

Apriori-Analyse von virtuellen parallelen Simulationsmodellen

Автор: Prof. Dr.-Ing. V.A. Svjatnyj

Источник: Virtuelle parallele Simulationsmodelle und ein Devirtualisierungsvorgang der Entwicklung von parallelen Simulatoren für dynamische Netzobjekte mit verteilten Parametern. Vortrag am HLRS, Universität Stuttgart – 2008

Inhaltsverzeichnis

Vorwort

1. Eine Einführung

2. Die Parallelisierungsansätze und virtuelle Parallelitätsebene

3. Parallelitätsebene 1: eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung; Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches; Schema der Verbindung zwischen den virtuellen Prozessen, virtuelles Verbindungsnetzwerk; eine Möglichkeit der Zuordnung „Prozess – Prozessor“

4. Parallelitätsebene 2: eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung; Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches; Schema der Verbindung zwischen den virtuellen Prozessen, virtuelles Verbindungsnetzwerk; eine Möglichkeit der Zuordnung „Prozess – Prozessor“

5. Parallelitätsebene 3: eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung; Tabelle TABLAST; Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches; Schema der Verbindung zwischen den virtuellen Prozessen, virtuelles Verbindungsnetzwerk; eine Möglichkeit der Zuordnung „Prozess – Prozessor“

6. Parallelitätsebene 4: eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung; Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches; Schema der Verbindung zwischen den virtuellen Prozessen, virtuelles Verbindungsnetzwerk; eine Möglichkeit der Zuordnung „Prozess – Prozessor“.

Zusammenfassung und Ausblick

Vorwort

Im Rahmen der Vereinbarung über wissenschaftliche Zusammenarbeit zwischen dem HLRS Universität Stuttgart und der Fakultät für Rechentechik und Informatik (FRTI) der Nationalen technischen Universität Donezk (DonNTU) wird das Forschungsthema „Virtuelle parallele Simulationsmodelle und ein Devirtualisierungsvorgang der Entwicklung von parallelen Simulatoren für dynamische Netzobjekte mit verteilten Parametern“ bearbeitet. Während der Gastprofessur vom 01.03.06 bis zum 30.04.06 hat Prof. V. Svjatnyj drei Teile dieses Modellierungs- und Simulationsproblems als Vorträge am HLRS-Seminar präsentiert. Die Untersuchungen werden fortgesetzt, die verschiedene Aspekte des Problems wurden an der FRTI während des Besuches von Prof. M. Resch (Okt. 2006), während der Gastprofessur von Prof. V. Svjatnyj (April-Mai 2007) sowie am Forschungsseminar FRTI-HLRS in Laspi

(Krym, Sept. 2007) besprochen und diskutiert, die laufende Ergebnisse wurden im gemeinsamen Vortrag bei ASIM-Symposium Simulationstechnik (Sept. 2006, Hannover) vorgestellt.

Vorliegender Bericht informiert über Ergebnisse des wichtigen Bestandteils des Devirtualisierungsvorganges von virtuellen parallelen Simulationsmodelle – von Apriori-Analyse der vier vorgeschlagenen Parallelitätsebenen der dynamischen Netzobjekte mit verteilten Parametern. Die Analyse wurde von Prof. V.A. Svjatnyj während des Forschungsaufenthaltes am HLRS vom 03.03.08 bis zum 30.04.08 durchgeführt und als 4. Teil des obigen Forschungsvorhabens dargestellt.

Die erhaltene Ergebnisse dienen als ein Grund für systematische Implementierung der parallelen Simulatoren auf vorhandenen Zielrechnerarchitekturen. Die Implementierungen sind begonnen und werden im Rahmen der Magister- und Promotionsarbeiten durchgeführt. Die vielseitige Simulationsexperimente sind nach die Prüfung und Bestätigung von Apriori-Analyse zielgerichtet und werden als 5. Teil des Forschungsvorhabens in 2008-2009 geplant.

Berichtsstoff gehört thematisch zum 10. Kapitel „Eine Devirtualisierung von virtuellen parallelen Simulationsmodellen der Netzobjekte mit verteilten Parametern“ des entwickelten gesamten Forschungsberichtes und folgt den Paragraphen 10.1 „Definitionen“ und 10.2 „Etappen der Devirtualisierung“ nach (s. Anlage 1).

1. Eine Einführung

Die Definition von minimalkörnigen Prozessen im Zusammenhang mit der Dekomposition und der Ortsdiskretisierung des DNOVP führt zu dem virtuellen parallelen DNOVP-Simulationsmodell, das die folgende Menge der luftdynamischen Q- und P-Gleichungen (Prozesse) beinhaltet:

$$N_{QGI} = M_1 + M_2 + \dots + M_m = \sum M_i, i = 1, 2, \dots, m;$$

$$N_{PGI} = n + \sum (M_i - 1),$$

wo m die Kantenmenge, n die Knotenmenge und M_i die Menge der diskreten Elementen von Kanten des DNOVP-Graphen sind. Es wird M_i abhängig von den Kantenlängen l_i und von der ausgewählten Ortsschrittweite Δx berechnet:

$$M_i = l_i / \Delta x$$

Dabei wird dieselbe Ortsschrittweite für alle DNOVP-Kanten ausgewählt. Für die Untersuchungszwecke werden verschiedene Δx -Werte geplant (z. B. 30, 20, 10, 5, 2 Meter). Eine dialogunterstützte Erstellung der virtuellen parallelen Simulationsmodelle sieht die entsprechende Parameternberechnungen vor.

Die Ziele der Apriorianalyse werden von uns so formuliert:

- ausgehend von minimalkörnigen Prozessen mögliche Ansätze zur Parallelisierung und die strukturelle Darstellungen der DNOVP-Parallelitätsebenen zu betrachten;
- Die Parallelitätsebene nach Kriterien Lastverteilung, Datenaustausch, Synchronisation der Prozesse, Rechenaufwand zu prüfen;
- Mögliche alternative Lösungen der Organisation von Parallelitätsebenen vorzuschlagen und zu prüfen;
- Zusammenfassender Vergleich der Parallelitätsebenen durchzuführen und Empfehlungen für die Implementierungsarbeiten auf möglichen ZRA zu formulieren.

2. Die Parallelisierungsansätze und die virtuelle Parallelitätsebenen

Die luftdynamische Prozesse in den DNOVP-Kanten und -Knoten laufen gleichzeitig und stellen deshalb Objekte mit der inneren Parallelität und der engen Verbindung über die Knotenrandbedingungen dar. Ausgehend von der oben definierten minimalkörnigen Prozesse betrachten wir die mögliche Parallelisierungsansätze und die daraus entstehende Parallelitätsebenen der virtuellen parallelen Simulationsmodelle.

1. Ansatz: Nach der Ortsdiskretisierung haben wir obige $\sum M_i$ elementaren Luftwege mit den $n + \sum (M_i - 1)$ Knotenrandbedingungen erhalten. In dem einfachen Fall werden diese Luftwege mit den $\sum M_i$ Q-Gleichungen, $\sum (M_i - 1)$ P-Gleichungen für die innere Stützpunkte der Kanten und mit den n P_U -Gleichungen für die Randknoten der Kanten charakterisiert. Zwischen den n P_U -Prozesse sollen wir die äußere Randbedingungen (Anschlüsse zur Atmosphäre und zu den Ventilatoren) unterscheiden. Diese Randbedingungen werden vom Gleichungsgenerator berücksichtigt. Ordnen wir den Q- und P-, P_U -Gleichungen entsprechende Q-, P- und P_U -Prozesse (1 Gleichung – 1 Prozess) zu. Dieser Ansatz verursacht maximal mögliche Menge von minimalkörnigen Prozessen, die für die Berechnung von Q-, P- und P_U -Vektorkomponenten zuständig sind. Auch virtueller Verbindungsnetzwerk (Kommunikationsgraph) soll kantenorientiert gebaut werden, um die obige Prozesse nachbarweise entsprechend der Ortsdiskretisierung zu verbinden. Dieses virtuelle paralleles Simulationsmodell gehört zu der ersten (Gleichungs)-Parallelitätsebene PE-1.

2. Ansatz: Ordnen wir den Q- und P-Gleichungen elementenweise entsprechende QP-Prozesse (2 Gleichungen des Elementes – 1 Prozess) zu. Dann erhalten wir im virtuellen parallelen Simulationsmodell:

- die reduzierte Menge

$$\sum M_i - n_k$$

von QP- Prozessen, die für die Lösung der Gleichungssysteme von Kantenelementen einschließlich P_U -Randwerten zuständig sind;

- die Menge

$$n_A + n_V$$

der Prozesse, die für die Eingabe der Randbedingungen für die Atmosphäre- und Ventilatoren-Schächte zuständig sind.

Hier ist n_k die Menge von P_U -Prozessen, die formal für die Berechnung der Druckrandwerten in den Knoten entstehen, welchen mehr als eine Kante mit dem zufließenden Luftstrom inzident sind. Zum Beispiel, für die Knoten 4 in dem Graphen der Abb. 18 sind die zufließende Zweige (Kanten) Q5, Q6, Q7 sowie ausfließender Zweig Q8 inzident. Druckwert P_{U4} wird von den für Zweigen Q5, Q6, Q7 entlang hergeleiteten Gleichungen entsprechend der letzten Elementen Q_{5M5} , Q_{6M6} , Q_{7M7} definiert:

$$dP_{U4}/dt = \gamma_5 (Q_{5M5} + Q_{6M6} + Q_{7M7} - Q_{81})$$

$$dP_{U4}/dt = \gamma_6 (Q_{5M5} + Q_{6M6} + Q_{7M7} - Q_{81})$$

$$dP_{U4}/dt = \gamma_7 (Q_{5M5} + Q_{6M6} + Q_{7M7} - Q_{81})$$

Dabei soll die Bedingung

$$\gamma_5 = \gamma_6 = \gamma_7 = \gamma_{u4}$$

erfüllt werden und für die Berechnungen nur eine Gleichung

$$dP_{U4}/dt = \gamma_{u4} (Q_{5M5} + Q_{6M6} + Q_{7M7} - Q_{81}) \quad (1)$$

statt drei obigen verwendet werden. Für dem Graphen Abb.18 machen die Knoten U2 und U4 insgesamt

$$n_k = n_k(U2) + n_k(U4) = 1+2 = 3$$

Gleichungen und P_U-Prozesse übrig. Im Gleichungsgenerator ist diese Eigenschaft der ortsdiskretisierten DNOVP berücksichtigt.

3. Ansatz: Eine weitere Vergrößerung der Arbeitsumfänge von Prozessen wird durch die Zuordnung „1 Graphenkante – 1 Prozess“ erreicht. Solcher Ansatz gibt uns genau m Prozesse, die für die Berechnung der Komponenten von Multivektoren Q, P

$$Q_j(Q_{j1}, Q_{j2}, \dots, Q_{jM_j}), j=1, 2, \dots, m$$

und

$$P_j(P_{AKJ}=P_{j1}, P_{j2}, \dots, P_{jM_j}, P_{jM_{j+1}}=P_{EKJ})$$

zuständig sind. Dabei sind P_{jAKJ} und P_{jEKJ} entsprechend die Anfangs- und Endknoten der Q_j-Kanten. Von jedem j-Prozess werden 2M_j Cauchi-Aufgaben inklusiv die Berechnung der Knotenrandwerten P_{jAKJ}, P_{jEKJ} und der äußeren Randbedingungen (Anschlüsse zur Atmosphäre und zu den Ventilatoren) gelöst.

4. Ansatz: Die nächste Struktureinheit im DNOVP-Graphen ist ein Teilnetz. Die Zerlegung des Graphen in gewisse Menge von Teilgraphen wird als ein Optimierungsproblem der Minimierung von Kommunikationsaufwand zwischen den Teilnetzen formuliert. DNOVP-Graph können wir aufgrund folgenden Überlegungen in eine Menge von Teilgraphen zerlegen:

- die formale Zerlegungsverfahren anwenden;
- die gegenstandsgebietspezifische Eigenschaften der Grubenwetternetzen benutzen (die Netzteilung in v Teile entsprechend der Ventilatorenmenge; Lokalisierung in erhaltenen v Teilen der Schemata von Abteilungsbewetterung; Lokalisierung in den Abteilungsschemata der Filtrationsmedien – Alten Männer; Definition der Anschlußknoten als der Randbedingungen und der Verbindungskanten als der Netzkomponenten);
- Kombination obigen Verfahren für die dialogunterstützte Optimierung der DNOVP-Zerlegung.

Die vorgeschlagene Parallelisierungsansätze sind in der Abb. 1 als die vier virtuellen Parallelitätsebenen (PE) zusammengefasst.

Es scheint als sinnvolles, folgende Methodik der Apriori-Analyse zu verwenden:

1. Die Analyse soll bezüglich der Parallelitätsebene vollständig durchgeführt werden
2. Die Analysevollständigkeit besteht in der Betrachtung folgenden Fragen der Funktionierung von Parallelitätsebenen:

- Analyse des Operationenumfanges im virtuellen Prozess;
- Vergleich der Prozesse nach Kriterium „Prozesslasthöhe“;
- Vorschläge zur möglich notwendige Lastbalancierung;
- Analyse der Kommunikationsoperationen zwischen der Prozessen, eine Erstellung des virtuellen Kommunikationsgraphen und Definition der Zeitschritten des Datenaustausches;
- Entwicklung des virtuellen Verbindungsnetzwerkes mit minimal möglichem Zeitaufwand für den Datenaustausch.

3 Die Parallelitätsebene 1

3.1 Eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung

Eine Struktur des virtuellen parallelen Simulationsmodells (VPSM) der nach Ortskoordinate mit M Elementen diskretisierten DNOVP-Kante wird in Abb. 2 dargestellt.

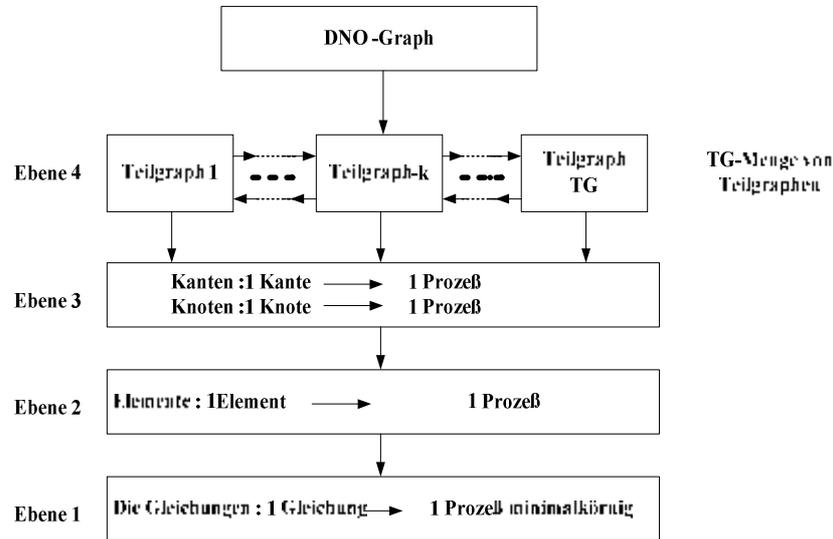


Abb. 1 – Die virtuellen Parallelitätsebenen des parallelen DNOVP-Simulationsmodells

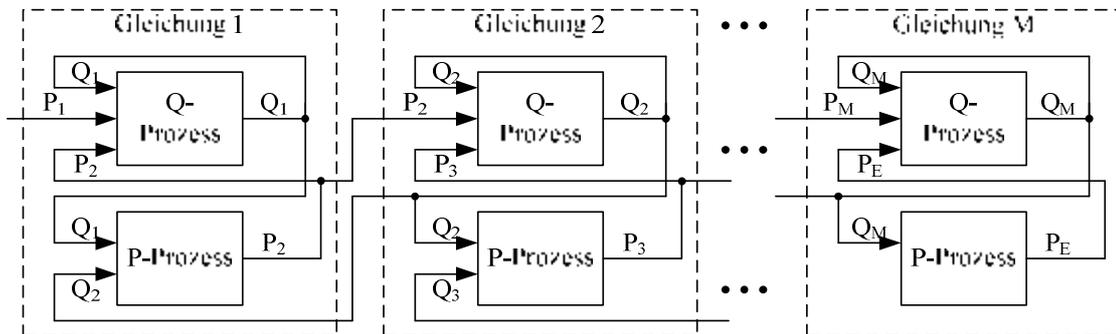


Abb. 2. – VPSM-Struktur für diskretisierte Kante des DNOVP-Graphen

Hier werden die Gleichungen

$$\begin{cases} \frac{dQ_{jk}}{dt} = \alpha_j (P_{jk} - P_{jk+1}) - \beta_j Q_{jk} |Q_{jk}| \\ \frac{dP_{jk+1}}{dt} = \gamma_j (Q_{jk} - Q_{jk+1}) \end{cases} \quad (2)$$

$$\begin{cases} \frac{dQ_{jk}}{dt} = \alpha_j (P_{jk} - P_{jk+1}) - \beta_j Q_{jk} |Q_{jk}| - \beta'_j (t) Q_{jk} |Q_{jk}| - \mu_j Q_{jk} (Q_{jk} - Q_{jk+1}) \\ \frac{dP_{jk+1}}{dt} = \gamma_j (Q_{jk} - Q_{jk+1}) - \gamma_j g_{jk} \end{cases} \quad (3)$$

$$\left. \begin{aligned} \frac{dQ_{qs}}{dt} &= \alpha_{Q_q} (P_{qs} - P_{qs+1}) - \beta_{Q_q} Q_{qs} |Q_{qs}| - \beta''_{Q_q} Q_{qs} |Q_{qs}| - \mu_{Q_q} Q_{qs} (Q_{qs} - Q_{qs+1}) \\ \frac{dP_{qs+1}}{dt} &= \gamma_{Q_q} (Q_{qs} - Q_{qs+1}) + \gamma_{Q_q} (g_{1Q_{qs}} + g_{2Q_{qs}}) \end{aligned} \right\} \quad (4)$$

$$\begin{cases} \frac{dg_{1qs}^{(v)}}{dt} = \alpha_{1qs}(g_{1qs}^{P(v)} - g_{1qs}^{P(v+1)}) - \beta_{1qs}^L g_{1qs}^{(v)} - \beta_{1qs}^T |g_{1qs}^{(v)}| |g_{1qs}^{(v)}| \\ \frac{dg_{1qs}^{P(v+1)}}{dt} = \gamma_{1qs}(g_{1qs}^{(v)} - g_{1qs}^{(v+1)}) \end{cases} \quad (5)$$

$$\begin{cases} \frac{dg_{2qs}^{(v)}}{dt} = \alpha_{2qs}(g_{2qs}^{P(v)} - g_{2qs}^{P(v+1)}) - \beta_{2qs}^L g_{2qs}^{(v)} - \beta_{2qs}^T |g_{2qs}^{(v)}| |g_{2qs}^{(v)}| \\ \frac{dg_{2qs}^{P(v+1)}}{dt} = \gamma_{2qs}(g_{2qs}^{(v)} - g_{2qs}^{(v+1)}) \end{cases}$$

durch die minimalkörnige Q- und P-Prozesse gelöst.

Die Belastung der virtuellen Prozesse (BVP) ist von Menge und Art der für die Lösung der Q- und P-Gleichungen notwendigen mathematischen Operationen abhängig. Aus (2) folgt, daß die rechte Seite (RS) der Q-Gleichung unterscheidet sich von der rechten Seite der P-Gleichung durch den Glied $\beta_j Q_{jk}/Q_{jk}$. Die Dauer der Erfüllung von RS-Operationen beträgt

$$T_Q = t_{mult} + t_{sum} + t_{mod} + 2t_{mult} + t_{sum} = 2t_{sum} + 3t_{mult} + t_{mod}, \quad (6)$$

$$T_P = t_{mult} + t_{sum}, \quad (7)$$

wo sind die t_{mult} , t_{sum} , t_{mod} entsprechend die Dauer der Operationen von Multiplikation, Summierung und der Moduldefinition der Variablen.

Die Dauerdifferenz

$$\Delta T = T_Q - T_P = t_{mod} + 2t_{mult} \quad (8)$$

ist von ausgewählten numerischen Verfahrens unabhängig und zeigt, daß die Q-Prozesse fast verdoppelte Belastung in Vergleich mit den P-Prozesse haben. Für die Gleichungen (3), (4), (5) haben wir ähnliche Schlußfolgerung. Andere minimalkörnige Prozesse sind auf Belastung mit den P-Prozesse der Gleichungen (3) ähnlich.

Bei der parallelen Prozeßausführung wird die Rechenzyklusdauer als die Zeit der RS-Berechnung im kompliziersten Q-Prozeß aus der Gleichungssysteme von Type (4) erfolgen:

$$T_Q = t_{sum} + t_{mod} + 2t_{mult} + t_{sum} + 4t_{mult} + 3t_{sum} = 5t_{sum} + 6t_{mult} + t_{mod}. \quad (9)$$

Die Dauer T_P wird für (4) als

$$T_P = 2t_{mult} + 3t_{sum} \quad (10)$$

sein, deshalb beträgt die Dauerdifferenz

$$\Delta T = T_Q - T_P = 2t_{sum} + 4t_{mult} + t_{mod} \quad (11)$$

und wird die Wartezeit für die P-Prozesse in den Gleichungen (4) verursachen. Dabei wird die maximale Wartezeit ein P-Prozeß haben, deren RS-Operationen folgende Dauer haben:

$$\Delta T_{max} = T_{Qmax} - T_{Pmax} = (5t_{sum} + 6t_{mult} + t_{mod}) - (t_{sum} + t_{mult}) = 4t_{sum} + 5t_{mult} + t_{mod} \quad (12)$$

Dieser Wert ist größer als nach (6) berechnete T_Q .

Also, die minimalkörnige Prozesse des virtuellen parallelen Simulationsmodells (Parallelitätsebene 1) haben nicht gleichmäßige Lastverteilung, die zu den Effizienzverlusten bei der parallelen Ausführung von Prozessen führt.

3.2 Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches

Wie folgt aus Abb. 2, jeder Q-Prozeß hat drei Eingangsvariablen und eine Ausgangsvariable, während der P-Prozeß – die zwei Eingangs- und eine Ausgangsvariable. Die im k -Element berechnete Q_k -Variable ist eine Rückkopplungsvariable (Abb. 2), deshalb brauchen wir nicht im Q-Prozeß eine Übergabe des Parameters Q_k von dem Ausgang nach Eingang zu organisieren. Deshalb soll Q-Prozeß die folgende Hilfsoperationen ausführen:

- Eingabe von P_k aus dem k -ten P-Prozeß;
- Eingabe von P_{k+1} aus dem $k+1$ -ten P-Prozeß;
- Sendung von Q_k dem k -ten P-Prozeß;
- Sendung von Q_k dem $k+1$ -ten P-Prozeß.

Seinerseits soll P-Prozeß ausführen:

- Sendung von P_k dem $k-1$ -ten Q-Prozeß;
- Sendung von P_k dem k -ten Q-Prozeß;
- Eingabe von Q_{k-1} aus dem $k-1$ -ten Q-Prozeß;
- Eingabe von Q_k aus dem k -ten Q-Prozeß.

Also, auf jedem Berechnungsschritt soll jeder Q- und P-Prozeß vier Datenaustauschoperationen durchführen. Bezüglich dem Q-Prozeß, der entsprechend der (6) sechs und entsprechend der (9) zwölf Berechnungsoperationen erfüllt, die Datenaustauschoperationen betragen $2/3$ und $1/3$ des Operationsumfanges. Bezüglich den P-Prozesse dieser Parameter ist noch schlimmer: entsprechend den Formeln (7) und (10) bekommen wir $4/2 = 2$ (!) und $4/5 = 0,8$.

Insgesamt für die $N_{RO}=8$ Rechenoperationen (RO) des Gleichungssystems (2) und 16 Rechenoperationen des Gleichungssystems (3) soll je $N_{DAO}=8$ Datenaustauschoperationen (DAO) durchgeführt werden. Deshalb wird Verhältnis N_{RO}/N_{DAO} für ganzen DNOVP die folgenden Werte erhalten

$$1 \leq N_{RO}/N_{DAO} \leq 2 \quad (13)$$

Aus (13) folgt, daß die 1.Parallelitätsebene „Eine Gleichung – ein Prozeß“ ungünstiges Verhältnis zwischen den Umfänge von Rechen- und Hilfsoperationen hat. Beispielsweise haben wir für die in Abb. 2 dargestellten und mit den Gleichungen (2) beschriebenen VPSM-Struktur die gesamte Prozeßdauern

$$\begin{aligned} \sum T_Q &= M T_Q, \\ \sum T_P &= M T_P. \end{aligned}$$

Die Ausführungszeit der Rechenoperationen aller Prozesse auf einem virtuellen Prozessor wird

$$T(I) = M(T_P + T_Q) \quad (14)$$

sein. Natürlich hat $T(I)$ keine Hilfsoperationen: Struktur Abb. 2 wird im Prozessor mit dem einheitlichen Speicheradressraum realisiert, wo die Vektoren $\bar{Q} = (Q_1, Q_2, \dots, Q_M)^T$ und $\bar{P} = (P_1, P_2, \dots, P_M)^T$ abgespeichert werden. Bei der Zeitschrittweite h und der Nummern der Schleifenwiederholung $0, 1, 2, \dots, n$ wird die Lösung als die Vektoren mit zeitabhängigen Komponenten dargestellt:

$$\bar{Q}(nh) = \begin{pmatrix} Q_1(nh) \\ Q_2(nh) \\ \vdots \\ Q_M(nh) \end{pmatrix} = \begin{pmatrix} Q_1(0) & Q_1(h) & Q_1(2h) & \cdots & Q_1(nh) \\ Q_2(0) & Q_2(h) & Q_2(2h) & \cdots & Q_2(nh) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Q_M(0) & Q_M(h) & Q_M(2h) & \cdots & Q_M(nh) \end{pmatrix}; \quad (15)$$

$$\bar{P}(nh) = \begin{pmatrix} P_1(nh) \\ P_2(nh) \\ \vdots \\ P_M(nh) \end{pmatrix} = \begin{pmatrix} P_1(0) & P_1(h) & P_1(2h) & \cdots & P_1(nh) \\ P_2(0) & P_2(h) & P_2(2h) & \cdots & P_2(nh) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_M(0) & P_M(h) & P_M(2h) & \cdots & P_M(nh) \end{pmatrix}.$$

Bei der parallelen Realisierung von Prozessen auf $2M$ virtuellen Prozessoren wird die Rechenzeit entsprechend (6) sein

$$T(2M) = \max(T_Q, T_P) \quad (16)$$

Entsprechend ist die Rechenzeit der parallelen Lösung nach Abb. 2 so zu definieren

$$T(2M) = \max(T_{QDA}, T_{PDA}). \quad (17)$$

Analyse von Prozessen zeigt, daß

$$T(2M) = T_{QDA} \quad (18)$$

ist, weil die Menge und Dauer der DA-Operationen in (17), (18) gleich und $T_Q > T_P$ sind.

Die virtuelle Beschleunigung mit der Berücksichtigung der virtuellen DA-Operationen ist

$$S_{vDA} = \frac{T(1)}{T(2M)} = \frac{M(T_Q + T_P)}{\max(T_{QDA}, T_{PDA})} = M \frac{T_Q + T_P}{T_{QDA}} = M \frac{1 + \frac{T_P}{T_Q}}{1 + \frac{T_{DA}}{T_Q}}$$

oder

$$S_{vDA} = M \frac{1 + \frac{T_P}{T_Q}}{1 + \frac{T_{DA}}{T_Q}}. \quad (20)$$

Aus (20) folgt, daß bei der gleichmäßigen Belastung von Q- und P-Prozessen, d. h. $T_Q = T_P$, kann eine Beschleunigung des maximal möglichen Wertes erreichen:

$$S_{vmax} = 2M \frac{1}{1 + \frac{T_{DA}}{T_Q}}. \quad (21)$$

Aus (21) folgt deutlich, wie wichtig ist die Werte T_{DA}/T_Q zu minimieren.

3.3 Schema der Verbindung zwischen den virtuellen Prozessen, ein virtuelles Verbindungsnetzwerk

Die obere Abb. 2 illustriert die Zusammenwirkungen von Q- und P-Prozessen bezüglich der Eingänge und Ausgänge. Es werden die virtuelle PQ- und QP-Kommutatoren vorgeschlagen, die aus üblichen für parallele Rechentechnik Schaltelementen (Abb. 3) zusammengestellt werden. Variable Q_k (Ausgang des Q-Prozesses) wird repliziert und nach Eingänge der Prozesse P_k, P_{k+1} eingeschaltet (Abb. 3, a); Variable P_{k+1} (Ausgang des P-

Prozesses) wird auch mit der Replizierung nach Eingänge der Prozess Q_k, Q_{k+1} eingeschaltet (Abb. 3, b).

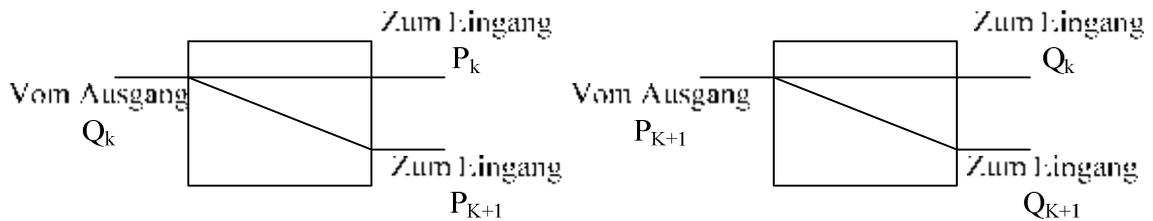


Abb. 3 – Die Schaltelemente des virtuellen Kommutators.

Abb. 4 zeigt die PQ- und QP-Kommutatoren, die Verbindungen zwischen den virtuellen parallelen Prozesse des Kantensimulationsmodells (Abb. 2) realisieren: «Ausgang P_{k+1} – Eingänge Q_k, Q_{k+1} », «Ausgang Q_k – Eingänge P_k, P_{k+1} ».

Virtueller Kommutator für die Knotenprozesse wird ähnlich gebaut. Beispielsweise haben wir für die Knote U_1 des Testobjektes die Gleichung (P_U -Prozeß)

$$\frac{dP_{U_1}}{dt} = \frac{dP_{1M_1}}{dt} = \gamma_{U_1} (Q_{1M_1} - Q_{21} - Q_{51} - Q_{61}) \quad (22)$$

Die Abbildung 5 zeigt die Schema des P_{U_1} -Prozesses und die Einschaltung der Ausgangsvariable P_{U_1} zu den Eingänge der Prozesse $Q_{1M_1}, Q_{21}, Q_{51}, Q_{61}$ über die Schaltelemente. Hier wird P_{U_1} für die Sendung in 4 Adresse repliziert. Nach P_{U_1} -Eingänge werden die Variablen entsprechend (22) von QP-Kommutatoren der Kanten Q_1, Q_2, Q_5, Q_6 gesendet.

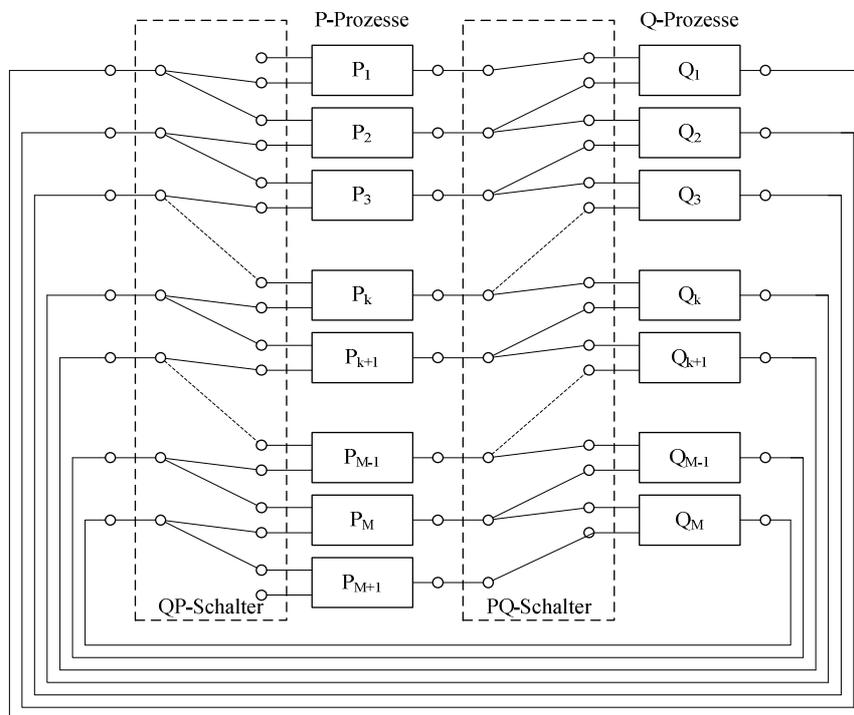


Abb. 4 – Die Virtuelle QP- und PQ-Kommutatoren

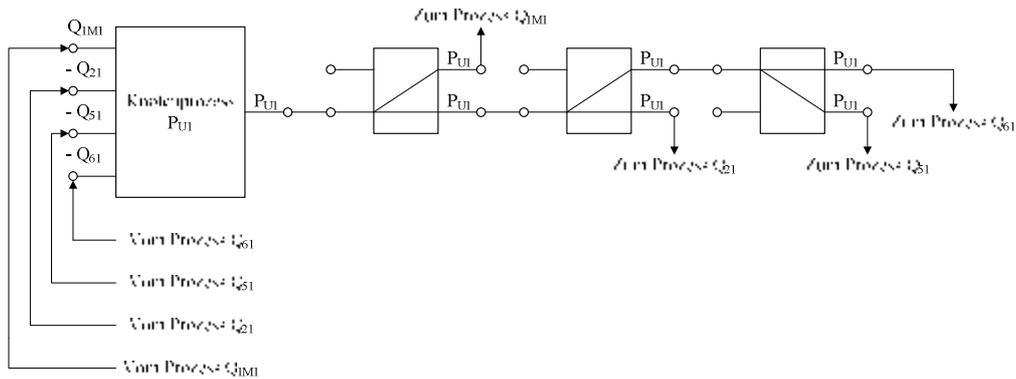


Abb. 5 – Zur Erstellung des virtuellen Knotenkommutators

Für die Minimierung der DA-Zeit T_{DA} in (21) soll die Funktionierung des virtuellen Kommutators (Abb. 4) in folgender Weise organisiert werden:

1. Die Verbindungen werden vor der Lösung erstellt und bleiben unveränderlich für ganze Simulationsperiode
2. Im i -Lösungsschleife werden die berechnete Komponentenwerte der Vektoren $\bar{Q}_j(ih), \bar{P}_j(ih)$ gespeichert und in den Ausgangsregistern der Q-, P-Prozesse abgelegt.
3. Alle Prozesse werden im Zyklus "Berechnung – Speicherung" nach längsten Prozeß synchronisiert.
4. Für alle Q- und P-Prozesse werden synchron die Send- und Receiveanweisungen aktiviert, in einem Takt erscheinen auf den Eingänge von Q- und P-Prozessen die aktuelle Werte der Variablen und werden im nächsten Takt gespeichert.
5. Es wird $i+1$ -ten Lösungsschleife gestartet.

Bei der obigen Organisation der Funktionierung des Kommutators und der parallelen Q-, P-Datenübertragung wird die DA-Zeit so definiert:

$$T_{DA} = t_{er} + t_{lr} + t_{üb} + t_{em} + t_{rs}, \quad (23)$$

wo t_{er} – Dauer der Einschreibung in Register im längsten Q-Prozeß; t_{lr} – die Lesenzzeit aus Register; $t_{üb}$ – die Übertragungszeit durch Verbindungslinie; t_{em} – die Empfangszeit aus Verbindungslinie in Register; t_{rs} – die Schreibzeit aus Register in Speicher.

3.4 Eine Möglichkeit der Zuordnung „Prozess – Prozessor“

Die minimalkörnige Prozesse des virtuellen parallelen Simulationsmodells (VPM) spiegeln die Vektor-Matrix-Darstellung von Druck- und Luftstromwerten als gesuchten Variablen wider. Der Parallelisierungsalgorithmus der betrachteten Parallelitätsebene PE-1 sieht vor, daß jeder Prozeß ein Matrixelement berechnet. Deshalb braucht PE-1 die folgende Menge von Prozessen:

$$N_{PE1} = 2 \sum_{j=1}^m M_j \cdot \quad (24)$$

Hier sind m – die Kantenmenge in DNOVP, M_j – die Elementenmenge in den ortsdiskretisierten j -Kanten. Die Parameter M_j sind von den Längen der j -Kanten und der Ortsschrittweite $\Delta \xi$ abhängig:

$$M_j = \frac{l_j}{\Delta \xi} \cdot \quad (25)$$

Parameter $\Delta\xi$ wird für ganzes Netzobjekt als einziges gewählt, deshalb kann die Prozessmenge durch gesamte Länge von Kanten berechnet werden:

$$N_{PE1} = \frac{2}{\Delta\xi} \sum_{j=1}^m l_j \cdot \quad (26)$$

Beispielsweise beträgt die gesamte Länge von Abbauten im Testnetzobjekt 7739 Meter. Bei den von verschiedenen Quellen empfohlenen $\Delta\xi$ im Diapason $10 \leq \Delta\xi \leq 100$ m erhalten wir

$$1550 \geq N_{PE1} \geq 155. \quad (27)$$

Hierfür folgt es, daß die PE-1 die Menge von virtuellen Prozessen und Prozessoren braucht, die wesentlich die technische Möglichkeiten der den Benutzern zur Verfügung stehenden MIMD-Systeme überschreiten.

Nach (26) kann man die mögliche Ortschaftweite $\Delta\xi$ und die Parametern M_j abhängig von der vorhandenen Zielrechnerarchitektur (ZRA) mit bekannten Prozessorenmenge N_{ZRA} abzuschätzen:

$$\Delta\xi_{mögl} = \frac{2}{N_{ZRA}} \sum_{j=1}^m l_j \quad (28)$$

Zum Beispiel, ein Cluster mit $N_{ZRA} = 32$ erlaubt für das Testobjekt bei der Zuordnung "1 Prozess – 1 Prozessor" die mögliche Schrittweite 484m. Solche Diskretisierung war realisierbar bei der Anwendung von Analog- und Hybridrechnern. Diese $\Delta\xi_{mögl}$ entspricht dem Diskretisierungsverfahren, das aus Analyse der Frequenzcharakteristiken von aerodynamischen Prozessen folgt.

4. Die Parallelitätsebene 2

4.1 Eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung

Auf der Parallelitätsebene PE-2 bedeutet die Zuordnung "1 Kantenelement – 1 Prozeß", daß die Gleichungssysteme von Type (2)...(4) fürs k -Element ($1 \leq k \leq M_j$) der j -Kante ($1 \leq j \leq m$) im entsprechenden k -Prozeß gelöst wird. Virtuelles paralleles Simulationsmodell zeigt Abb. 6.

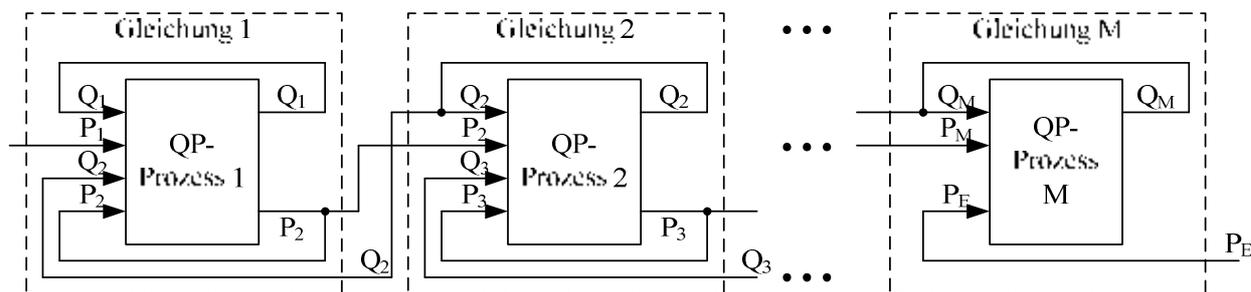


Abb. 6 – Strukturdarstellung des virtuellen parallelen Simulationsmodells für PE-2 "1 Kantenelement – 1 Prozeß"

Die Prozesse haben hier größere Arbeitsaufwand in Vergleich mit der PE-1, die Rechenzeit für die rechten Seiten im k -Prozeß beträgt

$$T_{QP} = T_Q + T_P. \quad (29)$$

Für die einfache Kanten (Luftwege) ohne der Luftverluste werden die T_Q , T_P nach den Formeln (6), (7) berechnet und T_{QP} wird minimal zwischen den Kanten des Netzobjektes sein:

$$T_{QPmin} = 2t_{sum} + t_{mod} + 3t_{mult} + t_{mult} + t_{sum} = 3t_{sum} + 4t_{mult} + t_{mod}. \quad (30)$$

Für die Kanten mit den Luftverlusten aus zwei Alten Männer entsprechen die Parametern T_Q und T_P den Formeln (9), (10) und T_{QP} wird als maximal möglichen zwischen den DNOVP-Kanten sein:

$$T_{QPmax} = 5t_{sum} + t_{mod} + 6t_{mult} + 2t_{mult} + 3t_{sum} = 8t_{sum} + 8t_{mult} + t_{mod}. \quad (31)$$

Die Zeitdifferenz der Ungleichmäßigkeit

$$\Delta T_{QP} = T_{QPmax} - T_{QPmin} = 5t_{sum} + 4t_{mult} \quad (32)$$

charakterisiert die maximale Wartezeit für die Prozesse, die die Gleichungen von Type (2) lösen. Aus der (32) und der (12) folgt, daß die Ungleichmäßigkeit auf der PE-2 kleiner geworden ist und zwar

$$\Delta T_{QP} < \Delta T_{max},$$

$$\Delta T_{max} - \Delta T_{QP} = (4t_{sum} + t_{mod} + 5t_{mult}) - (4t_{mult} + 5t_{sum}) = t_{mult} - t_{sum} + t_{mod}. \quad (33)$$

4.2 Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches

In der PE-2 ist die Verhältnis zwischen den Umfänge von Rechen- und Datenaustauschoperationen in Richtung der Senkung von letztgenannten geändert. Aus der Abb. 6 folgt es, daß k -Prozeß der j -Kante $(QP)_{jk}$ die in Abb. 7 gezeigte Struktur hat und die folgende Hilfsoperationen erfüllt :

- Eingabe der Variable P_k aus des $k-1$. Nachbarprozesses;
- Eingabe der Variable Q_{k+1} aus des $k+1$. Nachbarprozesses;
- Sendung der Variable Q_k dem $k-1$. Nachbarprozeß;
- Sendung der Variable P_{k+1} dem $k+1$. Nachbarprozeß.

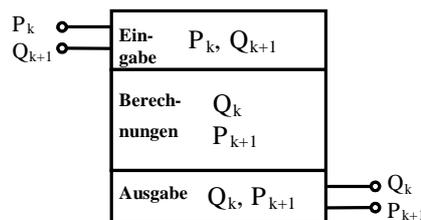


Abb. 7 – Struktur des $(QP)_{jk}$ -Prozesses.

Also, für die 8 Rechenoperationen bei der Berechnung der rechten Seite in der Gleichungen von Type (2) und für 16 Rechenoperationen – in der Gleichungen von Type (3) sollen je die 4 Datenaustauschoperationen erfüllt werden, d. h. wird die Verhältnis (13) folgendes Diapason übernehmen

$$2 \leq N_{RO}/N_{DA} \leq 4. \quad (34)$$

Aus (34) können wir zusammenfassen, daß auf der PE-2 die Verhältnis zwischen den Rechen- und Datenaustauschoperationen in 2 Mal besser geworden ist.

Mit der Berücksichtigung der virtuellen DA-Operationen wird die Dauer der parallelen Lösung der Gleichungen von j -Kante durch M Prozesse betragen

$$T(M) = T_{QP} = T_{QPmax} + T_{DA2}. \quad (35)$$

Hier ist T_{DA2} die Zeit der DA-Operationen auf der 2.Parallelitätsebene PE-2. Die virtuelle Beschleunigung ist dabei

$$S_{vQP} = \frac{T(1)}{T(M)} = \frac{M(T_Q + T_P)}{T_{QPmax} + T_{DA2}}. \quad (36)$$

Mit der Voraussetzung $T_Q + T_P = T_{QPmax}$ erhalten wir folgende Abschätzung:

$$S_{vQP} = M \frac{1}{1 + \frac{T_{DA2}}{T_{QPmax}}}. \quad (37)$$

Bei der Vergleichung von T_{DA}/T_Q in (20) mit T_{DA2}/T_{QPmax} in (37) soll berücksichtigt werden, daß $T_{DA} \approx 2T_{DA2}$ und $T_Q \approx 0,5T_{QPmax}$ sind, deshalb $\frac{T_{DA2}}{T_{QPmax}} = \frac{0,5T_{DA}}{2T_Q} = 0,25 \frac{T_{DA}}{T_Q}$ und in PE-2 beeinflussen die DA-Operationen auf die virtuelle Beschleunigung weniger:

$$S_{vQP} = M \frac{1}{1 + 0,25 \frac{T_{DA}}{T_Q}}. \quad (38)$$

4.3 Schema der Verbindung zwischen den virtuellen Prozessen, virtuelles Verbindungsnetzwerk

Schema der Verbindung zwischen den virtuellen Prozessen folgt aus der VPSM-Struktur (Abb.6). $(QP)_{jk}$ -Prozeß für k -Element der j -Kante hat die Verbindungsschema mit den Nachbarnprozesse $(QP)_{jk-1}$, $(QP)_{jk+1}$, die auf Abb. 8 dargestellt wird.

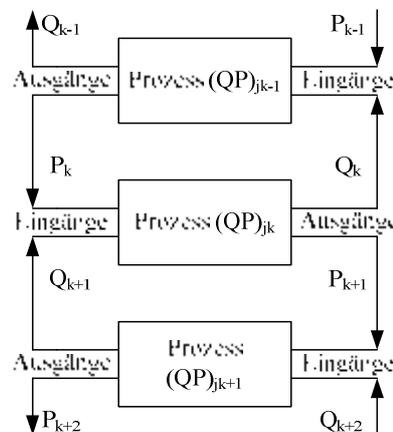


Abb. 8. – Verbindungsschema des $(QP)_{jk}$ -Prozesses

Eine Verbindung mit den Nachbarnprozesse wird gleichmäßig durch die Schaltelemente 2×2 mit der gekreuzten Einschaltung (in Abb. 8 sind die Eingänge rechts). Auf dieser Parallelitätsebene werden die Knotenprozesse P_U mit beliebigem Q-Prozeß der inzidenten Kante vereinigt. In Abb. 9 ist die Verbindungsschaltung für den Knoten U_l des Testobjektes dargestellt.

Die Verbindungsschaltungen der Abbildungen 8 und 9 können wir leicht für die Kanten und gesamtes DNOVP ausdehnen. Die Erstellung des virtuellen Verbindungsnetzwerkes wird als eine Funktion der Bedienoberfläche (BOF) der parallelen

Simulationsumgebung geplant. Diese BOF-Funktion wird die Informationen des Topologieanalysators sowie des Gleichungsgenerators benutzen.

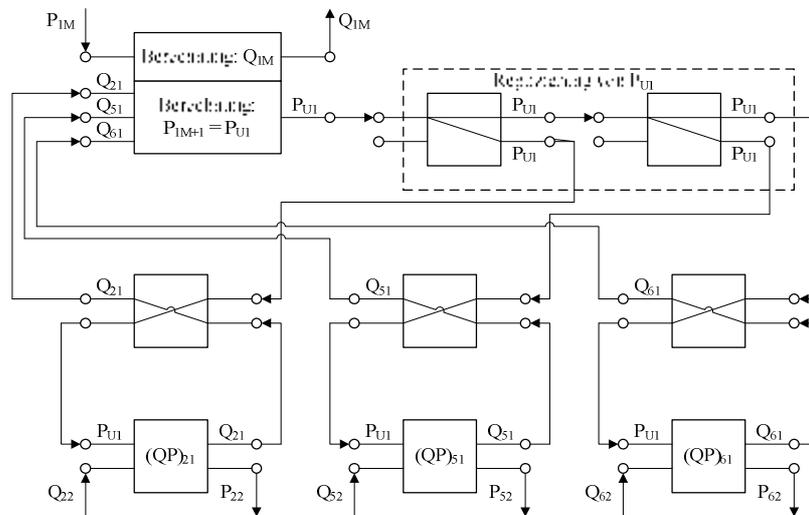


Abb. 9. – Die Verbindungen des Knotenprozesses P_{U1} mit den Prozesse der inzidenten Kanten

4.4 Eine Möglichkeit der Zuordnung „Prozess – Prozessor“

Das virtuelle parallele Simulationsmodell des DNOVP braucht auf PE-2 folgende Menge von virtuellen Prozessen

$$N_{PE2} = \sum_{j=1}^m M_j. \quad (39)$$

Mit der Berücksichtigung von (25) haben wir

$$N_{PE2} = \frac{1}{\Delta \zeta} \sum_{j=1}^m l_j. \quad (40)$$

Die Abschätzung (27) wird für das Testobjekt folgendes Diapason haben

$$775 \geq N_{PE2} \geq 78.$$

Diese Menge von Prozessen ist näher zu der Prozessorenmenge der verfügbaren MIMD-Systeme. Bei der bekannten Prozessorenmenge N_{ZRA} der Zielrechnerarchitektur können wir die mögliche Ortsschrittweite für die Diskretisierung des DNOVP wissen:

$$\Delta \zeta_{m\ddot{o}gl} = \frac{1}{N_{ZRA}} \sum_{j=1}^m l_j. \quad (41)$$

5 Die Parallelitätsebene 3

5.1 Eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung

Die Parallelitätsebene PE-3 «1 Kante – 1 Prozeß» soll m QP-Prozesse haben, die die Gleichungssysteme bezüglich der Variablen Q_{jk} , P_{jk+1} der

j -Kante ($k = 1, 2, \dots, M_j$) lösen, und noch n P_U -Prozesse, die entsprechende Knotengleichungen lösen und die Randbedingungen als Druckwerte in den Knoten für alle Kanten liefern. Aus den Gleichungen für die Kantenelemente, die als letzte vor oder als erste nach den Knoten inzident sind (PE-2), folgt, daß die P_U -Berechnungen mit den QP-Prozesse

in den mit den Knoten verbundenen Kantenelementen integriert werden. Deshalb ist nicht unbedingt nötig, die P_U -Prozesse separat zu betrachten.

Die Prozessebelastung L wird von Menge der gelösten Gleichungen definiert und beträgt pro Kante

$$L_j = 2M_j = \frac{2l_j}{\Delta\xi}. \quad (42)$$

Wegen unterschiedliche Länge der Kanten (Abbauten, Luftwege) wird die Belastung der Prozesse sich in einem Diapason

$$L_{min} \leq L_j \leq L_{max} \quad (43)$$

ändern. Hier sind L_{min} , L_{max} entsprechend die Belastungen bezüglich den Kanten der minimalen und maximalen Längen. Die Ungleichmäßigkeit beträgt dabei

$$\Delta L = L_{max} - L_{min} = 2(M_{jmax} - M_{jmin}) \quad (44)$$

und kann aus der Tabelle der DNOVP-Parametern TABUR (die Spalte l_j) berechnet werden. Nach der Ausführungszeit T_{QP} des QP-Prozesses von PE-2 kann man die maximale Wartezeit T_{Wmax} für ganzes Netzobjekt

$$T_{Wmax} = \Delta L \cdot T_{QP} \quad (45)$$

und die Wartezeiten T_{Wj} für die Prozesse aller Kanten bezüglich dem Prozeß der l_{max} -Kante

$$T_{Wj} = \Delta L_j \cdot T_{QP} \quad (46)$$

abschätzen. Hier ist $\Delta L_j = L_{max} - L_j$.

Die Abbautenlänge von Grubenbewetterungsnetzen haben die Abweichungen zueinander in mehrere Male: beispielsweise sind die Länge der Strebe $100 - 200 \text{ m}$, der Strecken – bis 2000 m , der Schächten – bis 1500 m . Deshalb wird die Ungleichmäßigkeit der Prozeßbelastungen wesentlich groß.

Aus dieser Tatsache werden eine Modifikation der Parallelitätsebene PE-3 als „ l_{jmin} -Kante – 1 Prozeß“ und folgender Algorithmus der gleichmäßigen Lastverteilung vorgeschlagen:

1. Aus der Tabelle TABUR (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:

$$N_{Pj} = l_j / l_{min} \quad (47)$$

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

$$M_j = l_j / \Delta\xi. \quad (48)$$

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 Elementendefizite ΔM_j gesucht:

$$M_{Pj} = M_{min} N_{Pj} \quad (M_{min} = l_{min} / \Delta\xi), \quad (49)$$

$$\Delta M_j = M_j - M_{Pj}. \quad (50)$$

5. Es wird ein Parameter L_{Pj} «Last» des Prozesses von j -Kante als die Elementenmenge pro Prozeß eingeführt:

$$L_{Pj} = \frac{M_{Pj}}{N_{Pj}} = M_{min}. \quad (51)$$

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

$$L_j = \frac{M_{P_j} + \Delta M_j}{N_{P_j}} = \frac{M_j}{N_{P_j}}. \quad (52)$$

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

7.2. Alle $N_{P_j} + 1$ Prozesse der j -Kante werden so belastet, daß wenigstens ein Prozeß die „nominale“ Belastung M_{min} hätte; dann werden alle andere Prozesse gleichmäßige, kleinere als M_{min} Belastung haben.

8. Die Ergebnisse obiger Operationen werden für $j = 1, 2, \dots, m$ in der Lastbalancierungstabelle (TABLAST) zusammengefasst.

9. Also, auf Parallelitätsebene PE-3 wird eine Formalisierung der Lastverteilung und Lastbalancierung erreicht. Erstellung der Tabelle TABLAST soll als eine BOF-Funktion betrachtet werden.

Lastbalancierungstabelle (TABLAST)

Номер ветви Q_j	Длина l_j , м	Миним. длина l_{jmin} , м	Шаг $\Delta \xi$, м	Кол-во элементов M_j	Кол-во процессоров $N_{P_j} = l_j / l_{jmin}$	Остаток длины Δl_j , м	$\Delta M_j = \Delta l_j / \Delta \xi$	Загрузка $L_j = M_j / N_{P_j}$	Распределение загрузки					
									$N_{P_j} + 1, j \neq j_{min}$	Вариант 1			Вариант 2	
										M_{P_j}	$\Delta L_j = \Delta M_j$	Примечание	$M_j / N_{P_j} + 1$	Примечание
Q ₁	800	–	5	160	5	50	10	32	5 + 1	30 · 5	10	5 пр. по 30 1 пр. – 10	160 / 6 = 26 + 4	1 пр. – 30 5 пр. по 26
Q ₂	150	150	5	30	1	0	0	30	1	30 · 1	0	1 пр. – 30	30 / 1	1 пр. – 30
Q ₃	1925	–	5	385	12	125	25	32 + 1/12	12 + 1	30 · 12	25	12 пр. по 30 1 пр. – 25	385 / 13 = 29 + 8	8 пр. по 30 5 пр. по 29
Q ₄	1410	–	5	282	9	60	12	31 + 1/3	9 + 1	30 · 9	12	9 пр. по 30 1 пр. – 12	282 / 10 = 28 + 2	1 пр. – 30 9 пр. по 29
Q ₅	1720	–	5	344	11	70	14	31 + 3/11	11 + 1	30 · 11	14	11 пр. по 30 1 пр. – 14	344 / 12 = 28 + 8	4 пр. по 30 8 пр. по 28
Q ₆	934	–	5	187	6	34	7	31 + 1/6	6 + 1	30 · 6	7	6 пр. по 30 1 пр. – 7	187 / 7 = 26 + 5	1 пр. – 30 5 пр. по 26 1 пр. – 27
Q ₇	150	150	5	30	1	0	0	30	1	30 · 1	0	1 пр. – 30	30 / 1	1 пр. – 30
Q ₈	650	–	5	130	4	50	10	32,5	4 + 1	30 · 4	10	4 пр. по 30 1 пр. – 10	130 / 5 = 26	5 пр. по 26
Общий параметр	$\Sigma l_j = 7739$	–	5	–	$\Sigma N_{P_j} = 49$	389	68	$L_{jmin} = 30$	$\Sigma N_{P_j} = 55$	–	–	6 недогруж. процессов		$L_{ном} = 30$ $L_{min} = 26$

5.2 Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches

Die Verhältnisse zwischen den Umfänge der Rechen- und Datenaustauschoperationen wurden auf PE-3 in Richtung der Verkleinerung des Umfanges der Hilfsoperationen verbessert. Die Prozeß-Strukturen, die der j -Kante insgesamt und der auf $N_{Pj} = l_j/l_{min}$ Subkanten geteilten j -Kante entsprechen, sind in Abb. 10 ($j = 1$, Kante Q_1 des Testobjektes) gezeigt. Betrachtete Kante Q_1 hat $l_1 = 800 \text{ m}$ und wird bei $\Delta\xi = 5 \text{ m}$ mit den $M_1 = 160$ Elementen approximiert. Kanteprozeß löst $2M_j$ Gleichungen und erfüllt folgende Hilfsoperationen:

- Eingabe P_{j1} aus des Knotenprozesses von inzidenten Anfangsknoten (für Q_1 : $P_{11} = P_A - \text{Athmospheredruck}$);
- Eingabe der Luftströme von Anfangselementen der Kanten, die dem Endknoten der j -Kante inzident sind (für $Q_1 - \text{Knote } U_1$ und die Luftströme Q_{21}, Q_{51}, Q_{61});
- Sendung des Luftstroms Q_{j1} dem Prozeß des Anfangsknotens (für Q_1 wird nicht erfüllt);
- Sendung des in Endknoten der Kanten Q_j berechneten Druckes $P_{jM_{j+1}}$ (für $j = 1$: $P_{1M_1+1} = P_{U1}$) den Prozesse, die Luftströme Q_{j1} in den Inzidentkanten berechnen (für $j = 1$ sind die Elemente Q_{21}, Q_{51} und Q_{61}).

Also, für je $N_{RO} = 8M_j$ Rechenoperationen bei der Abrechnungen der rechten Seiten Gleichungen (2) und für je $N_{RO} = 16M_j$ Rechenoperationen – der Gleichungen (3), sollen wie auf PE-2 $N_{DA} = 4$ Hilfsoperationen erfüllt werden, d. h., wird Verhältnis (13) besseres Diapason übernehmen:

$$M_j \leq N_{RO}/N_{DA} \leq 4M_j. \quad (53)$$

Also, auf PE-3 beim Variant „1 Kante – 1 Prozeß“ (Abb. 10, b) wird M_j -malig das Verhältnis zwischen den Umfänge von Rechen- und DA-operationen für alle Kanten verbessert.

Beim Variant „1 Subkante – 1 Prozeß“ (Abb. 10, c) entsprechen jeder Kante $N_{Pj} + 1$ Prozesse. Aus Prozeßstrukturen folgt, daß jeder Prozeß nicht weniger als $M_{min} = l_{jmin}/\Delta\xi$ Gleichungspaare (2) oder (3) löst und dabei die folgende Hilfsoperationen erfüllt:

- Eingabe P_{jk} aus dem letzten Elementen des Nachbarnprozesses (für $j = 1, k = 31$ ist das P_{131} in Abb. 10, c);
- Eingabe Q_{jk+h} aus dem ersten Elementen des Nachbarnprozesses ($h \leq M_{min} -$ eine Schrittweite nach Elementenmenge in den Prozesse; für $k = 31, h = 26$ und $Q_{jk+h} = Q_{1157}$, Abb. 10, c);
- Ausgabe Q_{jk} aus dem ersten Elementen nach den letzten Elementen des Nachbarnprozesses;
- Ausgabe P_{jk+h} aus dem letzten Elementen nach den ersten Elementen des Nachbarnprozesses.

Daraus folgt, daß die Menge der Hilfsoperationen zwischen den Prozesse gleich 4 geblieben ist. Also, für je $N_{RO} = 8M_{min}$ Operationen der Berechnungen der rechten Seiten der Gleichungen (2) und für je $R_{BO} = 16M_{min}$ Operationen – Gleichungen (3) sollen $N_{DA} = 4$ DA-Operationen erfüllt werden. Deshalb wird das Verhältnis (13) folgendes Wertediapason übernehmen:

$$2M_{min} \leq N_{RO}/N_{DA} \leq 4M_{min}, \quad (54)$$

Das ist M_{min} -malig besser, als auf PE-2. Im Vergleich mit dem Variant 1 ist hier den Teil der DA-Operationen höher, weil für jede Kante haben wir größere Menge der Prozesse (M_j/M_{min}) zugeordnet. Dabei wurde bessere Gleichmäßigkeit von Prozesslasten erreicht.

Analyse zeigt, daß die Lösungszykluszeit (LZ) von Gleichungen der j -Kanten auf Ebene «Kante – Prozeß» sein wird

$$T_{jLZ} = M_j T_{QPj}, \quad (55)$$

wo T_{QPj} ist Dauer des QP-Prozesses, die von Art der Gleichungen der j -Kante abhängig ist.

Eine Realisierung der m Prozesse in dem virtuellen Einprozessorsystem wird für ganzen DNOVP der folgenden Rechenzeit brauchen:

$$T_{LZ}(1) = \sum_{j=1}^m M_j T_{QPj}. \quad (56)$$

Bei der parallelen Realisierung in dem virtuellen m -Prozessorsystem wird die LZ-Dauer für DNOVP von der Dauer des Prozesses der l_{max} -Kante und der DA-Zeit $T_{3DA}^{(1)}$ (PE-3, 1.Variant) abhängig sein

$$T_{LZ}(m) = \text{MAX}_{j=1}^m (M_j T_{QPj}) + T_{3DA}^{(1)}. \quad (57)$$

Die virtuelle Beschleunigung für PE-3, Variant 1, beträgt

$$S_{v3}^{(1)} = \frac{T_{LZ}^{(1)}}{T_{LZ}(m)} = \frac{\sum_{j=1}^m M_j T_{QPj}}{\text{MAX}_{j=1}^m (M_j T_{QPj}) + T_{3DA}^{(1)}}. \quad (58)$$

Die Operation MAX wird bezüglich der in der Tabelle TABLAST befindene Daten erfüllt und gibt:

$$\text{MAX}_{j=1}^m (M_j T_{QPj}) = M_{j_{max}} T_{QPj_{max}}$$

Mit dieser Ergebnis kann man die Formel (58) so umwandeln:

$$S_{v3}^{(1)} = \frac{1 + (\sum_{k=1}^{m-1} M_k T_{QPk}) / M_{j_{max}} T_{QPj_{max}}}{1 + \frac{T_{3DA}^{(1)}}{M_{j_{max}} T_{QPj_{max}}}}, \quad (59)$$

wo $k \neq j_{max}$ und j_{max} ist Nummer der l_{max} -Kante.

Aus (59) ist zu sehen, daß die DA-Organisation der entscheidende Faktor für die Effizienz der Parallelisierungsansätze ist. Maximale Menge der DA-Operationen ist in PE-3 von der Knotenmenge des DNOVP abhängig. Im ungünstigsten Fall der sequentiellen DA-Organisation werden wir haben:

$$T_{3DA}^{(1)} = n \cdot T_{UDA}. \quad (60)$$

Bei der parallelen gleichzeitigen Ausführung diesen Operationen auf alle Knoten erhalten wir

$$T_{3DA}^{(1)} = T_{UDA},$$

wo T_{UDA} ist die Ausführungszeit von DA-Operationen im j -Kanteprozess, der den Knotendruck P_{Ui} ($i = 1, 2, \dots, n$) berechnet.

Im 2. Variant der PE-3 „ l_{min} -Kante – Prozeß“ wird die Prozessesmenge für DNOVP sein

$$N_p = \sum_{j=1}^m N_{p_j} + (m - m_{ml}). \quad (61)$$

Hier sind $N_{p_j} = \frac{l_j}{l_{min}}$ mit Verzichten vom Rest; m_{ml} – die Menge von l_{min} -Kanten; $(m - m_{ml})$ – die Menge von zusätzlichen Prozessen in den Kanten mit $l_j > l_{min}$ (1 Prozeß je Kante, Tabelle TABLAST).

Die parallel erfüllte Prozesse N_{p_j} lösen die M_{min} Paare der Gleichungen (2) – (4). Die Zeit T_{QPmax} entspricht den Gleichungen von Type (4). Dabei erfüllt jeder Prozeß die 4 Datenaustauschoperationen mit den Nachbarnprozesse.

Bezeichnen wir die Dauer des Datenaustausches zwischen den Prozesse als T_{zpDA} . Eine Menge solchen „Austauschblöcke“ beträgt im virtuellen parallelen Simulationsmodell des DNOVP $N_p - n$ (n – die Knotenmenge). Es geht um Tatsache, daß in n Subkanten, die den Knoten inzident sind, die DA-Operationen erfüllt werden, die für die Berechnungen der Knotendrücken notwendig sind. Die Dauer diesen DA-Operationen ist gleich der Dauer solchen Operationen im Variant 1, d. h. T_{UDA} . Die Dauer des Lösungszyklus, also, wird nach folgende Formel definiert:

$$T_{LZ}(N_p) = M_{min} T_{QPmax} + (N_p - n) \cdot T_{zpDA} + n T_{UDA}. \quad (62)$$

Die virtuelle Beschleunigung wird für Variant 2 der PE-3 sein:

$$S_{v3}^{(2)} = \frac{T_{LZ}^{(1)}}{T_{LZ}(N_p)} = \frac{\sum_{j=1}^m M_j T_{QPj}}{M_{min} T_{QPmax} + (N_p - n) \cdot T_{zpDA} + n T_{UDA}}. \quad (63)$$

In (63) ist ungünstiger Fall der sequentiellen Funktionierung von „Austauschblöcken“ in Betracht genommen. Es entsteht aktuelle Aufgabe der Organisation des parallelen Datenaustausches mit der gesamten Dauer der DA-Operationen

$$T_{3DA}^{(2)} = \max(T_{zpDA}, T_{UDA}) \quad (64)$$

Die Zeit T_{zpDA} ist von 4 DA-Operationen abhängig. Die Dauer der Knotenaustauschoperationen T_{UDA} wird abhängig von der Menge der Kanten, die den entsprechenden Knoten inzident sind, definiert. Deshalb ist die Abschätzung $T_{UDA} > T_{zpDA}$ gültig und die virtuelle DA-Zeit wird wie in Variant 1 der PE- 3 sein:

$$T_{3DA}^{(2)} = T_{UDA} \quad (65)$$

Also, die mögliche maximale Beschleunigung nach (63) kann so abgeschätzt werden

$$S_{v3max}^{(2)} = \frac{\sum_{j=1}^m \frac{M_j T_{QPj}}{M_{min} T_{QPmax}}}{1 + \frac{T_{UDA}}{M_{min} T_{QPmax}}}. \quad (66)$$

5.3 Schema der Verbindung zwischen den virtuellen Prozessen, ein virtuelles Verbindungsnetzwerk

Aus der Struktur von Prozessen in Abb. 10 (b, c) ist zu entnehmen, daß die Verbindungsschemata zwischen den j -Prozesse (Variant 1 der PE-3) und den Prozesse der Knoten inzidenten Teilkanten (Variant 2 der PE-3) ähnlich sind und die Schaltung der Abb. 9 wiederholen. Die Prozesse der Struktur Abb. 10 (c) beinhalten M_{min} QP-Prozesse der PE-2 (QP_{jk} -Prozeß in Abb. 8, ist als Subprozeß bezüglich dem Prozeß der PE-3 zu betrachten), die die Q-, P-Variablen der j -Kante von $k = 1$ bis M_{min} (erster Prozeß), $k = M_{min} + 1$ bis $k = 2M_{min}$ (zweiter Prozeß) usw. berechnen. Entsprechend der Verbindungsschema sendet der letzte Subprozeß $QP_{jM_{min}}$ dem ersten Subprozeß $QP_{jM_{min}+1}$ des Nachbarnprozesses den Druck $P_{jM_{min}+1}$ und empfängt von ihm den Luftstrom $Q_{jM_{min}+1}$. Genau so sendet letzter Subprozeß $QP_{j2M_{min}}$ des Nachbarnprozesses den Druck $P_{j2M_{min}+1}$ seinem Nachbarn und empfängt von ihm den Luftstrom $Q_{j2M_{min}+1}$. Wenn Elementenmenge in den Prozesse ist kleiner als M_{min} (Tabelle TABLAST), dann bleibt Verbindungsschema ohne Änderungen und entspricht genau der Abb. 8: die Übertragung von Druck- und Luftstromwerten wird durch die Schaltelemente 2×2 durchgeführt. Ihre Menge pro eine Kante ist $N_{Pj} - 1$ (oder N_{Pj} , wenn in TABLAST wurde $N_{Pj}+1$ Prozesse vorgesehen). In Abb. 11 ist das Verbindungsschema zwischen den Prozesse der PE-3 für das Testobjekt dargestellt: Prozeß 2 der Kante $j=1$ ist mit den Nachbarn, dem Prozeß 1 und dem Prozeß 3, durch die gekreuzt eingeschaltete Schaltelemente verbunden. Dabei haben diese Nachbarnprozesse folgende Menge von PE-2-Subprozesse entsprechend der TABLAST:

- Prozeß 1 bearbeitet $M_{11} = 30$ Elemente (M_{min}), d. h. hat 30 Subprozesse, die $Q_{11} \dots Q_{130}$, $P_{12} \dots P_{131}$ berechnen;
- Prozeß 2 bearbeitet $M_{12} = 26$ Elemente ($< M_{min}$), d. h. hat 26 Subprozesse, die $Q_{131} \dots Q_{156}$, $P_{132} \dots P_{157}$ berechnen;
- Prozeß 3 bearbeitet $M_{13} = 26$ Elemente ($< M_{min}$), d. h. hat 26 Subprozesse, die $Q_{157} \dots Q_{182}$, $P_{158} \dots P_{183}$ berechnen.

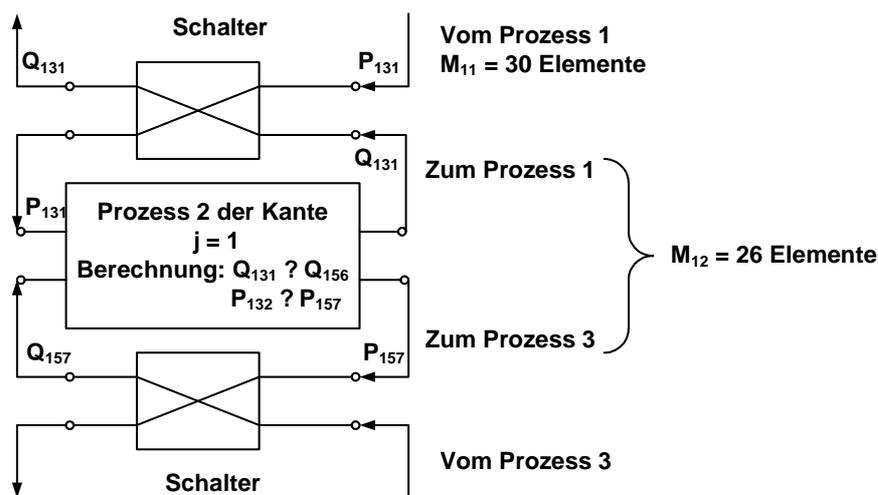


Abb. 11 – Schema der Verbindungen zwischen den PE-3-Prozesse

Also, virtueller Kommutator (virtuelles Verbindungsnetzwerk) des virtuellen parallelen PE-3-Simulationsmodells wird ähnlich mit dem PE-2-Kommutator mit der Verwendung von (2×2) -Schaltelementen gebaut. Es wird die Kommutatorsynthese

entsprechend den Angaben des Topologieanalysators und des Gleichungsgenerators sowie der TABLAST automatisiert. Für die Minimierung der gesamten Dauer der Datenaustauschoperationen auf der PE-3 (Variant 2)

$$T_{3DA}^{(2)} = (N_P - n) \cdot T_{zpDA} + nT_{UDA}$$

und Realisierung der Bedingung (64) ist eine Steuereinheit für virtueller Kommutator erforderlich, die die Speicherung der zu übertragene Daten in den Pufferregistern sowie gleichzeitige parallele Datenübertragung den Nachbarnprozesse verwalten kann.

5.4 Eine Möglichkeit der Zuordnung „Prozess – Prozessor“

Wie schon erwähnt wurde, die 1. und 2. Varianten der PE-3 brauchen für ganzes DNOVP folgenden Prozessmengen:

$$N_{PE3}^{(1)} = m \quad (67)$$

$$N_{PE3}^{(2)} = N_P = \sum_{j=1}^m N_{Pj} + (m - m_{ml}) \cdot \quad (68)$$

Mit der Berücksichtigung $N_{Pj} = \frac{l_j}{l_{\min}}$ erhalten wir

$$N_{PE3}^{(2)} = \frac{1}{l_{\min}} \sum_{j=1}^m l_j + (m - m_{ml}) \cdot \quad (69)$$

Die Abschätzung von Art (27) ist auf PE-3 von $\Delta\zeta$ direkt unabhängig und wird nach (67) und (68) definiert. Die Schrittweite $\Delta\zeta$ wirkt auf PE-3 nach den Aufwand von Rechenoperationen in den Prozesse und, natürlich, nach der Effizienzabschätzungen (59) und (66).

Die Zuordnung auf parallele virtuelle Systeme sieht das Verhältnis „1Prozeß – 1Prozessor“ als üblich aus. Die solche Zuordnung auf eine Zielrechnerarchitektur ist von beiden Varianten der PE-3 möglich: die Werte $N_{PE3}^{(1)}$ und $N_{PE3}^{(2)}$ befinden sich in dem Gebiet der Mengen von CPUs in vorhandenen MIMD-ZRA.

6. Die Parallelitätsebene 4

6.1 Eine Lastverteilung zwischen den virtuellen Prozessen und die Vorschläge zur Lastbalancierung

Auf der PE-4 wird DNOVP-Graph in eine Menge G_s Teilgraphen zerlegt. Dabei wird versucht, die Menge der Verbindungen zwischen den Teilgraphen zu minimieren. Die den Teilgraphen zugehörige Gleichungssysteme werden in G_s MIMD-Prozesse gelöst. Hier sind die Prozessmenge von der Menge der Teilgraphen abhängig. Die Gleichmäßigkeit der Prozessebelastung ist voll durch die Graphenzerlegung definiert.

Es wird kombinierten Ansatz zur Graphenzerlegung vorgeschlagen, der gemeinsam technologische (gegenstandsgebietspezifische) Überlegungen und formale Zerlegungsmethoden verwendet. In dem Fall von Grubenbewetterungsnetze werden deutliche Netzhierarchie und bergbautechnische Bedingungen in Betracht genommen:

- Entsprechend der Menge HV von Hauptventilatoren wird der Bewetterungsnetz nach HV der größten Netzfragmente (Teilgraphen) zerlegt;
- Als nächste Hierarchieebene werden die von Tiefen abhängige Arbeitshorizonte betrachtet. Im Testobjekt beim $HV=1$ gibt es zwei Horizonten: die Kanten Q_3, Q_4 vereinigen

$$M_{F_1} = \frac{l_1 + l_2 + l_7 + l_8}{\Delta \xi} = \frac{1750}{5} = 350,$$

$$M_{F_2} = \frac{l_3 + l_4}{\Delta \xi} = \frac{3335}{5} = 667,$$

$$M_{F_3} = \frac{l_5 + l_6}{\Delta \xi} = \frac{2654}{5} = 531.$$

Bei gesamt für DNOVP Schrittweite $\Delta \xi$ führt die obige technologische Zerlegung zur wesentlichen Ungleichmäßigkeit der Prozessebelastung. Für die Lastbalancierung wird folgendes empfohlen:

1. Den technologisch definierten Fragmenten mehr als einen Prozeß zuordnen. Man kann, beispielsweise, für obigen Fragmenten F_2 und F_3 , je zwei Prozesse zuordnen und ihre Belastung halbieren. Zu diesen Fragmenten in den Kanten Q_3 , Q_4 , Q_5 und Q_6 werden auch die Prozesse der Regelungssysteme eingefügt werden.

2. Bei der Formierung von Fragmenten sind die Kanten mit der Berücksichtigung der technologischen Faktoren und ihrer Länge l_j zu wählen, um die gleichmäßige resultierende Fragmentenlänge zu erhalten.

Wegen großen Verschiedenheiten von Abbautenlängen sieht ersten Weg als vorteilhaften aus. Alle Daten für die Anwendung beiden Wege kann man aus Tabelle TABLAST aus PE-3 abholen.

6.2 Vergleich des Umfanges der Datenaustauschoperationen mit dem Umfang der Berechnungen und die Vorschläge zur Minimierung des Datenaustausches

Für die Abschätzung der Verbindungsschemata zwischen den Prozesse und der Verhältnisse zwischen den Rechen- und Datenaustauschoperationen betrachten wir die in den Fragmenten F_1, F_2, F_3 (Abb. 12) gelösten Gleichungssysteme:

Prozeß F_1

$P_A = P_{11}$ – die Randbedingung

$$\begin{array}{l}
 \left\{ \begin{array}{l} \frac{dQ_{11}}{dt} = \alpha_1(P_{11} - P_{12}) - \beta_1 Q_{11} | Q_{11} \\ \frac{dP_{12}}{dt} = \gamma_1(Q_{11} - Q_{12}) \end{array} \right. \\
 Q_1 \vdots \\
 \left\{ \begin{array}{l} \frac{dQ_{1M_1}}{dt} = \alpha_1(P_{1M_1} - P_{U_1}) - \beta_1 Q_{1M_1} | Q_{1M_1} \\ \frac{dP_{U_1}}{dt} = \gamma_1(Q_{1M_1} - Q_{21} - Q_{51} - Q_{61}) \end{array} \right. \quad \text{Berechnung } P_{U_1} \\
 \\
 \left\{ \begin{array}{l} \frac{dQ_{71}}{dt} = \alpha_7(P_{71} - P_{72}) - \beta_7 Q_{71} | Q_{71} \\ \frac{dP_{72}}{dt} = \gamma_7(Q_{71} - Q_{72}) \end{array} \right. \\
 Q_7 \vdots \\
 \left\{ \begin{array}{l} \frac{dQ_{7M_7}}{dt} = \alpha_7(P_{7M_7} - P_{U_4}) - \beta_7 Q_{7M_7} | Q_{7M_7} \\ \frac{dP_{U_4}}{dt} = \gamma_7(Q_{7M_7} + Q_{5M_5} + Q_{6M_6} - Q_{81}) \end{array} \right. \quad \text{Berechnung } P_{U_4}
 \end{array}
 \quad ; \quad
 \begin{array}{l}
 \left\{ \begin{array}{l} \frac{dQ_{21}}{dt} = \alpha_2(P_{U_1} - P_{22}) - \beta_2 Q_{21} | Q_{21} \\ \frac{dP_{22}}{dt} = \gamma_2(Q_{21} - Q_{22}) \end{array} \right. \\
 Q_2 \vdots \\
 \left\{ \begin{array}{l} \frac{dQ_{2M_2}}{dt} = \alpha_2(P_{2M_2} - P_{U_2}) - \beta_2 Q_{2M_2} | Q_{2M_2} \\ \frac{dP_{U_2}}{dt} = \gamma_2(Q_{2M_2} - Q_{31} - Q_{41}) \end{array} \right. \quad \text{Berechnung } P_{U_2} \\
 \\
 \left\{ \begin{array}{l} \frac{dQ_{81}}{dt} = \alpha_8(P_{U_4} - P_{82}) - \beta_8 Q_{81} | Q_{81} \\ \frac{dP_{82}}{dt} = \gamma_8(Q_{81} - Q_{82}) \end{array} \right. \\
 Q_8 \vdots \\
 \left\{ \begin{array}{l} \frac{dQ_{8M_8}}{dt} = \alpha_8(P_{8M_8} - P_V) - \beta_8 Q_{8M_8} | Q_{8M_8} \\ P_V = f(Q_{8M_8}) - \text{Ventilatorencharakteristik} \end{array} \right.
 \end{array}$$

Also, Prozeß F_1 wird die Knotendrucke P_{U_1} , P_{U_2} und P_{U_4} berechnen und die Verbindungen mit den Fragmenten F_2, F_3 erfüllen; hier wird auch das Ventilatorsmodell realisiert.

Prozeß F_2

$$\begin{cases} \frac{dQ_{31}}{dt} = \alpha_3(P_{U_2} - P_{32}) - \beta_3 Q_{31} | Q_{31} | \\ \frac{dP_{32}}{dt} = \gamma_3(Q_{31} - Q_{32}) \quad P_{U1} \text{ - ist bekannt aus } F_1 \end{cases} \quad ; Q_3 \vdots \begin{cases} \frac{dQ_{41}}{dt} = \alpha_4(P_{U_2} - P_{42}) - \beta_4 Q_{41} | Q_{41} | \\ \frac{dP_{42}}{dt} = \gamma_4(Q_{41} - Q_{42}) \quad P_{U2} \text{ - ist bekannt aus } F_1 \end{cases} ; Q_4 \vdots \\
 \begin{cases} \frac{dQ_{3M_3}}{dt} = \alpha_3(P_{3M_3} - P_{U_3}) - \beta_3 Q_{3M_3} | Q_{3M_3} | \\ \frac{dP_{U_3}}{dt} = \gamma_3(Q_{3M_3} + Q_{4M_4} - Q_{71}) \text{ Berechnung } P_{U3} \end{cases} \quad ; Q_3 \vdots \begin{cases} \frac{dQ_{4M_4}}{dt} = \alpha_4(P_{4M_4} - P_{U_3}) - \beta_4 Q_{4M_4} | Q_{4M_4} | \\ P_{U_3} \text{ - bekannt aus } F_2 \end{cases}$$

Prozeß F_3

$$\begin{cases} \frac{dQ_{51}}{dt} = \alpha_5(P_{U_1} - P_{52}) - \beta_5 Q_{51} | Q_{51} | \\ \frac{dP_{52}}{dt} = \gamma_5(Q_{51} - Q_{52}) \quad P_{U1} \text{ - ist bekannt aus } F_1 \end{cases} \quad ; Q_5 \vdots \begin{cases} \frac{dQ_{61}}{dt} = \alpha_6(P_{U_1} - P_{62}) - \beta_6 Q_{61} | Q_{61} | \\ \frac{dP_{62}}{dt} = \gamma_6(Q_{61} - Q_{62}) \end{cases} ; Q_6 \vdots \\
 \begin{cases} \frac{dQ_{5M_5}}{dt} = \alpha_5(P_{5M_5} - P_{U_4}) - \beta_5 Q_{5M_5} | Q_{5M_5} | \\ P_{U_4} \text{ - ist bekannt aus } F_1 \end{cases} \quad ; Q_5 \vdots \begin{cases} \frac{dQ_{6M_6}}{dt} = \alpha_6(P_{6M_6} - P_{U_4}) - \beta_6 Q_{6M_6} | Q_{6M_6} | \\ P_{U_4} \text{ - ist bekannt aus } F_1 \end{cases}$$

In Abb. 13 werden der Blockschaltbild und die Verbindungen von Prozessen dargestellt, die obigen Fragmenten F_1 , F_2 , F_3 entsprechen.

Die Verbindungen F_1 mit F_2 :

- 1) Es wird P_{U2} für die Berechnungen von Q_3 und Q_4 gesendet;
- 2) Es wird Q_{71} an Q_3 -Subprozeß für die Berechnung von P_{U3} gesendet;
- 3) Es wird P_{U3} für die Berechnung Q_{71} empfangen;
- 4) Es werden die Q_{31} und Q_{41} für die Berechnung von P_{U2} empfangen.

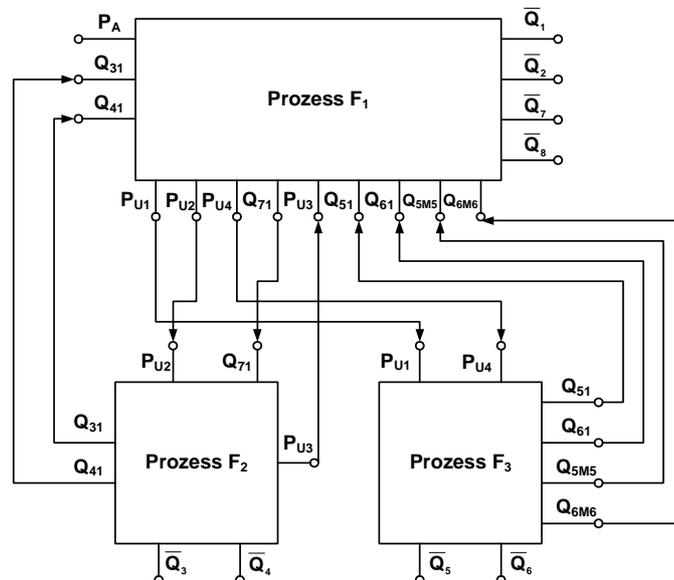


Abb. 13 – Blockschaltbild der Fragmentprozesse

Die Verbindungen F_1 mit F_3 :

- 1) Es werden P_{U1} und P_{U4} für die Berechnungen von Q_5 und Q_6 gesendet;
- 2) Es werden die Q_{51} , Q_{61} und Q_{5M5} , Q_{6M6} für die Berechnungen von P_{U1} und P_{U4} empfangen.

Die Fragmente F_2 und F_3 haben keine direkte Verbindung, ein indirekten Datenaustausch fließt über Fragment F_1 .

Die Abschätzung der Verhältnisse zwischen den Umfänge von Rechn- und Datenaustauschoperationen soll bezüglich der erhaltenen Fragmenten-Prozesse durchgeführt werden. Aus Abb. 13 folgt, daß Prozeß F_1 die Menge $M_{F_1} = \frac{l_1 + l_2 + l_7 + l_8}{\Delta\xi}$ der Gleichungspaare

für die dem Fragment gehörten Kantenelemente sequentiell löst. Dabei die Anzahl der Eingangsoperationen gleich 7 ist ($Q_{31}, Q_{41}, P_{U3}, Q_{51}, Q_{61}, Q_{5M5}, Q_{6M6}$) und der Ausgabeoperationen – 4 ($P_{U1}, P_{U2}, P_{U4}, Q_{71}$). Daraus können wir festzustellen, daß für je $8M_{F_1}$ Rechenoperationen der Berechnung von rechten Seiten der Gleichungen von Type (2) und für je $16M_{F_1}$ solchen Operationen bezüglich der Gleichungen von Type (3) die 11 Datenaustauschoperationen durchgeführt werden sollen, d. h., wird das Verhältnis der Art (13) folgendes Diapason übernehmen

$$\frac{8M_{F_1}}{11} \leq N_{RO}^{F_1} / N_{DA}^{F_1} \leq \frac{16M_{F_1}}{11}.$$

Für die Prozesse F_2 und F_3 (Abb. 13) bekommen wir entsprechend:

$$\frac{8M_{F_2}}{5} \leq N_{RO}^{F_2} / N_{DA}^{F_2} \leq \frac{16M_{F_2}}{5},$$

$$\frac{8M_{F_3}}{6} \leq N_{RO}^{F_3} / N_{DA}^{F_3} \leq \frac{16M_{F_3}}{6}.$$

Diese Verhältnisse zeigen, daß auf der PE-4 ein Teil der Datenaustauschoperationen in der den Netzfragmenten entsprechenden Prozesse wesentlich niedriger in vergleich mit den Ebenen PE-1, PE-2, PE-3 ist.

Die Lösungszykluszeit der parallelen Realisierung von Prozessen wird für ganzes DNOVP von längsten Prozeß $F_{max} = F$ bedingt, deshalb bekommen wir die virtuelle Beschleunigung

$$S_{v4} = \frac{T(1)}{T_F}. \quad (70)$$

Hier ist

$$T(1) = \sum M_j T_{QPj} \text{ für } j=1,2,\dots,m.$$

Die Zeitdauer der QP-Lösungsoperationen T_{QPj} ist von j-Kanten- bzw. Gleichungspaartype abhängig. Die Zeitdauer des F-Prozesses besteht aus der Rechenzeit und der DA-Zeit:

$$T_F = \sum M_i T_{QPi} + T_{FDA} \text{ für } i=1,2,\dots,m_F,$$

dabei m_F ist die Kantenmenge im maximalen Netzfragment F. Die Formel (70) können wir in mehr informativen Form darstellen:

$$S_{v4} = \left[\frac{\sum M_j T_{QPj}}{\sum M_i T_{QPi}} \right] * \frac{1}{\left[1 + \frac{T_{FDA}}{\sum M_i T_{QPi}} \right]}. \quad (71)$$

6.3 Schema der Verbindungen zwischen den virtuellen Prozessen, virtuelles Verbindungsnetzwerk

Die Verbindungen zwischen den F-Prozesse nach Schema der Abb. 13 werden mit Hilfe dieselben Schaltelementen $2*2$, wie auf den Ebenen PE-2, PE-3 realisiert.

6.4 Eine Möglichkeit der Zuordnung „Prozeß – Prozessor“

Die Zuordnung „Prozeß – Prozessor“ ist leicht in modernen MIMD-Systeme realisierbar.

Zusammenfassung und Ausblick

- Virtuelle parallele Simulationsmodelle des DNOVP haben vier möglichen Parallelitätsebenen (PE).
- Es wurde die Analyse PE-1...PE-4 bezüglich
 - der Lastverteilung zwischen den Prozesse,
 - der RO/DA-Verhältnisse,
 - der virtuellen Beschleunigung,
 - des virtuellen Verbindungsnetzwerks,
 - der Zuordnung „Prozeß – Prozessor“
- durchgeführt, die Abschätzungsformeln hergeleitet. Die in Tabelle TABAPRIANA zusammengestellte Analyseergebnisse dienen für systematische Implementierungen der virtuellen Simulationsmodelle auf verschiedenen Zielrechnerearchitekturen.
- Die weitere Arbeiten in Kooperation mit dem HLRS werden so geplant:
- Vorbereitung der Versuchsnetze $m=8$, $m=117$ und der Netze von zwei Gruben ($m>500$) zur parallelen Modellierung und Simulation;
- Implementierung der parallelen Simulatoren für PE-1,...,PE-4 entsprechend dem Devirtualisierungsvorgang;
- Effizienzanalyse der parallelen Simulatoren;
- Untersuchung der nach hybriden Programmiermodell realisierten Simulatoren;
- Entwicklung und Realisierung der BOF-, TA- und GG-Funktionen für die rechnergestützte Apriori-Analyse von PE-1,...,PE-4;
- Integration der VPSU mit den GRID-Technologien.