiy R.I., Mursayev A.X., Ugryumov E.P. Designing systems on programmable logic chips. - St. Petersburg: BHV. - Petersburg, 2002. – 608 p.
- [2] S. Baranov. Logic synthesis for control automata. – Boston: Kluwer Academic Publishers, 1994. - 257 pp.
- [3] I. Skliarova, V. Sklyarov, A. Sudnitson. Design of FPGA-based circuits using hierarchical finite state machines. – Tallinn: TUT Press, 2012. – 240 pp.
- [4] R. Czerwinski, D. Kania. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. – Berlin: Springer, 2013. – 182 pp.
- [5] A. Barkalov, L. Titarenko, J. Bieganski. Logic synthesis for finite state Machines Based on Linear Chains of States. – Berlin: Springer, 2018. – 218 pp.
- [6] Altera: www.altera.com
- [7] Xilinx: www.xilinx.com
- [8] C. Maxfield. The Design Warriors Guide to FPGAs – Orlando: Academic Press, 2004. – 542 pp.
- [9] V. Sklyarov, I. Skliarova, A. Barkalov, L. Titarenko. Synthesis and Optimization of FPGA-based Systems. - Berlin: Springer, 2014. – 432 pp.
- [10] G. DeMicheli. Synthesis and Optimization of Digital Circuits. – N.Y.: McGraw-Hill, 1994. – 636pp.
- [11] J. Grout. Digital System Design with FPGAs and CPLDs. – Amsterdam: Elsevier, 2008. – 784 pp.
- [12] A. Barkalov, L. Titarenko, S. Chmielewski. Design of CPLD-based Mealy FSMs with counters. – proceeding of MOCAS 18, 2018. – pp. 1-4.
- [13] Solov'yev V. V. Design of digital systems based on FPGAs. - М.: Hot line - TELECOM, 2001. – 636 pp.
- [14] S. Yang. Logic Synthesis and Optimization benchmarks user guide. MCNC. – North Carolina, 1991. – 43 pp.

Надійшла до редакції 17.12.2019 р.

Баркалов Александр Александрович – д. т. н., проф., Зеленогурський Університет, Інститут комп'ютерної інженерії та електроніки (ул. Подгорная, 50, 65-246 Зелёна-Гура, Польща).
E-mail: a.barkalov@iie.uz.zgora.pl.

Титаренко Лариса Александровна – д. т. н., проф., Зеленогурський Університет, Інститут комп'ютерної інженерії та електроніки (ул. Подгорная, 50, 65-246 Зелёна-Гура, Польща).
E-mail: i.tittarenko@iie.uz.zgora.pl.

Ковалев Сергей Александрович – к. т. н., доцент, Донецький національний технічний університет (пл. Шибанкова, 2, г. Покровск, 85300, Україна).
E-mail: sergiy.kovalov@donntu.edu.ua

IMPLEMENTATION OF MEALY FINITE-STATE MACHINE ON THE COUNTERS

One of the most important problems associated with the implementation of digital circuits is the reduction of equipment costs. In order to optimize the circuits of control units having irregular structures, methods are considered that reduce the crystal area when implementing the Mealy scheme for Microprogram Automata (MPA). The scheme is implemented on FPGA-type VLSI and the solution of this problem allows to reduce the signal delay time and power consumption.

Typically, Mealy MPA schemes use registers that store state codes. This article looks at the case where the register is replaced by a counter. The scheme is based on table-type elements called LUTs (look-up table). Triggers that are connected via multiplexers to the outputs of the LUT elements are used to implement the counter circuits.

The main feature of LUT elements is a limited number of inputs (5-6). It is considered that the number of inputs equal to 6 is optimal in terms of the area occupied by the element and the energy consumed by it. Since the output signals in MPA Miles depend on the input signals (logical conditions) and states, the system of Boolean functions (SBF) describing the output signals has dozens of

arguments. This results in multi-level MPA circuits and, as a consequence, an increase in power consumption and delay time of the signals in the circuit.

To reduce the number of terms in the SBF, the status register can be replaced with a status counter. In this case some transitions are made by increasing of the counter content. A method for reducing the number of table type elements (LUTs) in a Mealy circuitry by encoding stable pairs of logical conditions is also proposed. This approach reduces the number of arguments in memory excitation functions. This reduces the number of inputs of the block replacement logic conditions with additional variables. Synthesis example is considered and studies for standard test cases using the Xilinx Vivado package and the ABSYNTH software package have been conducted. Studies have shown that the replacement of registers with counters and coding of logical conditions pairs can reduce the hardware costs for circuits of 35 automata (out of 48). At the same time the maximum gain reached 19%, and the average - 12%.

Keywords: *Mealy FSM, synthesis, FPGA, LUT, pairs of logical conditions.*

Barkalov Aleksandr Aleksandrovich – Prof. Dr., University of Zielona Gora, Inst. Computer Engin. and Electronics (Podgorna Street 50, 65-246 Zielona Gora, Poland).

E-mail: a.barkalov@iie.uz.zgora.pl.

Titarenko Larisa Aleksandrovna – Prof. Dr., University of Zielona Gora, Inst. Computer Engin. and Electronics (Podgorna Street 50, 65-246 Zielona Gora, Poland).

E-mail: i.tittarenko@iie.uz.zgora.pl.

Kovalov Serhii Oleksandrovych – PhD, Associate Professor, State Higher Education Establishment “Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: sergiy.kovalov@donntu.edu.ua

РЕАЛІЗАЦІЯ СХЕМ АВТОМАТІВ МІЛІ НА ЛІЧИЛЬНИКАХ

Однією з найбільш важливих проблем, пов'язаних з реалізацією цифрових схем, є зменшення витрат апаратури. Для оптимізації схем пристроїв управління, які мають нерегулярні структури, розглядаються методи, що зменшують площу кристала при реалізації схеми мікропрограмного автомата (МПА) Мілі. Схема реалізується на НВІС типу FPGA і рішення цього завдання дозволяє зменшити час затримки сигналів та споживану потужність.

Як правило, в схемах МПА Мілі використовуються регістри, що зберігають коди станів. У цій статті розглядається випадок, коли регістр замінюється лічильником. При цьому схема будується на елементах табличного типу, званих LUT (look-up table). Для реалізації схем лічильників використовуються тригери, які пов'язані через мультиплексори з виходами елементів LUT.

Основною особливістю елементів LUT є обмежене число входів (5-6). Вважається, якщо кількість входів дорівнює 6, це є оптимальним з точки зору площі, займаної елементом, і споживаної ним енергією. Оскільки в МПА Мілі вихідні сигнали залежать від вхідних сигналів (логічних умов) і станів, система булевих функцій (СБФ), що описує вихідні сигнали, має десятки аргументів. Це призводить до багаторівневих схем МПА і, як наслідок, до збільшення споживаної потужності та часу затримки сигналів в схемі.

Для зменшення числа термів в СБФ, регістр станів можна замінити лічильником станів. При цьому деякі переходи здійснюються шляхом збільшення вмісту лічильника. Також запропонований метод зменшення числа елементів табличного типу (LUT) в схемі автомата Мілі за рахунок кодування стабільних пар логічних умов. Цей підхід дозволяє зменшити число аргументів у функціях збудження пам'яті. При цьому зменшується число входів блоку заміни логічних умов додатковими змінними. Розглянуто приклад синтезу і проведені дослідження для стандартних тестових

прикладів з використанням пакета Vivado фірми Xilinx і розробленого програмного пакета ABSYNTH. Дослідження показали, що заміна регістрів лічильниками і кодування пар ЛУ дозволяє зменшити апаратні витрати для схем 35 автоматів (з 48). При цьому максимальний виграш досягав 19%, а середній - 12%.

Баркалов Олександр Олександрович – д. т. н., проф., Зеленогурський Університет, Інститут комп'ютерної інженерії та електроніки (вул. Підгорна, 50, 65-246 Зелена-Гура, Польща).

E-mail: a.barkalov@iie.uz.zgora.pl.

Титаренко Лариса Олександрівна – д. т. н., проф., Зеленогурський Університет, Інститут комп'ютерної інженерії та електроніки (вул. Підгорна, 50, 65-246 Зелена-Гура, Польща).

E-mail: i.tittarenko@iie.uz.zgora.pl.

Ковальов Сергій Олександрович – к. т. н., доцент, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: sergiy.kovalov@donntu.edu.ua

INTEGRATION OF DISTRIBUTED PARALLEL SIMULATION ENVIRONMENT WITH CLOUD-INFRASTRUCTURES

Shcherbakov O., Polsterer K., Svjatnyj V.

This article describes an integration of the subsystems of the distributed parallel simulation environment with cloud infrastructures. A complex support for simulation of the dynamic network object with distributed parameters on Amazon AWS cloud is provided. As well as a tool helping to significantly save running costs for cloud simulations.

Keywords: *distributed parallel simulation environment, cloud computing, AWS spot-market, cost optimization.*

Introduction. Cloud computing market is rapidly growing. Recent studies [1] predict that spending on cloud services will grow by 16.5% in 2016 – up to \$204 billion. The largest increase is expected for cloud services (infrastructure as a service [IaaS]). Cloud in [2] is defined as “a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resource(s) based on service-level agreements established through negotiation between the service provider and consumers.”

In [2], and then [3] and others cloud computing called a „fifth utility” (after water, electricity, gas, and telephony), because it is necessary to meet the needs of everyday life and should be available at any time when needed, and be paid for the volume of services used.

The concept of integration of DPSE with cloud-infrastructures. Works [4, 5, 6, 7] describe a distributed parallel simulation environment (DPSE) as a user-friendly system organization of a collaboration of hardware, parallel computing resources, system- and modeling-software, which supports development, implementation and usage of models of complex dynamic systems with distributed and concentrated parameters. DPSE is described as a comprehensive system of hardware and software (system and special). In this aspect a decomposition of DPSE into subsystems was proposed, what on one hand simplifies the development and support of DPSE through a modular approach, on the other hand – preserves the logical and functional integrity of a system being developed.

To determine the aspects of integration of DPSE with cloud infrastructures consider the representation of DPSE on the Software as a

Service (SaaS) principle. In this case, the user does not need to own any computing resources. Access to the simulation environment is done through the browser, if necessary specialized applications can be used. Hardware components of such DPSE may include: servers available to the DPSE administrator; HPC-clusters belonging administrator or a particular user, or those to which they have access; cloud resources ordered by administrator or user, or those that can be ordered; user clients (hardware) can be used with limitations. We will later consider the case of “zero ownership” – when DPSE administrator does not own any hardware resources, but can order the necessary resources by cloud providers.

There are different requirements to the subsystems determining hardware resources necessary for their work. Some subsystems have to be always available – *dialog, databases, IT-support*. Others may work “on demand”. According to the needs in computational power we can distinguish subsystems that require a lot of computing resources (including parallel distributed ones) and those for which the parallelization of work is impractical.

Depending on the type of user access, the type of used hardware resources and the type of available HPC resources, following requirements and restrictions may apply due to network security.

The “interface subsystems” – those that provide network interaction with users and remote resources – are located on a server in the local network. DPSE will be accessed only by local users. There are no restrictions on the usage of local and remote HPC-resources. Usage of cloud resources can have slight restrictions – it is not possible to receive notifications about status changes of the cloud due to local network

security policies, such information must be retrieved and/or updated on time intervals. Disadvantages of the configuration: external users can not access DPSE, restrictions on feedback from the cloud (and other external) services. Advantages: making external access impossible reduces the risk of unauthorized access, such system can have no restrictions on access to local HPC-resources.

The "interface subsystems" are located in the demilitarized zone (DMZ) [8, p. 388], with external access available. External and local users will have access to DPSE, usually through a global network. Usage of local HPC-resources is possible for the resources in this (or other neighboring) DMZ only. Access to remote HPC and cloud resources is possible, but it is important to keep in mind that in case of DPSE server being compromised, accounts used to access any resources (including cloud) will also be compromised.

The "interface subsystems" are located on servers of external (including cloud ones) provider. Choosing a placement strategy of the subsystems is guided by the availability of hardware resources, funds and human resources to support – systems with an external access are potentially vulnerable and require constant maintenance and keeping their software up to date.

DPSE subsystems that do not need to be constantly available – equations solver, visualization, etc. – can be started "on demand", usually on HPC resources. Running all subsystems that do not use parallelization on the same server is possible, but it can cause high server load causing difficulties with access to the DPSE.

Consider a typical scenario of working with DPSE.

1. User being authorized in the system.
2. Selects, load or describes the network desired topology of a network object.
3. Topology being analyzed.
4. Equations describing a given topology being generated in a convenient form for simulation.
5. The analysis of available (to the user) and of necessary parallel simulation resources is being made.
6. Generated equations being adapted to available resources.
7. Prepared simulation model being sent to the queue for running on the HPC.

8. Simulation results being post processed and stored.

9. Results being presented to the user.

For this scenario DPSE needs the following hardware resources for its work: a server for subsystems that provide interaction with the user and other subsystems; resources for other subsystems – HPC-resources for parallel simulation, other available servers.

Consider the following scenario of use: access to DPSE is available to external users, thus DPSE server is located outside the security perimeter and has no access to the local cluster. DPSE can run parallel simulations on HPC resources available to its users or can order computing resources by cloud providers. We consider DPSE integration with cloud services of Amazon. A significant advantage of this provider for scientific simulations is the offer of spot-instances [9] – free resources are sold on the principle of trading. This allows the user to save a lot. Stated that the price of these instances is defined by bets of all users of the service and the number of free instances offered for trades. The disadvantage of this proposal consist in having no guarantees of continuous work of a spot-instance. If the auction price rises above the bet of the user, his instance(s) will be shut down and given to someone else. There is an automatic notification shortly before the shut down but it is not guaranteed.

DPSE subsystems for *dialog*, *databases* and *IT-support* must always be available to users, thus they may run only on standard "on demand" instances. Other DPSE subsystems related to the preparation of simulation models, simulation and processing of results, can be run on spot-instances and other HPC-resources available to the user. Interrupting their work is less critical, in addition their algorithm may provide a mechanism of restoring the interrupted simulation (checkpointing). It is worthwhile to order cloud resources for these subsystems only when users run a simulation and to free them when the simulation is complete.

Prototype of the simulation environment DPSE – cloud-services. Work with cloud resources on more or less complex systems needs automation. First of all placing the bets must be automated for spot-instances when starting the simulation, deleting unused instances after simulation, analysis of prices for spot-instances.

A prototype of DPSE was developed [10]. It runs completely in cloud environments and

provides support for the simulation of dynamic network object (DNO) at all stages – from model development to retrieving the results after parallel execution on cloud resources. Amazon AWS Services are used as a cloud provider. Ten subsystems of DPSE, as described in [4, 5, 6, 7], are represented in the various components of the prototype. The subsystems for the *IT-support* and *dialog* run on the EC2 on-demand instance. Amazon RDS service represents both software and hardware parts of the *database* subsystem. Subsystem for *load balancing* as well as a subsystem for *data exchange* are based on the CfnCluster [11] framework. Gnuplot [12] is used as a simple implementation of the *visualization* subsystem. Other DPSE subsystems – for *topological analysis*, for *equations generation*, for *virtual parallel simulation models*, for *parallel equation solvers* – provide at first hand the simulation of DNO. They run on a cloud cluster, running simulation on other HPC clusters is also supported. Software parts of these subsystems are briefly described below.

Currently the DNO model with distributed parameters is implemented and available for all users. A model developer has the possibility to prepare a description of different topologies. This description is provided as a text file in a specific format. A user can select one of the available topologies for simulation and define different model parameters: numerical method, maximum number of parallel processes and so on. Whereat the topology analysis, equations generation and analysis of virtual parallel simulation-models are performed automatically for the model with these parameters.

The numerical solution of the Cauchy problem (for previously generated equations) is provided by GNU Scientific Library (GSL) [13], which is quite popular in the scientific community (over 1000 citation of GSL documentation according to Google Scholar), is cross-platform and has a simple and well structured source code. GSL can be easily extended with the implementation of other numerical methods. Implementation of the block methods is planned for the future [14, 15]. The results of the simulation are visualized by Gnuplot as three-dimensional plots and stored in a git repository. This way of storing and accounting the results is not suitable for models with huge amount of output data, and for long time storage, but it is implemented to demonstrate the work of DPSE and to simplify

models debugging. Results of the simulation are stored in a local (DPSE server) repository with mirroring to the external one – GitHub and AWS CodeCommit. The question of optimally storing and managing simulation results is still open.

Besides DNOs other types of models are supported and can be launched, as well as creating and managing standalone cloud clusters. User can specify the desired cluster configuration to be created, and a maximum price for the spot-instances. Because CfnCluster framework instances that are not used are automatically turned off and are turned on again when required, costs and administrative time is saved.

A variety of additional DPSE features improve the work of simulation model developers and users. The developed prototype has a built-in version control system (git) for the models, supports accounting of experiments on HPC and allows to open terminal sessions directly in a browser. Instant notifications on a mobile phone about the status of cloud cluster and simulations helps to organize simulation workflow and to save time.

A function for monitoring and analysis of spot instance prices allows user to understand the dynamic (fig. 1) of AWS spot prices and helps to choose cheapest spot-instances or to choose a proper bet for certain instance. Fig. 2 shows an example price comparison, instances are sorted by the rising cost of virtual performance (conditionally “price per gigahertz”).

Assume we have a model to launch on the cloud cluster with the lowest budget. The model has been executed with test data on different types and different numbers of instances. For simplification we take into account only two types of EC2 spot-instances: *c4.large* and *c4.8xlarge*. *Frankfurt* is chosen as an AWS region. Execution time of this simple benchmark model in minutes is shown in table 1. Having this benchmark done helps to choose a better type of instances with respect to cost / performance for models of this type and to make a better parallelization. Table 2 shows a price comparison for one simulation running on the *onDemand*- and on *spot*-instances. Full utilization is assumed. Average spot prices were computed by the spot-market analysis tool of DPSE (fig. 2).

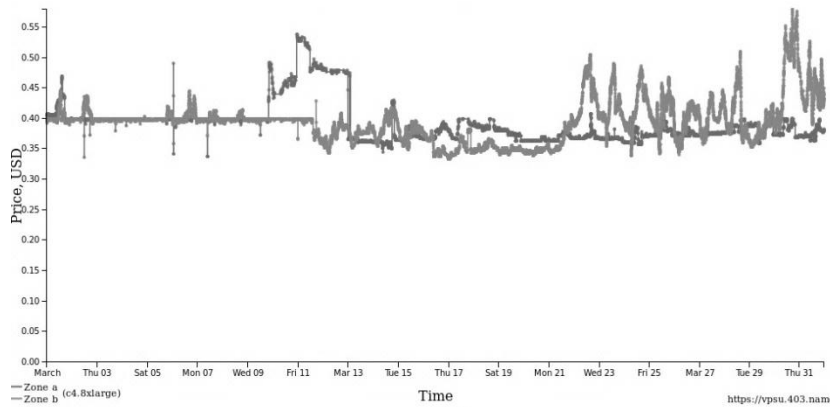


Fig. 1. Spot-price changes for *c4.8xlarge* instance in March 2016, region *Frankfurt*

ECU	Instance	vCPU	RAM	Max Price	Min Price	min Avg Price	σ	σ (A)	σ (B)	min Acc Price	min Acc / Core	min Acc / ECU
8.0	<i>c4.large</i>	2	3	0.1500	0.0171	0.0205	0.0061	0.0072	0.0061	15.2365	7.6182	1.9046
16.0	<i>c4.xlarge</i>	4	7	2.9600	0.0419	0.0476	0.0978	0.1216	0.0978	35.3383	8.8346	2.2086
31.0	<i>c4.2xlarge</i>	8	15	0.5510	0.0817	0.1383	0.0807	0.0807	0.0870	102.7768	12.8471	3.3154
62.0	<i>c4.4xlarge</i>	16	30	1.1250	0.1658	0.1794	0.0120	0.1124	0.0120	133.3041	8.3315	2.1501
132.0	<i>c4.8xlarge</i>	36	60	0.5793	0.3334	0.3930	0.0339	0.0339	0.0350	292.0196	8.1117	2.2123

Fig. 2. Prices analysis results for spot instances in March 2016, region *Frankfurt*

Table 1. Simulation benchmarking for different types of instances

# of instances	Instance type	Duration (min)	Runs/hour
1	<i>c4.large</i>	14	4.3
2	<i>c4.large</i>	8	7.5
3	<i>c4.large</i>	6	10
4	<i>c4.large</i>	5	12
6	<i>c4.large</i>	5	12
1	<i>c4.8xlarge</i>	1	60

The performance of the simulations on 4 and 6 instances is the same because of relatively slow interconnections of instances done with 1Gbit-Ethernet and intensive data exchange during the simulation of this model.

Table 2. Prices per run for the benchmarked model

# of instances	Instance type	OnDemand price/run	Spot price/run
1	<i>c4.large</i>	0.0312	0.0048
2	<i>c4.large</i>	0.0357	0.0055
3	<i>c4.large</i>	0.0402	0.0062
4	<i>c4.large</i>	0.0447	0.0068
1	<i>c4.8xlarge</i>	0.0356	0.0066

Values for 6 *c4.large* instances are removed as they are redundant and will wrongly increase a calculation of the resulting benefit.

Table 2 lets to compute the benefit from using spot-instances. For running a test model in March 2016 it can be about 650%. Running a benchmark on the instances of different type and choosing the best one lets to benefit by about 140%. Both together can give 910%.

At the moment of writing this article an automation for described benchmarking and further research on cost optimization are under development.

Conclusions. The growing popularity of cloud service providers is caused by their affordability and flexibility. For scientific purposes the possibility to order unused resources with a substantial discount (spot instances), for tasks not requiring real-time simulation, but still having need for powerful (CPU speed, available memory, etc.) resources, seems to be attractive.

A prototype of DPSE providing support for all phases of modeling and simulation of dynamic network objects and other models was developed. Running DPSE entirely on cloud resources allowed to prepare the ground for further experiments on the feasibility of using cloud resources for different classes of simulations.

An example of saving up to 910% of simulation costs for running a certain model in

cloud was shown. This was achieved by choosing an optimal type of instance for the presented simulation model and by carefully observing the prices on the spot-market. This benefit will differ for other types of models.

Calculations was made without taking into account the prices for EBS storage, EBS I/Os and data transfer for downloading the results of the simulations. This will be done in future research.

References

- [1] Gartner says worldwide public cloud services market is forecast to reach 204 billion in 2016, 01 2016. <http://www.gartner.com/newsroom/id/3188817> (retrieved 29.03.2016).
- [2] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25 (6):599–616, 2009.
- [3] Jose Moura and David Hutchison. Review and analysis of networking challenges in cloud computing. *Journal of Network and Computer Applications*, 60:113 – 129, 2016.
- [4] L.P. Feldmann, V.A. Svjatnyj, M. Resch, and M. Zeitz. Forschungsgebiet: Parallele Simulationstechnik. *Modelling and Computer Graphics - 2009*, 2009.
- [5] L.P. Feldmann, V.A. Svjatnyj, M. Resch, and M. Zeitz. Forschungsgebiet: Parallele Simulationstechnik. In *ASIM 2014 – 22. Symposium Simulationstechnik*, pages 3–7, 2014.
- [6] V. Svjatnyj, V. Kushnarenko, O. Shcherbakov, and M. Resch. Dekomposition der verteilten parallelen simulationsumgebung. *Scientific papers of Donetsk National Technical University. Series "Problems of modeling and design automation"*, 2012.
- [7] O. Shcherbakov and V. Svjatnyj. Decomposition into subsystems and organization of work of distributed parallel simulation environment on the web. *Informatics and Computer Technologies*, 2010.
- [8] E. Maiwald. *Network Security: A Beginner's Guide, Second Edition. Beginner's Guide.* McGraw-Hill Companies, Incorporated, 2003.
- [9] Amazon. Ec2 spot instances. <https://aws.amazon.com/ec2/spot/>.
- [10] Oleksandr Shcherbakov. Distributed parallel simulation environment. <https://vpsu.403.name/>.
- [11] AWS. Cfncluster. <https://aws.amazon.com/hpc/cfncluster/>.
- [12] Thomas Williams, Colin Kelley, et al. Gnuplot 4.4: an interactive plotting program. *Official gnuplot documentation*, <http://sourceforge.net/projects/gnuplot>, 2010.
- [13] Brian Gough. *GNU scientific library reference manual.* Network Theory Ltd., 2009.
- [14] A. Guseva and V. Kushnarenko. Discrete simulation-model of dynamic network object with distributed parameters based on the block numerical method. *Scientific papers of Donetsk National Technical University. Series "Problems of modeling and design automation"*, 9(179):356, 2011.
- [15] V. Kushnarenko. Simulating a branch of dynamic network object with lumped parameters based on the block numerical methods. *Scientific papers of Donetsk National Technical University. Series "Problems of modeling and design automation"*, (210):66–74, 2013.

Надійшла до редакції 18.01.2020 р.

Shcherbakov Oleksandr – PhD-student, Heidelberg Institute for Theoretical Studies (Schloss-Wolfsbrunnenweg 35, Heidelberg, 69118, Germany).

E-mail: oleksandr.shcherbakov@h-its.org.

Kai Polsterer – Dr., Heidelberg Institute for Theoretical Studies (Schloss-Wolfsbrunnenweg 35, Heidelberg, 69118, Germany).

E-mail: kai.polsterer@h-its.org.

Svyatnyy Volodymyr Andriiovych – Doctor of Technical Sciences, Professor, State Higher Education Establishment “Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: vsvjatnyj@gmail.com

ІНТЕГРАЦІЯ РОЗПОДІЛЕНОГО МОДЕЛЮЮЧОГО СЕРЕДОВИЩА З ХМАРНИМИ ІНФРАСТРУКТУРАМИ

У даній статті описується розробка, системна організація та інтеграція з хмарними інфраструктурами розподіленого паралельного моделюючого середовища (РПМС). Автори розглядають організацію роботи РПМС з огляду на наявні чи доступні обчислювальні ресурси з акцентом на використання хмарних сервісів. У статті уточнюються робота та функції підсистем РПМС, враховуючи можливість їх роботи на різних інфраструктурах, питань безпеки та доступності підсистем, а також можливості запуску за вимогою. Наведено прототип середовища, в якому реалізовано функції всіх підсистем. Сам прототип доступний користувачам за принципом «програмне забезпечення як сервіс» (Software as a Service – SAAS), та надає користувачеві підтримку на всіх етапах моделювання – від вибору комп'ютерної моделі до візуалізації отриманих результатів. Роботу підсистем середовища також розглянуто відносно підходу «нульової власності», коли адміністратор (розпорядник) системи не володіє власними обчислювальними ресурсами, а лише замовляє хмарні ресурси у разі потреби – на вимогу користувачів РПМС. Для такої моделі використання ресурсів актуальним стає питання оптимізації витрат на обчислювальні ресурси. В якості можливого рішення автори наводять приклад використання «spot-інстансів» (на прикладі AWS). Доведено, що при виконанні обчислень, для яких (низька) вартість отримання результатів є важливішою ніж швидкість їх отримання, можна отримати значну економію коштів вірним вибором регіону для хмарних ресурсів. Така оптимізація може бути вдосконалена, якщо розглянути швидкодію різних типів ресурсів для певної задачі та розрахувати їх відносну ціну. Різний попит на різні типи spot-інстансів зумовлює нелінійну залежність ціни від швидкодії. Для проведення тестів прототип РПМС доповнено функцією розгортання паралельного хмарного кластеру зі встановленими та налаштованими бібліотеками для паралельних обчислень. Особливістю реалізації є підтримка гетерогенних кластерів – хмарні ресурси у кластері можуть бути різних типів, та мати відповідно різні характеристики та ціни. Поєднання обох підходів до оптимізації витрат у наведеному прикладі дало економію у 910%.

Ключові слова: розподілене паралельне моделююче середовище, cloud computing, AWS spot-market, оптимізація витрат.

Щербakov Олександр Сергійович – аспірант, Гейдельберзький університет (Schloss-Wolfsbrunnenweg, 35, м. Гейдельберг, 69118, Німеччина).

E-mail: oleksandr.shcherbakov@h-its.org.

Кай Польстерер – доктор наук, Гейдельберзький університет (Schloss-Wolfsbrunnenweg, 35, м. Гейдельберг, 69118, Німеччина).

E-mail: kai.polsterer@h-its.org.

Святний Володимир Андрійович – доктор технічних наук, професор, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: vsvjatnyj@gmail.com

РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ МОДЕЛЮВАННЯ СИПКИХ МАТЕРІАЛІВ МЕТОДОМ ДИСКРЕТНИХ ЕЛЕМЕНТІВ

Кучерявенко А. О.

Сипучі речовини залучені в різні промислові процеси. Сучасні компанії потребують точного аналізу та прогнозування поведінки і стану даних матеріалів. Моделювання поведінки сипучих середовищ гарантує якість, точність проектування, допомагає аналізувати, прогнозувати і запобігати отриманню небажаних результатів у промислових процесах. В статті подано основні аспекти розробки мобільного середовища моделювання на основі методу дискретних елементів.

Ключові слова: МДЕ, сипучі речовини, моделювання, мобільність, симуляція.

Загальний опис. Для моделювання механічної поведінки твердих частинок сипких матеріалів широко застосовується метод дискретних елементів (МДЕ). МДЕ є найбільш потужним інструментом для розрахунку динаміки великої кількості частинок розміру мікрон і більше [1, 2]. Цей метод заснований на тому, що кожна тверда частка розглядається як окремий об'єкт, рух якого та взаємодії з іншими об'єктами описуються рівняннями Ньютона (рів. 2, 3). Дискретний підхід точно моделює мікромеханіку гранульованого матеріалу. На рис. 1 наведено загальний принцип розрахунку поведінки частинок за допомогою МДЕ.

Найбільш трудомісткою частиною МДЕ є виявлення контактів між великою кількістю моделей. Умова контакту частинок має наступний вигляд:

$$\delta_{ij} = R_i + R_j - |O_i - O_j| > 0 \quad (1)$$

де R_i і R_j - радіуси часток, δ_{ij} - перетин часток, O_i та O_j - центри часток.

Система рівнянь Ньютона, які вирішуються для всіх частинок, має вигляд [1]:

$$m_p \frac{dv}{dt} = \sum_{i=1}^n F_i \quad (2)$$

$$I_p \frac{dw}{dt} = \sum_{i=1}^n M_i \quad (3)$$

Тут t - час здійснення процесу на певному етапі; m_p - маса p -ї частинки; I_p - момент інерції p -ї частинки; F_i - сили, що діють на частинку (включаючи силу тертя об похилу кругову поверхню); M_i - обертаючий момент, v - швидкість частинки, w - кутова швидкість.

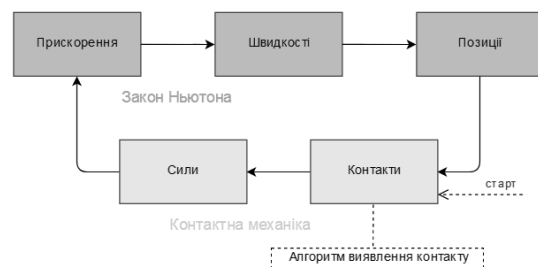


Рис. 1. Загальний алгоритм розрахунку поведінки частинок.

За допомогою розкладання в ряд Тейлора на кожному часовому кроці виконується оновлення позиції, швидкості та обертання частинок:

$$\begin{aligned} g(t + \Delta t) &= g(t) + \Delta t \cdot \frac{dg(t)}{dt} \\ x(t + \Delta t) &= x(t) + \Delta t \cdot v(t) \\ \omega(t + \Delta t) &= \omega(t) + \Delta t \cdot \frac{d\omega(t)}{dt} \end{aligned} \quad (4)$$

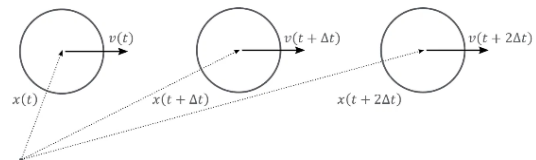


Рис. 2. Оновлення даних частинок на кожному часовому кроці.

Розробка. Основною метою цього проекту є розробка нової відкритої веб-орієнтованої системи для портативного моделювання сипких матеріалів методом дискретних елементів на основі консольної симуляційної програми Musen (стаціонарне програмне забезпечення, розроблено у інституті інженерії твердих процесів і

технологій дрібних частинок у м. Гамбург [3]). Веб-додаток має спростити проведення процесу моделювання за рахунок мобільності, що надає можливість проводити потужні розрахунки з будь-якого пристрою (планшет, ноутбук, телефон). Використовуючи Musen на віддалених серверах, користувач не повинен мати потужні локальні обчислювальні ресурси, це сприяє удосконаленню розуміння процесів та використанню в навчальних цілях.

На рисунку 3 наведено загальний алгоритм роботи веб-додатку.

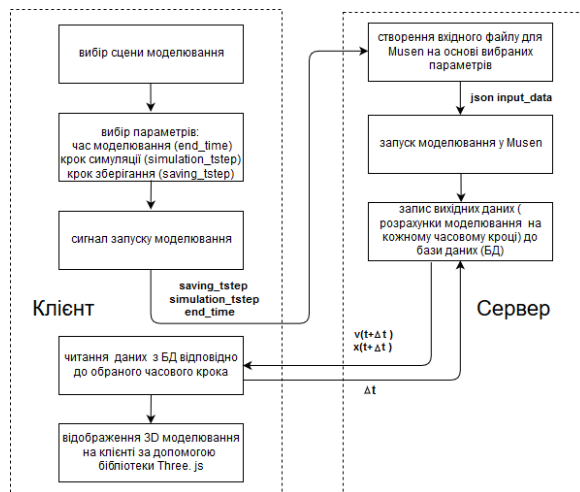


Рис. 3. Загальний алгоритм роботи веб-додатку

Розроблена програма має клієнт-серверну архітектуру. Серверну частину розроблено на базі NodeJS. Для обміну повідомленнями між браузером і веб-сервером в режимі реального часу застосовано протокол WebSocket. Клієнт-сервер співпрацює через Socket.IO та бібліотеку веб-додатків JavaScript. Це дозволяє здійснювати двостороннє спілкування між веб-клієнтами та серверами в реальному часі. У якості бази даних обрано MongoDB. Для відображення анімованої 3D-комп'ютерної графіки на клієнті обрано бібліотеку Three.js.

При відкритті головної сторінки відбувається встановлення зв'язку через websocket з сервером, та користувач отримує унікальний ідентифікатор (user_id).

Для початку моделювання потрібно обрати сцену моделювання в тривимірному просторі і задати наступні параметри

моделювання: крок моделювання (simulation_tstep), крок зберігання результату (saving_tstep) та тривалість моделювання (end_time) (рис.4). За допомогою клавіатури можна виконувати переміщення камери вліво, вправо, вперед і назад, використовуючи відповідні клавіші A, D, W, S. Утримуючи праву та ліву кнопку миші, можливо обертати та переміщувати сцену моделювання.

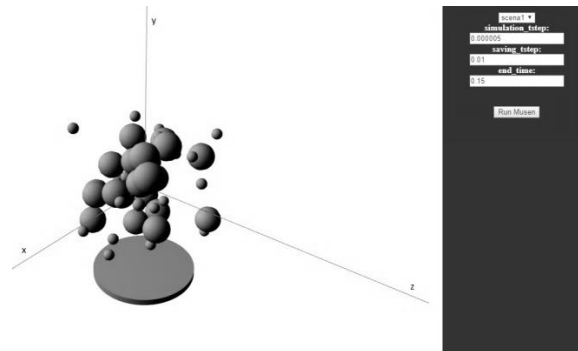


Рис. 4. Структура головної сторінки та ініціалізація параметрів моделювання.

Після запуску моделювання задані параметри у вигляді вхідного ключа (json input_data) передаються на серверну частину, де запускається Musen з отриманими параметрами. Після відпрацювання програми розраховані значення записуються в базу даних. Веб-сервер отримує код завершення Musen, зчитує з бази даних результати першої часової точки, та передає через websocket до клієнта. За допомогою three.js результати відображаються у браузері на результуючій сторінці.

Використовуючи слайдер (рис.5), користувач має можливість переглянути результат на кожному кроці моделювання. При змінні часової точки відбувається запит до сервера з метою зчитати нові параметри всіх частинок (координати, швидкості). На рисунку 5, 6 можна побачити моделювання в різні часові точки.

Також у програмі реалізована можливість розфарбовування частинок відповідно до таких параметрів, як маса, швидкість та координати (рис.6). Це потрібно для розуміння стану поточної системи, наприклад, які частинки ще рухаються, а які зупинились.

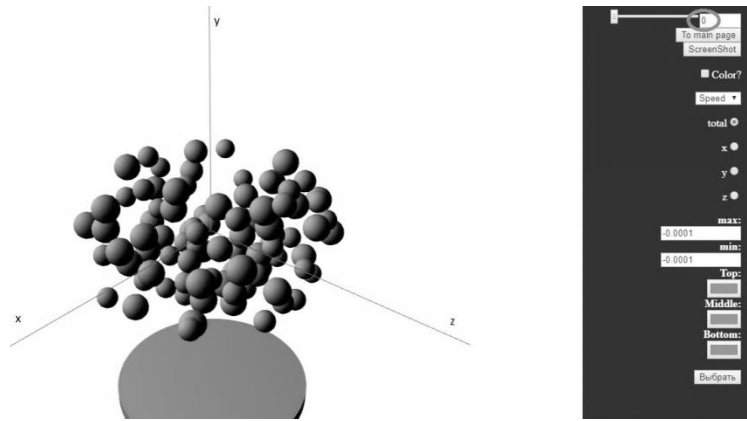


Рис. 5. Початковий стан.

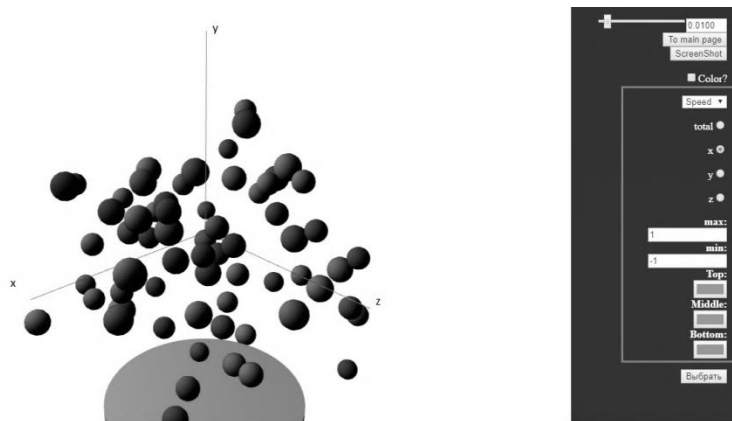


Рис. 6. Стан процесу на кроці 0,01 сек. та розфарбування частинок.

Висновки. Моделювання сипких матеріалів дає можливість передбачити їх поведінку, аналізувати вплив різних параметрів та факторів на технологічні процеси і розробляти стратегії контролю. Описану програму розроблено на базі сучасних веб-технологій, вона має зручний і інтуїтивно зрозумілий інтерфейс користувача. Слід відзначити, що при наявності великої кількості частинок процес

моделювання може тривати від декількох днів до декількох тижнів. На даний час розробляється концепція прискорення обчислень за рахунок паралельних технологій [4, 5]. Саме метод дискретних елементів дозволяє нам розглянути кожну частинку як окремий об'єкт, призначити їй MIMD-процес і виконати паралельне моделювання.

Література

- [1] P.A. Cundall, O.D.L. Strack. A discrete numerical model for granular assemblies. *Geotechnique* 29, 1979, С. 47-65.
- [2] Хорошевський В.Г. Обчислювальні методи, алгоритми та апаратно-програмний інструментарій паралельного моделювання природних процесів // *Інтеграційні проекти*, 2012, С. 202-210.
- [3] Institute of Solids Process Engineering and Particle Technology. [ONLINE] Режим доступу: <https://www.tuhh.de/spe/institute.html> [Перевірено: 30.10.2019].
- [4] Feldmann, L., Resch, M., Svjatnyj, V., Zeitz, M.: Softwarearchitektur für parallele Simulationsumgebungen (Forschungsgebiet: parallele Simulationstechnik). Plenarvortrag, ASIM'2014 in Berlin, Tagungsband 1, ARGESIM Report 43, Wien 2014, С. 3-7.
- [5] Svjatnyj V., Gilles E.-D., Zeitz M., Reuter A., Rothermel K. Simulationssoftware für eine parallele Simulationsumgebung für dynamische Systeme. 14. Symposium ASIM'2000. – Tagungsband, SCS, 2000. – С. 235–240.

References

- [1] P.A. Cundall, O.D.L. Strack. A discrete numerical model for granular assemblies. *Geotechnique* 29, 1979, pp. 47-65.
- [2] Khoroshevskiy V.H. Computational methods, algorithms and hardware and software tools for parallel modeling of natural processes // *Integration projects*, 2012, pp. 202-210.
- [3] Institute of Solids Process Engineering and Particle Technology. [ONLINE] Available at: <https://www.tuhh.de/spe/institute.html> [Accessed 30 October 2019].
- [4] Feldmann, L., Resch, M., Svjatnyj, V., Zeitz, M.: Softwarearchitektur für parallele Simulationsumgebungen (Forschungsgebiet: parallele Simulationstechnik). Plenarvortrag, ASIM'2014 in Berlin, Tagungsband 1, ARGESIM Report 43, Wien 2014, pp. 3-7.
- [5] Svjatnyj V., Gilles E.-D., Zeitz M., Reuter A., Rothermel K. Simulationssoftware für eine parallele Simulationsumgebung für dynamische Systeme. 14. Symposium ASIM'2000. – Tagungsband, SCS, 2000. – pp. 235–240.

Надійшла до редакції 16.10.2019 р.

Кучерявенко Аліна Олександрівна – аспірант, Донецький національний технічний університет (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).
E-mail: akucheriavenko@gmail.com.

WEB APPLICATION DEVELOPMENT FOR MODELING BULK MATERIALS BY DISCRETE ELEMENTS METHOD

The modeling of bulk media is an urgent problem in the context of solving a number of production problems, such as mining of metal ores, designing mine skips and walls, designing dump truck bodies, etc.

One of the main methods for modeling bulk media is the method of discrete elements (MDE), in which the particles of the medium are considered as elastic bodies of arbitrary shape, and the interaction between them is described by the system of equations of Newton's second law for a combination of translational and rotational motions.

This paper presents the results of developing a web application that provides remote access to the Musen simulation environment developed at the Institute for Solid State Process Engineering and Small Particle Technology in Hamburg.

The application has a client-server architecture. The server part of the application includes a layer between the client and Musen (implemented using the Node.JS library), as well as the MongoDB database server. The client part of the application allows you to set the simulation conditions and visualize the results obtained, for which the Three.js library is used. The program allows you to color the granules of the medium in different colors, depending on their mass, speed or position, which makes the simulation results more visual and provides the possibility of a deeper analysis.

The modeling of the dynamics of bulk media requires significant computing power, and can last up to several days or even weeks. Using HPC servers and the concept of parallel computing, in which a particle or group of particles is mapped to a single MIMD process, can significantly reduce the simulation time and associated overhead.

Keywords: MDE, bulk materials, modeling, mobility, simulation.

Kucheriavenko Alina Oleksandrivna – PhD-student, Donetsk National Technical University, (2, Shybankova square, Pokrovsk, Donetsk region, 85300, Ukraine).
E-mail: akucheriavenko@gmail.com.

АНАЛІЗ ТА РОЗРОБКА АЛЬТЕРНАТИВНИХ РІШЕНЬ ІНФРАЧЕРВОНОЇ ПАЙКИ

Зайцев К. С., Шамаєв В. В.

Робота описує розробку альтернативних рішень інфрачервоної пайки та розробку контролюючих приладів цього процесу. Аналіз існуючих рішень в серійних приладах, їх переваги і недоліки дозволяють запропонувати нові шляхи поліпшення характерних особливостей якості пайки за рахунок автоматизації процесу та оптимізації профілів пайки.

Keywords: *інфрачервоне випромінювання, інфрачервона паяльна станція, Arduino, автоматизовані профілі пайки.*

Вступ. Якість готових рішень інфрачервоної (ІЧ) пайки далеко не завжди відповідає заявленій, а дійсно хороші апарати коштують не малих грошей. У процесі пайки, наприклад, підвищення чи утримання температури нагрівача вище потрібної, заданої технологічними вимогами, може негативно вплинути на результат роботи - відбувається зниження надійності паяного з'єднання.

Оператори станцій іноді навмисно перевищують необхідну норму через полегшення паяння на високих температурах, що є тотальною помилкою. Протидіяти цьому важко, тому що у даному випадку присутній людський фактор. Виходом з положення є введення автоматизованого контролю паяльних станцій.

В даній роботі пропонується прилад, котрий буде відповідати сучасним вимогам інфрачервоної пайки, мати зручну та технологічну апаратну частину, доброзичливий інтерфейс з можливістю моніторингу та управління особливостями процесу пайки з комп'ютера.

Для оцінки конкурентоспроможності паяльної станції з віддаленим контролем необхідно провести порівняння з аналогами, виявити переваги і слабкі сторони виробу.

Однією з основних задач при реалізації інфрачервоної паяльної станції є чітке розуміння того, які задачі повинен виконувати прилад, основи фізики процесу, кінцевий результат та можливості приладу. Основою інфрачервоних паяльних станцій є керамічний або кварцовий нагрівальний елемент [1]. Станція має два таких елементи – нижній та верхній і процес розігріву контролюється мікропроцесорною системою управління.

Сучасна паяльна станція повинна забезпечувати:

- пайку BGA - CBGA, CCGA, PBGA, μBGA, FCBGA, LFBGA, CGA, CSP, QFN, QFP, MLF, PGA і інших чипів розміром до 60x60мм;
- можливість пайки не тільки BGA, а також пластикових конекторів;
- контроль температури через аналіз сигналів з термопар;
- автоматичний і ручний режими роботи.
- можливість моніторингу процесу за допомогою комп'ютеру.
- завдання термопрофілів для пайки (як для свинцевмісних припоїв так і для не вмісних), можливість їх зміни та додавання нових.

Апаратна частина. За типом нагрівального елемента паяльні станції поділяють на керамічні та кварцові. Одна з переваг керамічні нагрівачів - це пайка за допомогою випромінювання електромагнітних хвиль невидимого спектру, які є абсолютно безпечними для зору людини і дозволяють операторові робити візуальне спостереження за процесом. Вони також є найбільш надійними і забезпечують досить тривалий час експлуатації до відмови [2].

Кварцові нагрівачі, в свою чергу, відрізняються меншою інерційністю і забезпечують більшу однорідність зони нагрівання, хоча вони використовують крім невидимого ще й видимий, і тому, небезпечний для зору діапазон інфрачервоного випромінювання.

Як відомо, потужність ІЧ випромінювання з одиниці площі поверхні нагрівача пропорційна четвертому ступеневі температури (закон Стефана-Больцмана), а довжина

хвилі з максимальною амплітудою складових спектра, обернено пропорційна температурі. Кварцовий нагрівач в звичайному режимі має температуру нагріву спіралі 2200-2700С, максимум випромінювання на хвилі довжиною 1 мкм. В той же час згідно з графіком на рисунку 1 навіть при такій температурі максимум дуже пологий, в спектрі містяться в тому числі і довгі хвилі, причому амплітуда їх набагато менше максимуму [3]. При зниженні температури спіралі (послідовне з'єднання нагрівачів або діммірування) графік спектру плоскішає і шкідлива короткохвильова частина (довжина хвилі менш ніж 2.5мкм) становить досить малий відсоток загальної потужності.

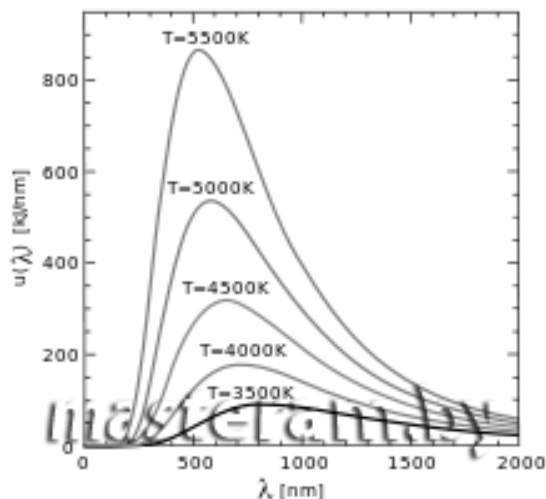


Рисунок 1 – Температурний спектр кварцового нагрівача
(зображення взято з сайту masteram.by)

Таким чином, можна відзначити, що кварцові нагрівачі, які працюють із слабким напруженням, за властивостями випромінювання не дуже відрізняється від промислового кварцового випромінювача. Але більше зниження температури спіралі - до невидимого випромінювання - неможливо, оскільки питома потужність випромінювання з одиниці площі суттєво знижується (вона пропорційна четвертому ступеневі температури), а поверхня випромінювання спіралі в кварцовому нагрівачі дуже мала. У цьому випадку не можливо забезпечити щільність потоку потужності на достатньому рівні. Слід також відмітити, що кварцове скло трубок ламп блокує хвилі довші 4 мкм.

Таким чином можна зробити важливий висновок: для створення якісного ІЧ приладу

з використанням кварцових нагрівачів необхідно створити максимально можливу потужність потоку випромінювання на мінімальній площі. Лампи розташовуються впритул, а необхідна не дуже велика потужність досягається шляхом їх послідовного з'єднання або дімміруванням, тобто вони штучно працюють із слабким напруженням [3].

Експерименти з використанням станції показали, що питомої потужності нижнього нагрівачу 8 Вт / кв. см вистачає не тільки для достатнього розігріву плати з будь-яким допустимими матеріалами, а й для випаювання різноманітних роз'ємів та сокетів взагалі без верхнього нагрівача, за допомогою одного нижнього підігріву, захистивши решту плати тепло-ізолюючим матеріалом. Для правильного функціонування верхнього нагрівача необхідна потужність не менш 10-12 Вт / кв. см.[4,5]. Також слід враховувати, що кварцові нагрівачі є нелінійним елементом, і при розрахунку потужності послідовне з'єднання двох однакових нагрівачів, дає не чверть номіналу для кожного, як впливає з закону Ома, а третину, при з'єднанні трьох ламп - 1/6 частину номіналу, при чотирьох-1/ 8.

Було встановлено, якщо відстань від ламп до опромінюваної поверхні, дорівнює або більше, ніж 4/3 відстані між осями нагрівачів, нерівномірністю нагрівання можна знехтувати. Це підтверджує тест з прожарювання паперу, при якому ніякої смугастості не спостерігається.

Структура та алгоритм роботи приладу. ІЧ паяльна станція має наступні основні вузли (рис. 2):

1. нижній ІЧ випромінювач – це інфрачервоний випромінювач, що встановлений в нижній частині (в корпусі) станції, для попереднього і рівномірного нагрівання всієї плати до 200-220 °С. Дуже важливо щоб плата була прогріта рівномірно, інакше її може «повести» в результаті чого вона деформується.

2. нижня термopа, яка власне і знімає температуру з низу плати і не дає їй перегрітися, даючи автоматиці сигнал на відключення живлення.

3. верхній кварцовий інфрачервоний випромінювач, який нагріває локальне місце на платі для розпаювання і подальшої пайки.

4. верхня термopapa, яка контролює температуру верхнього випромінювача в момент розпаювання.

5. контролюючий модуль на базі Arduino.

6. апаратно-виконавчу частину (система комутації обладнання та система охолодження)

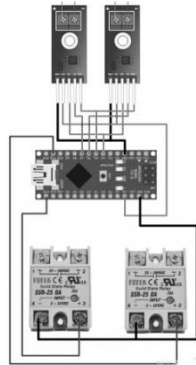


Рисунок 2 – Структурна схема приладу.

Основну частину управління приладом становить автомат з двома датчиками (в їх якості виступають термopapa плати і термopapa чіпу) і двома виконавчими механізмами – нижній та верхній нагрівач, які керуються за допомогою реле.

Було вирішено, що алгоритм регулювання потужності нагріву у автоматичному режимі роботи буде реалізований на ПК, за допомогою мікропроцесорного контролера Arduino [7] (далі - контролер). Контролер забезпечує зв'язок між станцією і ПК. Він циклічно виконує наступні функції: отримує з ПК параметри для забезпечення регулювання температури нагрівачів за допомогою широтно-імпульсної модуляції (ШИМ-регулювання), встановлює необхідну температуру нагрівачів, передає температуру термopар в ПК.

Контролер очікує на USB порту повідомлення типу SETxxx * ууу *, де xxx та ууу – відповідно потужність верхнього та нижнього нагрівача у відсотках. Якщо отримане повідомлення відповідає шаблону, виставляються коефіцієнти, що необхідні для ШИМ-регулювання нагрівачів і повертається повідомлення OKaaabbbccdd, де aaa і bbb – встановлена потужність верхнього та нижнього нагрівачів, ccc і ddd – показники температури, яка отримується від верхньої та нижньої термopари [6].

Апаратне ШИМ-регулювання за допомогою мікроконтролера з частотою дискретизації кілька кілогерц в випадку з твердотілим реле неможливе, тому що твердотіле реле не може бути відключено в довільний момент часу. Тому було вирішено реалізувати власний алгоритм ШИМ-регулювання з частотою близько 5 Гц. Така частота задовольняє вимогам при яких лампи повністю гаснути не встигають. При цьому мінімальний коефіцієнт заповнення в 10% цілком достатній для можливості захопити один період напруги. [6]

При написанні скетч-програми довелося відмовитись від завдання затримок за допомогою функції delay(), оскільки є вірогідність, що під час затримок дані з порту можуть втрачатися. Алгоритм являє собою нескінченний цикл, в якому перевіряється наявність даних з USB порту і контролюється значення лічильників часу в програмі, що реалізує ШИМ-регулювання. Якщо є в наявності коректні дані з порту, вони оброблюються, якщо лічильник часу має значення, що вказує на необхідність перемикання згідно із алгоритмом ШИМ-регулювання, проводяться відповідні дії з включення-вимикання нагрівачів [8].

Алгоритм роботи програми наведено на рисунку 3.

Додаток для комп'ютера. Програма дає можливість відображати стан нагрівачів, створює графік температури і має вбудовану примітивну мову моделювання. Програма складається з набору пар «умова - дія». Наприклад, «при досягненні нижньої термopарою температури 120 градусів встановити рівень потужності: для нижнього нагрівача - 10%, для верхнього - 80%». Така множина умов реалізує необхідний термопрофіль, швидкість нагріву, температуру утримання і т. п. У додатку довжина такту складає 1 секунду і вимірюється спеціальним таймером. Під час кожного такту в контролер відправляються поточні установки потужності та отримуються поточні значення температур нагрівачів. Останні відображаються у вікні параметрів та на графіку, потім викликається процедура перевірки логічних станів і підготовки даних для нового сеансу обміну.

Формування профілів ефективного ІЧ нагріву. Дослідження рівномірності зони теплового поля нагріву верхніх і нижніх ІЧ нагрівачів дають наступну інформацію:

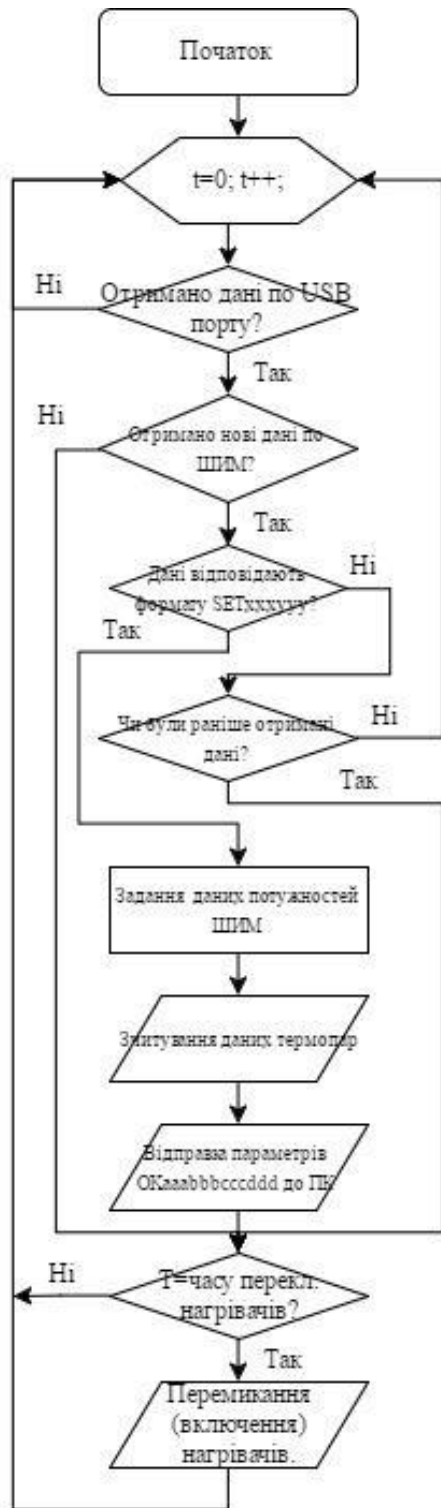


Рисунок 3– Алгоритм роботи контролеру.

Ізотермічні зони: центральна зі швидкістю нагріву $> 5^{\circ}\text{C}/\text{с}$ має форму кола, після зони - овальні, що обумовлено прямокутною формою рефлектора (рис. 4 зліва). На рис. 4 (справа) зображені зони теплового поля ІЧ нагрівача з маскою квадратної форми 4×4 мм. Ізотерми мають квадратну форму зі стороною 6 мм при $V >$

$3^{\circ}\text{C}/\text{с}$, 12 мм при $V = 2,2^{\circ}\text{C}/\text{с}$ і 18 мм при $V < 1^{\circ}\text{C}/\text{с}$. Застосування відражаючих екранів різної форми збільшує локальність нагрівання [8]. Найбільш швидке нагрівання $15^{\circ}\text{C}/\text{с}$ досягається при зменшенні відстані від плати до верхнього ІЧ нагрівача. Застосування якісного рефлектора для ІЧ джерела дозволяє при тій же інтенсивності нагріву вдвічі зменшити споживану потужність і замінити водяне охолодження повітряним. Підвищення локальності ІЧ нагрівання за рахунок світловідбиваючої маски не робить істотного впливу на швидкість росту температури, зменшуючи в цілому тільки нагрів плати [8].

Оптимальні режими ІЧ пайки наступні:

- нагрів нижнім нагрівачем плати до $90-110^{\circ}\text{C}$ зі швидкістю $6-8^{\circ}\text{C}/\text{с}$,
- нагрів верхнім і нижнім нагрівачами до температури пайки зі швидкістю $10-14^{\circ}\text{C}/\text{с}$.

За необхідності локалізації нагріву для термочутливих елементів застосовують світловідбиваючу маску.

Швидкість нагріву верхнього нагрівача потужністю 500 Вт з рефлектором становить $3-5^{\circ}\text{C}/\text{с}$; верхнього і нижнього нагрівача потужністю 500 Вт до $10^{\circ}\text{C}/\text{с}$. Застосування захисної маски підвищує швидкість нагріву до $15^{\circ}\text{C}/\text{с}$ за рахунок відображення, збільшуючи ступінь локалізації нагріву. Подальше збільшення швидкості нагріву можливо за рахунок поліпшення якості параболічних рефлекторів і підвищення їх здатності, що відображає швидкість зростання температури в зоні пайки знижується зі збільшенням відстані від ІЧ нагрівача до плати: для верхнього в 2 (рис. 5) і для нижнього в 3 рази (рис.6) [8].

Режими пайки оплавленням визначається температурним профілем, який оптимізований для ІЧ нагрівачів (рис. 7). Стадія попереднього нагрівання знижує тепловий удар на електронні компоненти і друковані плати. При швидкості нагріву не більше $1-3^{\circ}\text{C}/\text{с}$ відбувається випаровування розчинника з паяльної пасти. Висока швидкість нагріву призводить до передчасного випаровування розчинника і цілої низки дефектів: пошкодження компонентів за рахунок теплового удару, розбризкування кульок припою і виникненню перемичок [9-11]. Отже різниця температур «предподогріву» і оплавлення не повинна перевищувати 100°C .

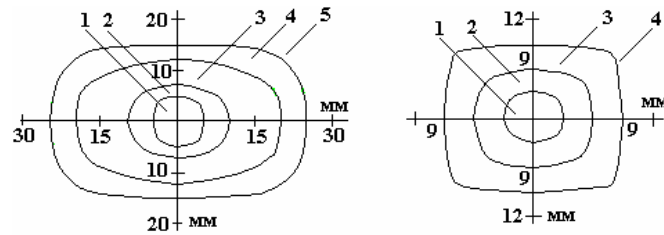


Рисунок 4. Ізотермічні зони для різної швидкості нагріву:
 а): 1 – $>5^{\circ}\text{C}/\text{c}$; 2 – $5^{\circ}\text{C}/\text{c}$; 3 – $3,8^{\circ}\text{C}/\text{c}$; 4 – $3,3^{\circ}\text{C}/\text{c}$; 5 – $<2,2^{\circ}\text{C}/\text{c}$;
 б): 1 – $>4^{\circ}\text{C}/\text{c}$; 2 – $2,2^{\circ}\text{C}/\text{c}$; 3 – $1,1^{\circ}\text{C}/\text{c}$; 4 – $<0,5^{\circ}\text{C}/\text{c}$

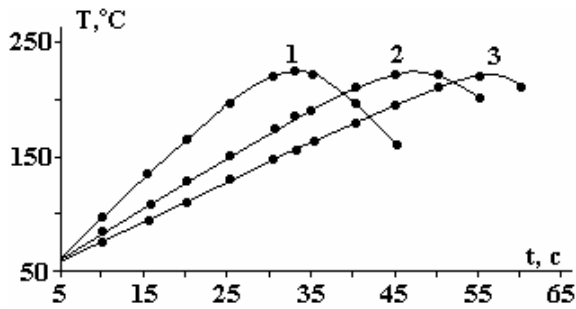


Рисунок 5 – Температурні залежності в зоні нагріву верхнього ІЧ випромінювача при відстанях до плати, мм: 1 - 15, 2 - 25, 3 - 35

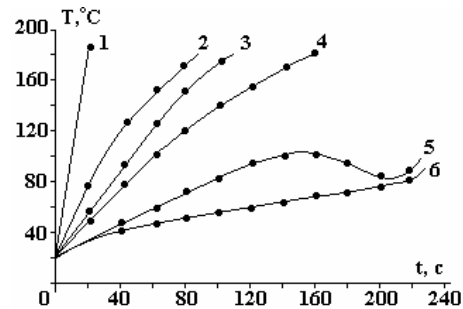


Рисунок 6 – Температурні залежності в зоні нагріву нижнього ІЧ випромінювача при відстанях до плати, мм: 1 - 10, 2 - 20, 3 - 30, 4 - 40, 5 - 50, 6 - 60

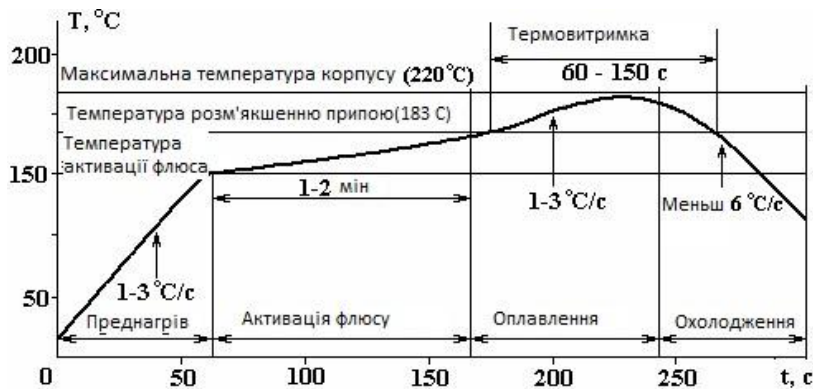


Рисунок 7 – Оптимальний температурний профіль пайки ІЧ нагріванням для олов'яно-свинцевих припоїв

Висновки. Застосування інфрачервоної пайки є перспективним напрямком в технології поверхневого монтажу, який забезпечує зменшення витрат на експлуатацію обладнання при одночасному підвищенні якості паяних з'єднань. ІЧ нагрівання вигідно відрізняється тим, що має більш просте устаткування, яке набагато економніше і доцільніше для поверхневого монтажу виробів.

На підставі аналізу причин виникнення дефектів поверхневого монтажу, зроблено висновок, що однією з причин їх утворення є недосконалість температурного профілю пайки. Вирішення цих проблем пропонується за допомогою сучасних приладів, що

забезпечують повну автоматизацію процесу пайки, зводячи нанівець людський фактор. Важливою перевагою запропонованої ІЧ станції також являється дешева вартість комплектуючих, із збереженням якості пайки в порівнянні із серійними приладами вартістю в тисячі доларів. Розроблене програмне забезпечення, що забезпечує дистанційне управління, робить процес ще простішим і, в кінцевому підсумку, не вимагає високих навичок оператора.

Література

- [1] Инфракрасное излучение. [электронный ресурс]. – Режим доступа: https://wikipedia.org/wiki/Инфракрасное_излучение
- [2] Как паяют «мосты» и чипы на материнских платах с помощью паяльной станции [электронный ресурс]. – Режим доступа: <https://habrahabr.ru/company/muk/blog/271515/>
- [3] Хельге Шимански (Helge Schimanski) Квалифицированные процессы ремонта на основе актуальных норм и стандартов I601 // Технологии в электронной промышленности. 2010
- [4] Валлин Л. Использование стандартов IPC на всех этапах производства электроники. Стандарты серии IPC-2220 и IPC-7351A // Технологии в электронной промышленности. 2010. № 1.
- [5] Валлин Л. Стандарт IPC-4101C и серия IPC-6010 // Технологии в электронной промышленности. 2010. № 2. 3.
- [6] Валлин Л. Стандарт и тренинг IPC-A-600H // Технологии в электронной промышленности. 2010. № 6.
- [7] Arduino-Nano [электронный ресурс]. – Режим доступа: <http://arduino.cc/en/Main/ArduinoBoardNano/>.
- [8] Хельге Шимански (Helge Schimanski) Квалифицированные процессы ремонта на основе актуальных норм и стандартов I601 // Технологии в электронной промышленности. 2010
- [9] Патрик Церпер NanoFlux — Флюс с нанохимически активными металлическими соединениями для дисперсионной стабилизации мягких припоев.
- [10] Максим Шмаков, Александр Тиханкин Оптимизация температурного профиля пайки оплавлением // Технологии в электронной промышленности, 2008.
- [11] Якушенков Ю. Г. Инфракрасные системы нагревательного типа. М., 2015.

References

- [1] Infrared radiation. [ONLINE]. - Available at: https://wikipedia.org/wiki/Инфракрасное_излучение
- [2] How to solder “bridges” and chips on motherboards using a soldering station [ONLINE]. - Available at: <https://habrahabr.ru/company/muk/blog/271515/>
- [3] Helge Schimanski. Qualified repair processes based on current norms and standards I601 // Technologies in the electronics industry. 2010
- [4] Vallin L. Use of IPC standards at all stages of electronics manufacturing. IPC-2220 and IPC-7351A Series Standards // Technology in the Electronics Industry. 2010. № 1.
- [5] Vallin L. IPC-4101C Standard and IPC-6010 Series // Technology in the Electronics Industry. 2010. № 2. 3.
- [6] Vallin L. Standard and Training IPC-A-600H // Technology in the Electronics Industry. 2010. № 6.
- [7] Arduino-Nano [ONLINE]. - Available at: <http://arduino.cc/en/Main/ArduinoBoardNano/>.
- [8] Helge Schimanski. Qualified repair processes based on current norms and standards I601 // Technologies in the electronics industry. 2010
- [9] Patrik TSerrer NanoFlux — Flux with nanochemically active metal compounds for dispersion stabilization of soft solders.
- [10] Maksim SHmakov, Aleksandr Tikhankin. Optimization of the temperature profile of reflow soldering // Technologies in the electronic industry, 2008.
- [11] YAkushenkov YU. G. Infrared heating systems. M., 2015.

Надійшла до редакції 05.01.2020 р.

Зайцев Кирило Сергійович – магістрант, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: hardworkerhappy@gmail.com.

Шамасєв Віталій Віталійович – к. т. н., доцент, Державний вищий навчальний заклад «Донецький національний технічний університет» (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).

E-mail: vitalii.shamaev@donntu.edu.ua

ANALYSIS AND DEVELOPMENT OF ALTERNATIVE INFRARED-SOLDER SOLUTIONS

The paper describes the development of alternative solutions for infrared soldering and control devices for this process, proposes a device that will meet the modern requirements of infrared soldering, have a convenient and technological hardware, a friendly interface with the ability to monitor and control the characteristics of the soldering process from the computer. The modern soldering station should provide the following basic functions: soldering of chips up to 60x60mm in size, possibility of soldering plastic connectors, temperature control through analysis of thermocouple signals; automatic and manual modes; ability to monitor the process using a computer, setting thermal profiles for soldering with different types of solders, the ability to change them and add new ones. By type of heating element, soldering stations are divided into stations with ceramic and quartz elements. The advantage of ceramic heaters is the soldering by means of radiation of electromagnetic waves of the invisible spectrum, which are absolutely safe for human vision and allow the operator to make visual observation of the process. Quartz heaters have less inertia and provide greater homogeneity of the heating zone, although they use a range of infrared radiation that is hazardous to humans. The optimum soldering mode using a quartz heater is determined by the temperature profile, which can significantly reduce the hazardous factors of the process and get a quality result. The use of infrared soldering is a perspective trend in surface mount technology, which reduces the cost of installing and operating equipment while improving the quality of solder joints. Designed software that allows remote control makes the soldering process easier and ultimately reduces operator skills requirements.

Keywords: *infrared radiation, infrared soldering station, Arduiono, automated soldering profiles.*

Zaitsev Kyrylo Serhiiiovych – undergraduate, State Higher Education Establishment “Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: hardworkerhappy@gmail.com.

Shamaiev Vitalii Vitaliiiovych – PhD, Associate Professor, State Higher Education Establishment “Donetsk National Technical University”, (Shybankova square, 2, Pokrovsk, Donetsk region, 85300, Ukraine).

E-mail: vitalii.shamaev@donntu.edu.ua

ВИКОРИСТАННЯ ОДНОПЛАТНИХ КОМП'ЮТЕРІВ ДЛЯ МОДЕЛЮВАННЯ ПРОЦЕСІВ ШАХТНИХ ВЕНТИЛЯЦІЙНИХ МЕРЕЖ

Новіков В. Г.

Виконано аналіз одноплатних комп'ютерів на базі Ordroid XU4. Зроблено портування програмного забезпечення (ПЗ) моделювання динамічних процесів шахтних вентиляційних мереж (ШВМ) на одноплатний комп'ютер. ПЗ має 4 режими моделювання: для одного елементу ШВМ, для повітряної секції з чотирьох елементів, для газово-повітряної секції з чотирьох елементів та для чотирьох елементної мережі з динамікою повітряних мас. Виконано відлагодження та модернізування ПЗ на одноплатному комп'ютері. Розглянуто способи збереження отриманих даних з використанням NoSQL баз даних (БД) на прикладі ElasticSearch. Приведено пакетний спосіб передачі інформації до БД з використанням JSON файлів.

Ключові слова: одноплатний комп'ютер, моделювання, шахтні вентиляційні мережі, бази даних, NoSQL, ElasticSearch, JSON.

Вступ. Шахтна вентиляційна система – система з'єднаних між собою підземних виробок шахти, що забезпечує спрямований рух повітря для провітрювання. Включає джерела тяги (вентилятори), вентиляційні регулятори і споруди, шляхи витоку повітря. Рух повітря у вентиляційній мережі підтримується за допомогою напору (або розрідження), що створюється одним або декількома вентиляторами [1].

В сучасних вугільних шахтах вентиляційна мережа містить до 300-500 гілок, в рудних - до 1000. Найпростішим елементом шахтної вентиляційної системи можна вважати гілку, яку можна уявити трубою (рис 1). Основні параметри гілки вентиляційної мережі (аеродинамічний опір – r , витрати повітря – q і втрати тиску – h , в кожній гілці) пов'язані між собою співвідношеннями виду $h = r \cdot q^2$ (гілки з турбулентним рухом повітря) або $h = r \cdot q$ (гілки з ламінарним рухом повітря). Вентиляційна мережа в цілому характеризується наступними параметрами (в разі одного вентилятора головного провітрювання): сумарна витрата повітря – Q ; сумарна втрата тиску – H ; загальний опір – R , які пов'язані рівнянням $H = R \cdot Q^2$, званим характеристикою вентиляційної системи [2].

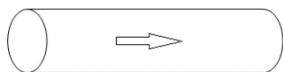


Рисунок 1 – Елемент вентиляційної системи.

Елементи вентиляційної системи можна об'єднати у найпростішу секцію (рис 2а) або мережу (рис 2б).

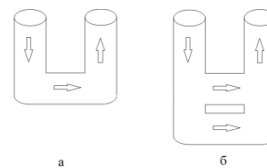


Рисунок 2 – Секція (а) та мережа (б) вентиляційної системи.

Особливою відмінністю секції або мережі є наявність декількох елементів, кожен з яких має свій опір. В кожному з елементів ШВМ можуть знаходитись: перешкоди; елементи зворотної тяги; інший склад повітряних мас.

У якості перешкод з елементами зворотної тяги можуть виступати шахтні машини та агрегати, що використовуються у виробництві, активна система охолодження, яка спрямовує потік повітря на зустріч потоку ШВМ.

Процес зміни руху повітряних мас не може відбуватися миттєво, саме тому, це може призвести до негативних або, навіть, катастрофічних наслідків (недостатність кисню, великий рівень метану тощо). Тому було розроблено додаток, який дозволяє моделювати ці процеси та попереджати надзвичайні ситуації.

Додаток працює на UNIX подібній системі персонального комп'ютера. Та для використання додатку в реальних умовах

необхідно перенести додаток на більш мобільну платформу. Саме тому було обрано використання одноплатних комп'ютерів.

Метою роботи є портування та модернізування існуючого ПЗ на одноплатний комп'ютер.

Задачі роботи:

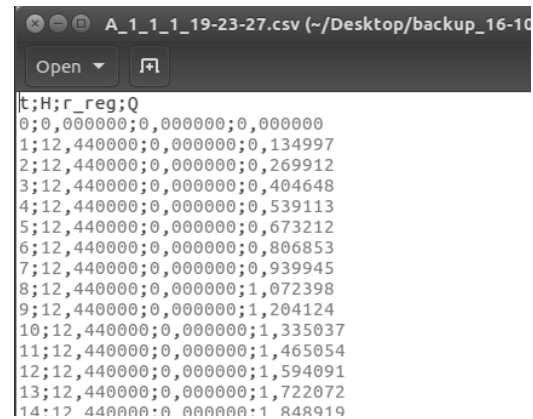
1. Аналіз ПЗ для моделювання динамічних процесів ШБМ;
2. Огляд та налагодження платформи одноплатного комп'ютера ODROID XU4;
3. Портування та налагодження ПЗ для моделювання динамічних процесів ШБМ на ОК ODROID XU4;
4. Розгортання NoSQL бази даних Elasticsearch на платформі ODROID XU4;
5. Реалізація та відлагодження функції запису даних до бази даних;
6. Дослідження продуктивності роботи із БД.

Особливості роботи додатку ПЗ моделювання динамічних процесів ШБМ. Додаток має чотири режими роботи: одиночний елемент з динамікою повітряних мас; чотирьох-елементна секція з динамікою повітряних мас; чотирьох-елементна секція з динамікою газових мас; чотирьох-елементна мережа з динамікою повітряних мас.

Результати роботи ПЗ, на початковому етапі, зберігаються у вигляді csv-файлів. Приклад наповнення результуючого файлу наведено на рис. 3.

Цей файл дуже складний в сприйнятті інформації людиною. Саме тому було вирішено відобразити інформацію у вигляді графіків. Для цього було використано

програмне забезпечення UNIX систем – GNU_plot.



```
lt;H;r_reg;Q
0;0,000000;0,000000
1;12,440000;0,000000
2;12,440000;0,000000
3;12,440000;0,000000
4;12,440000;0,000000
5;12,440000;0,000000
6;12,440000;0,000000
7;12,440000;0,000000
8;12,440000;0,000000
9;12,440000;0,000000
10;12,440000;0,000000
11;12,440000;0,000000
12;12,440000;0,000000
13;12,440000;0,000000
14;12,440000;0,000000
```

Рисунок 3 – Приклад наповнення результуючого файлу ПЗ.

GNU_plot – вільна програма для створення дво- і тривимірних графіків. Вона має власну систему команд, може працювати в режимі командного рядка або виконувати скрипти, які зчитуються з файлів. GNU_plot виводить графіки як безпосередньо на екран, так і в файли різних графічних форматів, таких як PNG, EPS, SVG, JPEG та інших [3].

Для обробки інформації з csv-файлу програмою GNU_plot, необхідно створити скрипт, в якому позначити ім'я файлу, розділовий символ у рядку, тип даних та всі інші параметри, необхідні для побудови графіку.

Результати побудови графіків для чотирьох-елементної мережі наведено на рис. 4.

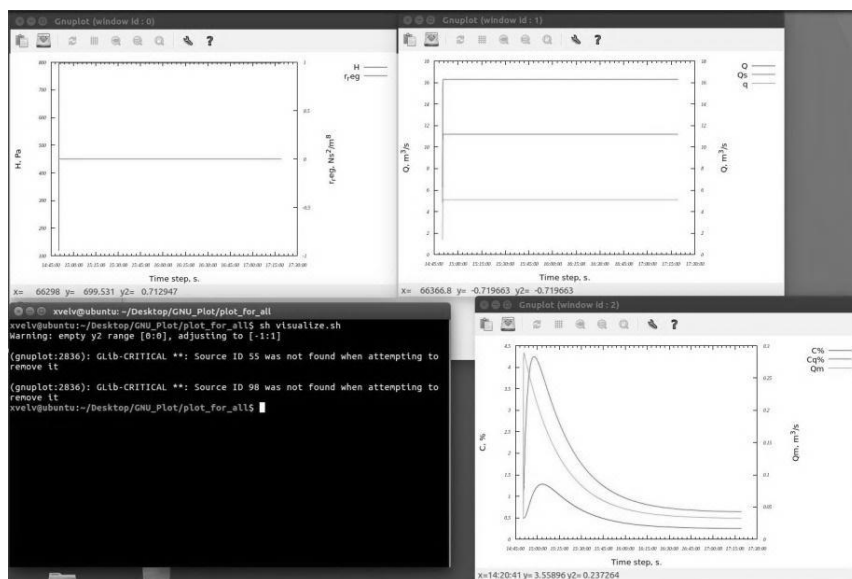


Рисунок 4 – Графіки GNU_plot, виконання четвертої функції ПЗ

Як можна помітити, GNU_plot дозволив візуалізувати отримані данні, та вони стали більш наочними. Також використання GNU_plot дозволяє експортувати графіки у графічні файли для подальшого аналізу або роздрукування.

Огляд та налагодження платформи одноплатного комп'ютера ODDROID XU4. Кінцева платформа, на якій повинно працювати ПЗ – ODDROID XU4. ODDROID – це серія одноплатних комп'ютерів і планшетних ПК, створених компанією Hardkernel Co., Ltd., що розташована в Південній Кореї. Незважаючи на те, що ім'я ODDROID є портмантом Open + Android, обладнання насправді не відкрите, тому що частина дизайну зберігається компанією. ODDROID XU4 (рис. 5) – це нове покоління обчислювальних пристроїв з більш потужним, енергозберігаючим обладнанням та меншим форм-фактором ніж його попередники. Запропонували підтримку з

відкритим вихідним кодом, плата може запускати різні версії Linux, включаючи останню версію Ubuntu 16.04 та Android 4.4 KitKat і 7.1 Nougat. Використовуючи інтерфейси eMMC 5.0, USB 3.0 та Gigabit Ethernet, ODDROID XU4 має чудову швидкість передачі даних, яка потрібна для підтримки високої потужності обробки на пристроях ARM. В табл. 2 представлені технічні характеристики одноплатного комп'ютера ODDROID-XU4 [4].

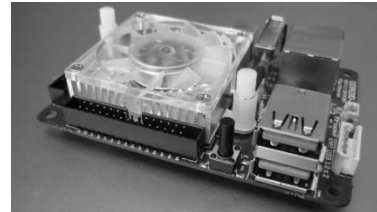


Рисунок 5 – Одноплатний комп'ютер ODDROID XU4

Таблиця 2 – Технічні характеристики ODDROID XU4.

Тип характеристики	Опис
Процесор	Samsung Exynos5 Octa ARM Cortex™-A15 Quad 2Ghz and Cortex™-A7 Quad 1.3GHz CPUs
Пам'ять	2Гб LPDDR3 O3Y на 933МГц (14.9Гб/с пропускна здатність пам'яті)
3D акселератор	Mali-T628 MP6(OpenGL ES 3.0/2.0/1.1 and OpenCL 1.1 Full profile)
Відео	підтримує 1080p через кабель HDMI (контейнер формату H.264 + AAC на основі MP4)
USB3.0 хост	SuperSpeed USB стандарту А x 2 порти, максимальне завантаження: загальний 2Amp для 2-х портів хоста USB 3.0
USB2.0 хост	Високошвидкісний стандартний роз'єм типу X 1 портів, максимальна навантаження: 500 мА/порт
Wi-Fi	USB IEEE 802.11b / g / n 1T1R WLAN з антеною (USB-модуль)

Портування та налагодження ПЗ моделювання динамічних процесів ШВМ на ОК ODDROID XU4. ПЗ моделювання динамічних процесів ШВМ на початковому етапі було розроблено для виконання на UNIX-подібних системах. На платформі ОК ODDROID XU4 встановлено саме таку систему. Тому, складності портування програмного забезпечення були пов'язані з використанням іншої версії компілятора GPP. При портуванні виникли незначні помилки компілятора, в більшості пов'язані з різницею бібліотек або типу змінних. Різницю в архітектурі було вирішено завдяки інтегрованим засобам компілятора, а всі інші

незначні помилки вирішувались за мірою їх виникнення.

При налагодженні програмного забезпечення було поставлено декілька додаткових задач:

1. виконати експорт до результуючого файлу часу у форматі HH:mm:ss, відштовхуючись від системного часу;
2. виконати експорт до результуючого файлу часу, відштовхуючись від заданого часу;
3. підрахувати час виконання кожного експерименту для кожної функції ПЗ.

Розгортання NoSQL бази даних ElasticSearch на платформі ODDROID XU4. Оскільки збереження інформації у файлах не

є найкращим рішенням, було прийнято рішення використовувати для цього БД. Проаналізувавши існуючі рішення було обрано Elasticsearch [5], перевагами якого є: безкоштовність; швидкість роботи; легка масштабованість; висока відмовостійкість; велика кількість типів даних.

ElasticSearch - це платформа для розподіленого пошуку та аналізу даних в режимі реального часу. Вона користується великою популярністю завдяки простоті, потужності і масштабованості. Платформа підтримує операції REST, що дозволяє застосовувати методи HTTP (GET, POST, PUT, DELETE тощо). ElasticSearch є вільним і відкритим програмним забезпеченням відомої компанії Elastic. Тому ElasticSearch відмінно підходить для використання в самих різних умовах [6]. Продукт можна завантажити з сайту проекту в форматі zip, tar.gz, deb і rpm. В системі Ubuntu краще використовувати пакет deb, який встановить все необхідне для коректної роботи ElasticSearch.

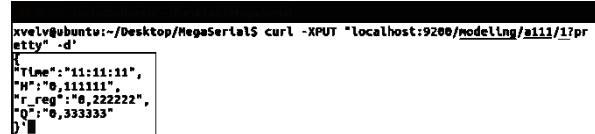
Файли Elasticsearch знаходяться в каталозі /etc/elasticsearch.

Файл elasticsearch.yml містить серверні налаштування ElasticSearch. Тут зберігаються всі параметри програми. У цьому файлі потрібно відредагувати змінні node.name і cluster.name. Змінна node.name задає ім'я сервера, а cluster.name – ім'я кластера, до якого належить сервер.

Значення cluster.name використовується функцією автоматичного виявлення і підключення серверів Elasticsearch в кластер. Таким чином, значення за замовчуванням допускає в кластер небажані сервери, знайдені в цій мережі. Тому потрібно відредагувати даний файл в текстовому редакторі та змінити два параметри: cluster.name та node.name. Після цього ElasticSearch можна вважати налаштованим та готовим до роботи [7].

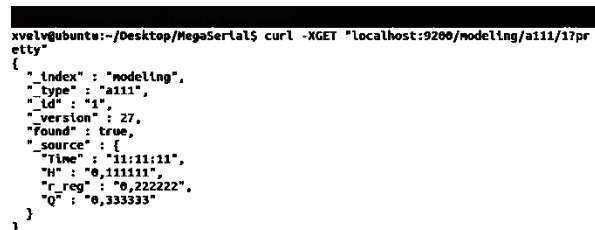
Реалізація та відлагодження функції запису даних до бази даних. Щоб записати до бази даних в таблицю під індексом даних необхідно виконати команду, наведену на рис. 6 [8].

Для того, щоб отримати данні за заданим записом від ElasticSearch необхідно виконати команду, наведену на рис. 7. Відповідь також зображено на цьому рисунку.



```
xvelv@ubuntu:~/Desktop/MegaSerial5$ curl -XPUT "localhost:9200/modeling/a111/17pretty" -d '{
  "time": "11:11:11",
  "H": "0,111111",
  "r_reg": "0,222222",
  "Q": "0,333333"
}'
```

Рисунок 6 – Команда запису інформації до ElasticSearch.



```
xvelv@ubuntu:~/Desktop/MegaSerial5$ curl -XGET "localhost:9200/modeling/a111/17pretty"
{
  "index": "modeling",
  "type": "a111",
  "_id": "1",
  "_version": 27,
  "found": true,
  "_source": {
    "time": "11:11:11",
    "H": "0,111111",
    "r_reg": "0,222222",
    "Q": "0,333333"
  }
}
```

Рисунок 7 – Команда отримання даних від ElasticSearch та її відповідь.

Вигляд, у якому зберігаються данні в ElasticSearch:

- «_index» – назва бази даних;
- «_type» – назва таблиці;
- «_id» – ідентифікатор запису;
- «_version» – яка версія інформації зберігається за даним ідентифікатором;
- «found» – признак того, що інформація знайдена;
- «_source» – інформація, що зберігається.

Для програмного збереження інформації у ElasticSearch спочатку було прийнято рішення використовувати саме цей підхід.

Але після тестування отриманого додатку, було виявлено, що після налаштування послідовного (запис за записом) додавання інформації до ElasticSearch на віртуальній машині, час виконання програми зріс більше ніж у 30 разів. Результати зображені на рис. 8.

Через це було обрано додавання інформації до бази даних за допомогою JSON файлів.

JSON - текстовий формат обміну даними, заснований на JavaScript. Як і багато інших текстових форматів, JSON легко може бути прочитаним людиною. Незважаючи на походження від JavaScript, формат вважається незалежним від мови і може використовуватися практично з будь-якою мовою програмування. Для багатьох мов існує готовий код для створення і обробки даних в форматі JSON [9].

```
AVC@ubuntu:~/Desktop/Serial (copy)$ ls -l\nrm: cannot remove '*.csv': No such file or directory\nPlease choose the benchmark to execute.
```

Available options:

1. A single Element with air dynamics (the simplest)
2. A 4-elements Section with air and air dynamics (more complex)
3. A 4-elements Section with air and gas dynamics (more complex)
4. A 4-elements Network with air dynamics

Make you choice now and press enter: 1
Created Air Element: Roadway; l=500, F=7, r=0.24, ro=1.29, K=92.1429

RUN WITHOUT ARGS

Element Roadway: Applied initial ctrl set: H=0, r_reg=0

Element Roadway: Applied ctrl set: H=12.44, r_reg=0

Element Roadway: Applied initial ctrl set: H=12.44, r_reg=0

Element Roadway: Applied ctrl set: H=12.44, r_reg=4.5

Element Roadway: Applied initial ctrl set: H=12.44, r_reg=4.5

Element Roadway: Applied ctrl set: H=12.44, r_reg=2.5

Time of the First experiment: 4ms

Time of the Second experiment: 1ms

Time of the Third experiment: 1ms

Time of the All experiments: 7ms

Simulation completed

xvel@ubuntu:~/Desktop/Serial (copy)\$

```
{ "index": "modeling", "type": "a111", "id": "517", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "518", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "519", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "520", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "521", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "522", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "523", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "524", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "525", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "526", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "527", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "528", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "529", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "530", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "531", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "532", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "533", "version": 2, "created": false }\n{ "index": "modeling", "type": "a111", "id": "534", "version": 2, "created": false }\nTime of the First experiment: 142ms\nTime of the Second experiment: 33ms\nTime of the Third experiment: 44ms\nTime of the All experiments: 220ms\nSimulation completed\nxvel@ubuntu:~/Desktop/Serial$
```

Рисунок 8 – Порівняння часу виконання ПЗ з послідовним занесенням до Elasticsearch та без занесення до БД

При виконанні ПЗ інформація, що була підрахована на теперішньому етапі, експортується до csv-файлу, як це було на початковому етапі, та до JSON файлу в необхідному форматі. Виходячи з цього, кількість записів у JSON файлі така ж, як і в csv-файлі та дорівнює кількості секунд у вказаному для моделювання часу. Перед завершенням роботи ПЗ виконується запит до Elasticsearch для передачі сформованого файлу.

Одним з стандартних структур файлів JSON є набір пар «ключ : значення». Саме ця структура використана при створенні файлу програмним забезпеченням (рис. 9) [10].

Дослідження продуктивності роботи із БД. Наступним етапом було зроблено порівняльну характеристику швидкодії ПЗ. Експеримент проводився з та без занесення до БД. Результати наведено в мілісекундах. Першим експериментом було порівняння роботи на ВМ та на платформі одноплатного комп'ютера моделювання першої функції ПЗ. Заданий час моделювання першої функції – 1 година, результати наведено на рис. 10.

Також порівняння часу моделювання було виконано для секції з повітрям на п'ятнадцять хвилин, для секції з повітрям та газом на шість з половиною годин та для

мережі на три години вперед. Результати представлені на рис. 11 – 13 відповідно.

```
{ "index": "modeling", "type": "a111", "id": "1" }\n{ "Time": "18:10:44", "H": "0,000000", "r_reg": "0,000000", "Q": "0,000000" }\n{ "index": "modeling", "type": "a111", "id": "2" }\n{ "Time": "18:10:45", "H": "12,440000", "r_reg": "0,000000", "Q": "0,134997" }\n{ "index": "modeling", "type": "a111", "id": "3" }\n{ "Time": "18:10:46", "H": "12,440000", "r_reg": "0,000000", "Q": "0,269912" }\n{ "index": "modeling", "type": "a111", "id": "4" }\n{ "Time": "18:10:47", "H": "12,440000", "r_reg": "0,000000", "Q": "0,404648" }\n{ "index": "modeling", "type": "a111", "id": "5" }\n{ "Time": "18:10:48", "H": "12,440000", "r_reg": "0,000000", "Q": "0,539113" }\n{ "index": "modeling", "type": "a111", "id": "6" }\n{ "Time": "18:10:49", "H": "12,440000", "r_reg": "0,000000", "Q": "0,673212" }\n{ "index": "modeling", "type": "a111", "id": "7" }\n{ "Time": "18:10:50", "H": "12,440000", "r_reg": "0,000000", "Q": "0,806853" }\n{ "index": "modeling", "type": "a111", "id": "8" }\n{ "Time": "18:10:51", "H": "12,440000", "r_reg": "0,000000", "Q": "0,939945" }\n{ "index": "modeling", "type": "a111", "id": "9" }\n{ "Time": "18:10:52", "H": "12,440000", "r_reg": "0,000000", "Q": "1,072398" }\n{ "index": "modeling", "type": "a111", "id": "10" }\n{ "Time": "18:10:53", "H": "12,440000", "r_reg": "0,000000", "Q": "1,204124" }\n{ "index": "modeling", "type": "a111", "id": "11" }
```

Рисунок 9 – Наповнення файлу JSON

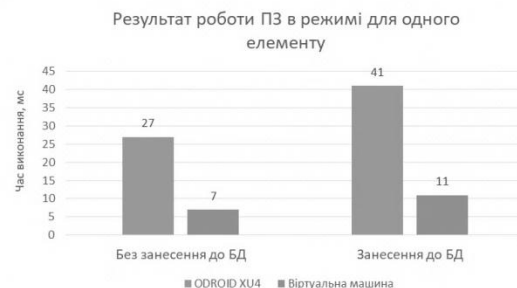


Рисунок 10 – Час виконання ПЗ для моделювання одного елементу

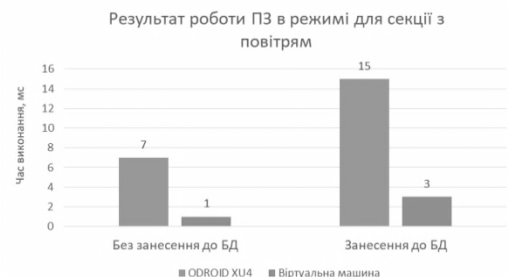


Рисунок 11 – Час виконання ПЗ для моделювання секції з повітрям

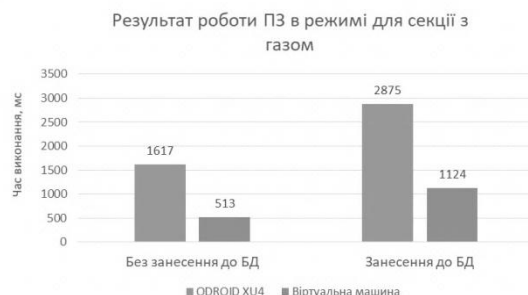


Рисунок 12 – Час виконання ПЗ для моделювання секції з повітрям та газом

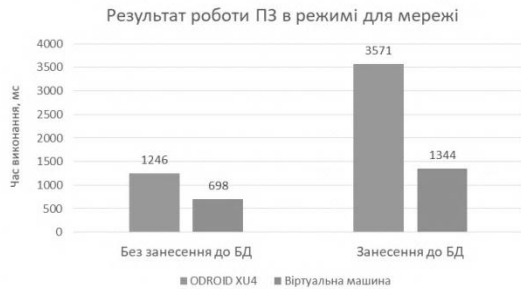


Рисунок 13 – Час виконання ПЗ для моделювання мережі

В загальному випадку різниця у часі з занесенням до БД та без становить 1,5 – 2 рази, а виконання ПЗ на різних платформах від 2 до 7 разів. При цьому найбільша різниця в режимі без занесення до бази даних для секції з повітрям.

Висновок. Результати портування ПЗ моделювання процесів ШВМ наступні: програмне забезпечення працює стабільно;

час моделювання може вважатися прийнятним (наприклад, моделювання секції з повітрям на 15 хвилин вперед виконується за 15 мілісекунд з занесенням до БД).

Збережена у БД інформація може бути в подальшому використана для більш детального аналізу та більш тонкого налаштування ПЗ.

Також слід зазначити що дана тема має актуальні напрямки розвитку, такі як:

- використання спільного серверу збереження даних з багатьох пристроїв;
- прогнозування в режимі реального часу в залежності від даних, отриманих одноплатним комп'ютером з датчиків;
- відображення графіків в реальному часі даних, що зберігаються на сервері, за допомогою інструменту Kibana;
- розпаралелювання ПЗ для підвищення швидкості роботи.

Література

- [1] Шахтна вентиляційна система: пат. 21781 Україна: опубл. 15.06.2001, 2с.
- [2] Аерологія шахтних вентиляційних мереж / В.О. Трофимов та інші. Донецьк: Норд-Прес, 2009. 87 с.
- [3] Gnuplot - Вікіпедія [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Gnuplot>.
- [4] Odroid-XU4 - ODROID Wiki [електронний ресурс]. – Режим доступу: <https://wiki.odroid.com/odroid-xu4/odroid-xu4>.
- [5] Elasticsearch: The Official Distributed Search & Analytics [електронний ресурс]. – Режим доступу: <https://www.elastic.co/products/elasticsearch>.
- [6] Elasticsearch — Вікіпедія [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Elasticsearch>.
- [7] Основы Elasticsearch / Хабр - Habr [електронний ресурс]. – Режим доступу: <https://habr.com/ru/post/280488/>.
- [8] 10 Elasticsearch Concepts You Need to Learn | Logz.io [електронний ресурс]. – Режим доступу: <https://logz.io/blog/10-elasticsearch-concepts/>.
- [9] JSON — Вікіпедія [електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/JSON>.
- [10] Введение в JSON - Stas Bagretsov - Medium | Logz.io [електронний ресурс]. – Режим доступу: <https://medium.com/@stason>.

References

- [1] Mine ventilation system: patent 21781 Ukraine: published. 15.06.2001, 2c.
- [2] Aerology of mine ventilation networks / V.O. Trofimov et al. Donetsk: Nord Press, 2009. 87 c.
- [3] Gnuplot - Wikipedia [ONLINE]. - Available at: <https://ru.wikipedia.org/wiki/Gnuplot>.
- [4] Odroid-XU4 - ODROID Wiki [ONLINE]. - Available at: <https://wiki.odroid.com/odroid-xu4/odroid-xu4>.
- [5] Elasticsearch: The Official Distributed Search & Analytics [ONLINE]. - Available at: <https://www.elastic.co/products/elasticsearch>.
- [6] Elasticsearch — Wikipedia [ONLINE]. - Available at: <https://ru.wikipedia.org/wiki/Elasticsearch>.

- [7] Elasticsearch Fundamentals / Habr [ONLINE]. - Available at: <https://habr.com/ru/post/280488/>.
- [8] 10 Elasticsearch Concepts You Need to Learn | Logz.io [ONLINE]. - Available at: <https://logz.io/blog/10-elasticsearch-concepts/>.
- [9] JSON — Wikipedia [ONLINE]. - Available at: <https://ru.wikipedia.org/wiki/JSON>.
- [10] Introduction to JSON - Stas Bagretsov - Medium | Logz.io [ONLINE]. - Available at: <https://medium.com/@stason>.

Надійшла до редакції 21.01.2020 р.

Новіков Віталій Геннадійович – аспірант, Донецький національний технічний університет (пл. Шибанкова, 2, м. Покровськ, 85300, Україна).
E-mail: vitalii.novikov@donntu.edu.ua.

USE OF SINGLE-BOARD COMPUTERS FOR SIMULATION OF MINE VENTILATION NETWORKS

Mine ventilation system is a system of interconnected underground workings of the mine that provides directional air movement for ventilation. The ventilation network of modern coal mines contains up to 300-500 branches, in the ore mines - up to 1000. The process of changing the air masses movement in such systems is quite complicated and can lead to negative or even catastrophic consequences (lack of oxygen, high levels of methane, etc.). Therefore, an software application was developed to simulate ventilation processes and prevent emergencies. The application runs on a UNIX-like personal computer system. But in order to use the application in real conditions, you need to move the application to a more mobile platform. The use of single-board computers was chosen and the task of appropriate porting and upgrading of existing software. An analysis of single board computers based on Ordroid XU4 has been performed. Porting software for simulating dynamic processes of mine ventilation networks (MVN) on a single board computer is made. The software has 4 simulation modes: for one element of the MVN, for the air section of four elements, for the gas-air section of four elements, and for the four element network with the calculation of air masses dynamics. Debugging and upgrading of the software on a single board computer has been completed. Porting errors have compiler errors, most of which are related to the difference in libraries or type of variables. Architectural differences were solved with the help of integrated compiler tools. The following tasks were solved under debugging: exporting to the resulting time file in HH: mm: ss format, starting from both the system time and the specified time; calculating the execution time of each experiment for each software function. The methods of the obtained data storing using NoSQL databases (DB) on the example of ElasticSearch are considered. Studies of different versions of software implementation showed that in the general case, the speed of storing the received data with the entry in the database is more than 1.5 - 2 times compared with the method without recording. A batch method of transferring information to a database using JSON files is provided.

Keywords: single board computer, simulation, mine ventilation networks, databases, NoSQL, ElasticSearch, JSON.

Novikov Vitalii Hennadiiovych – PhD-student, Donetsk National Technical University, (2, Shybankova square, Pokrovsk, Donetsk region, 85300, Ukraine).
E-mail: vitalii.novikov@donntu.edu.ua.

Наукове видання

Наукові праці Донецького національного технічного університету

Серія «Проблеми моделювання та автоматизації проектування»

Всеукраїнський науковий збірник

№ 1(15), 2020

(українською, російською, англійською, німецькою мовами)

Відповідальний за випуск: В. А. Святний

Технічний редактор: К. С. Гайдук

Підписано до друку 02.02.2020.

Формат 60 x 84 1/16. Гарнітура «Таймс»

Папір офсетний. Друк офсетний.

Умов. друк. арк. 5,3. Облік.- вид. арк. 3,1

Тираж 300 прим. Замовлення № 11/21

Адреса редакції: ДВНЗ «ДонНТУ», Україна, 85300, м. Покровськ, пл. Шибанкова, 2.

Тел.: (062) 392-03-09. E-mail: pmap@donntu.edu.ua

<http://pmap.donntu.edu.ua>

Видавець та виготовлювач: Державний вищий навчальний заклад

«Донецький національний технічний університет».

Україна, 85300, м. Покровськ, пл. Шибанкова, 2. Тел.: (062) 392-03-09

Свідоцтво про державну реєстрацію суб'єкта видавничої справи:

серія ДК № 4911 від 09.06.2015 р.