

# **DIE SUBSYSTEME EINER VERTEILTEN PARALLELEN SIMULATIONSUMGEBUNG FÜR DYNAMISCHE SYSTEME**

V.A. Svjatnyj, D.V. Nadeev, A.N. Solonin  
Fakultät für Rechentechnik und Informatik (FRTI)  
Nationale Technische Universität Donezk, Artemstr. 58, 83000, Ukraine  
svjatnyj@cs.dgtu.donetsk.ua

M. Zeitz  
Institut für Systemdynamik und Regelungstechnik (ISR), Universität Stuttgart  
Postfach 801140, 70511 Stuttgart

K. Rothermel  
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR),  
Universität Stuttgart,  
Breitwiesenstr. 20 – 22, 70565 Stuttgart

## **Abstract**

*Svjatnyj V.A., Nadeev D.V., Solonin A.N., Zeitz M., Rothermel K., Subsystems of a Distributed Parallel Simulation Environment for Dynamic Systems. The structure of the Distributed Parallel Simulation Environment (DPSE) is considered. The DPSE is utilized for the simulation of complex dynamic systems and consists of the simulation subsystem, data transfer subsystem and load-balancing subsystem. The simulation subsystem is represented by a parallel computer, which executes tasks received by means of the data transfer subsystem. Prior to execution the tasks are optimized in the load-balancing subsystem. The results of preliminary implementation and testing demonstrated the increase in the efficiency of parallel simulation.*

## **1. Einführung**

Im Rahmen einer langjährigen Kooperation von FRTI, IPVR und ISR wird eine verteilte parallele Simulationsumgebung (VPSU) für dynamische Systeme mit konzentrierten (DSKP) und mit verteilten (DSVP) Parametern entwickelt [1, 2]. Zur Zeit wird eine MIMD-Komponente der VPSU implementiert und die benötigte Simulationssoftware ist in der Entwicklung und Erprobung. In diesem Beitrag werden zwei wichtige VPSU-Subsysteme sowie erste Implementierungsergebnisse dargestellt.

## **2. Struktur und Dekomposition der VPSU**

Bisher wurde eine Version der Hardwarestruktur für die verteilte parallele Simulationsumgebung realisiert (Abb.1). VPSU beinhaltet die parallelen Rechenressourcen des HLRS der Universität Stuttgart (CRAY T3E-900/512, HITACHI SR8000), zwei Cluster und einen Intel-Paragon-Parallelrechner an der FRTI sowie eine Zugriffsumgebung mit IPVR-Server und HLRS-Fileserver für die Modellentwickler und Modellbenutzer in Donezk. Die VPSU-Struktur gehört zu den heterogenen örtlich verteilten Rechensystemen. Für die systemtechnische Organisation der VPSU-Funktionalitäten wird die Dekomposition der Umgebung auf einige Subsysteme vorgeschlagen. Nachfolgend werden zwei Subsysteme zum Datenaustausch bzw. zur Lastbalancierung vorgestellt.

## **3. Datenaustausch-Subsystem (DASUS)**

Dieses Subsystem wird als eine Teilkomponente der Hardware, Systemsoftware und Simulationssoftware definiert, die den benutzerfreundlichen und modellierungsbedingten Datenaustausch zwischen MPSU-Komponenten auf allen Modellierungs- und Simulationsetappen der parallelen Simulation von dynamischen Systemen realisiert.

**DASUS-Hardware.** Das Subsystem umfasst die in Abb.1 dargestellten Rechenressource und folgenden Kanäle zur Informationsübertragung:

- Arbeitsplatzrechner des Modellentwicklers (APRME) mit den vorhandenen parallelen Ressourcen des lokalen Netzes (Cluster-1, Cluster-2, Intel Paragon) und des globalen Netzes (SIMD- und MIMD-Rechner und Cluster in Stuttgart);
- Verbindungsnetzwerke zum Datenaustausch zwischen den Prozessoren, Prozessorelementen und Cluster-Rechnern der parallelen Ressourcen;
- Verbindungskanäle zwischen den Parallelrechnern und Clustern als VPSU-Einheiten;
- Kommunikationskanäle zwischen Modellentwicklern, die ein gemeinsames Simulationsproblem bearbeiten (lokales Netz) oder an Kooperationsprojekten (APRME-IPVR, APRME-ISR, APRME-FRTI) beteiligt sind;
- Kanäle zur Verbindung zwischen problemorientierten Simulatoren und realen Objekten;
- Zugriffskanäle für Industriebenutzer auf parallele Modelle eines Simulations- und Service-Zentrums [3].

**Anforderungen auf die DASUS-Organisation.** Ausgangspunkt des funktionellen Entwurfs von DASUS bilden die folgenden Forderungen, welche die komplizierte dynamische Systeme auf die Simulationsmittel stellen:

- DASUS-Benutzerfreundlichkeit ist als einen Entwicklungsstil zu verstehen, der den Modellentwicklern erlaubt, sich auf die Gegenstandsgebietaufgaben der effizienten parallelen verteilten Simulation von DSKP, DSVP zu konzentrieren. Dieser Stil setzt auch voraus, dass die weitere Anforderungen oft nur kompromißweise erfüllbar sind.
- DASUS soll erlauben, die real komplizierte dynamische Systeme zu modellieren und simulieren; dabei sollen die Topologieeigenschaften der DSKP, DSVP in der Datenaustauschorganisation widerspiegelt werden.
- Lösung der Echtzeitprobleme bei der modellgestützten Entwicklung von DSKP, DSVP: die DASUS-Organisation soll die Mittel für die Echtzeitsimulation sowie die rechtzeitige Erkennung der möglichen

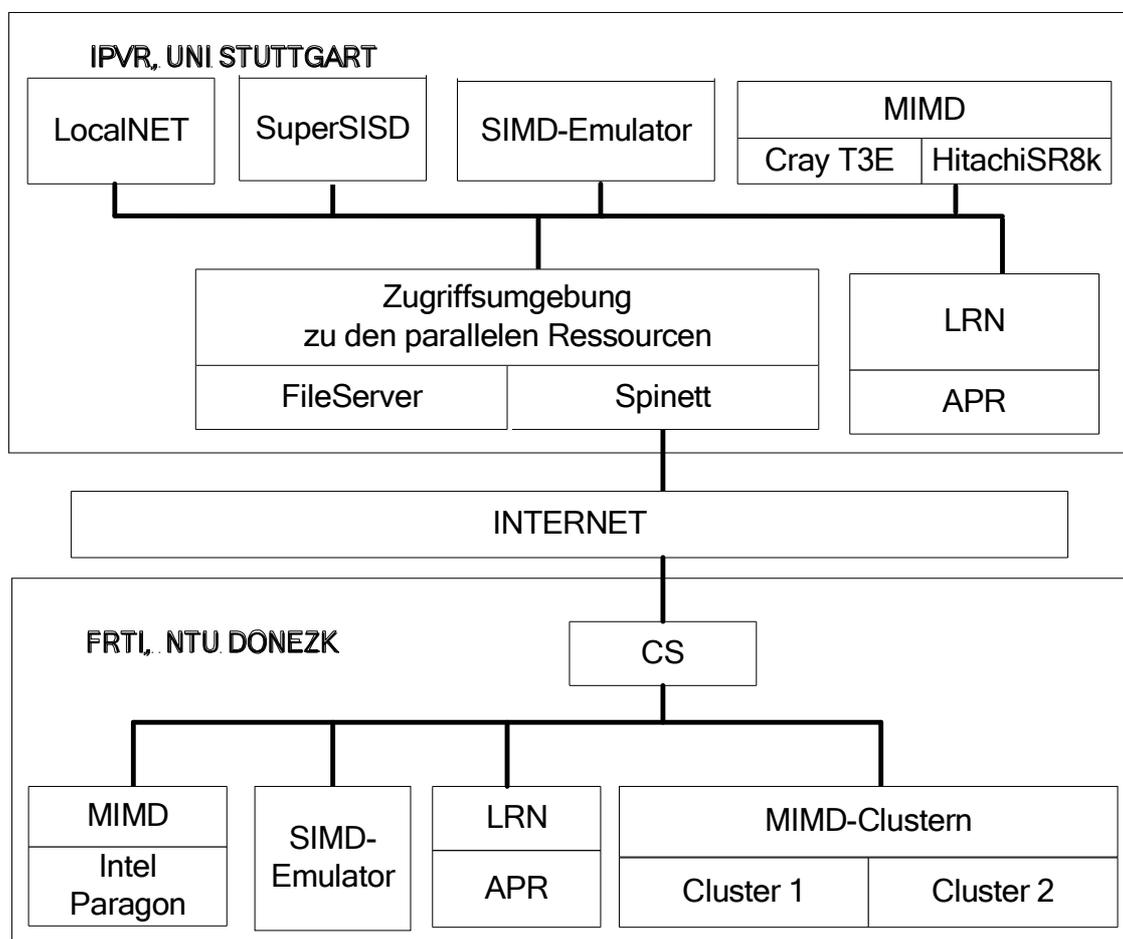


Abb.1. Eine Version der Hardwarestruktur fuer die VPSU

Abweichungen der Simulationsprozesse von den Echtzeitbedingungen vorzusehen.

- Vorhandensein der Spezifikationsmittel, die dem Modellentwickler schon auf der Gegenstandsgebiete eine virtuelle Beschreibung der simulationsbedingten Kommunikationsprozesse erlauben, sich mit

parallelen Simulationssprache integrieren und vom parallelen Compiler devirtualisiert werden.

- Dialogunterstützte Zugänglichkeit des Subsystems auf allen Etappen der Modellerstellung und Simulation.
- Hochentwickelte Testbarkeit des Subsystems: es ist vorzusehen, die Testen vor und nach der Lösung der Simulationsaufgaben durchzuführen.
- Vorhandensein des Schulungsprogramms, das die Benutzern schnell und qualitativ über alle Möglichkeiten der VPSU-Datenaustauschorganisation informiert und aktiv ausbildet.
- Implementation des Subsystems soll unter Berücksichtigung einer Plattformunabhängigkeit, der aktuellen verteilten Netztechnologien, der Integration mit dem MPI-Standard und den vorhandenen parallelen Programmiersprachen sowie des objekt-orientierten Ansatzes durchzuführen.

**DASUS-Funktionalität.** Entsprechend den obigen Definition, Hardwarestruktur und Anforderungen wird vorgeschlagen, die DASUS-Funktionen in folgenden Funktionsgruppen zu vereinigen:

**FG1** – die Funktionen, die den Zugang von Modellentwickler zu verschiedenen VPSU-Ressourcen erfüllen (einziges Ressource, gleichzeitig zu mehreren Ressourcen, zu inneren Strukturen der Ressourcen, zu den Arbeitsplätze anderen Modellentwicklern/Benutzern).

**FG2** – Datenaustausch in der MIMD-Komponente der VPSU (MIMD-Rechner und MIMD-Clustern).

**FG3** – Datenaustausch in der SIMD-Komponente der VPSU.

**FG4** – Datenaustausch zwischen den VPSU-Komponenten.

**FG5** – Testfunktionen des Subsystems von einzelnen Testbeispiele bis des globalen Testmodells mit den Beteiligung von allen DASUS-Komponenten.

**FG6** – das Sammeln von Simulationsergebnisse.

**FG7** – rechnergestützte Analyse der Effizienz von Daten- und Informationsaustausches unter Berücksichtigung der Modelleigenschaften und möglichen Varianten der Kommunikation zwischen den beteiligten VPSU- und DASUS-Ressourcen.

**FG7** – Schulung der Benutzern/Modellentwicklern.

**FG8** – DASUS-Präsentation.

**FG9** – Lösung der Echtzeitsimulationsaufgaben, ein Datenaustausch mit den realen Geräten bei der modellgestützten Projektierung von Automatisierungssystemen.

**DASUS-Systemsoftware** basiert sich auf den Betriebssysteme der VPSU-Komponenten und vorhandenen Mitteln der Systemunterstützung des Zugangs von räumlich entfernten und naheliegenden Arbeitsplatzrechnern.

**DASUS-Simulationsssoftware** ist wichtiger Bestandteil der VPSU-Simulationsssoftware [2] und wird aufgrund der **FG**-Algorithmen entwickelt.

#### **4. Lastbalancierungs-Subsystem**

**Die Lastquellen und Lastarten für VPSU-Ressource.** Die Lasten für die VPSU-Ressource entstehen an jedem APRME in Form von Modellierungs- und Simulationsaufträgen, die die folgende Punkte umfassen:

- Zugriffsinformation;
- Modellspezifikation mit vier möglichen Spezifikationsarten für parallele DSKP- und DSVP-Modelle: anwendungsnahe Modellspezifikation in der gegenwärtig entwickelten parallelen objektorientierten Simulationssprache (OO-Modellspezifikation); die block- (BO) oder gleichungsorientierte (GO) Spezifikation, wie diese in konventionellen Simulationssprachen vorhanden sind und für parallele Modellierungsverfahren angepasst werden; die Modellspezifikation in Form des in der parallelen Programmiersprache geschriebenen Modell-Codes;
- Auftrag für spezifikationsabhängige Modellierungsarbeiten; die Erfüllung des Auftrages beginnt mit dem Debugging der Modellfragmente und endet mit der komplexen Erprobung des gesamten parallelen DS-Modells;
- Auftrag für Simulationsuntersuchungen beinhaltet lauffähiges Modell und Plan für die Simulationsexperimente mit den relevanten Daten.

Die Lastarten für VPSU-Ressource und Inhalt der durchgeführten Rechenarbeiten sind von der formalen Beschreibung der komplizierten dynamischen Systeme (DSKP, DSVP) und der oben genannten Arten der Modellspezifikation abhängig. Die Arbeiten werden entsprechend der folgenden Etappen der Erstellung von parallelen Modellen durchgeführt.

Die formale DSKP-, DSVP-Beschreibung beinhaltet den topologischen Teil und eine mathematische Beschreibung in Form der algebraischen, gewöhnlichen und partiellen Differentialgleichungssysteme [1]. Der Aspekt der Kompliziertheit von dynamischen Systemen wird in der Topologiebeschreibung widerspiegelt und bei der rechnergestützten Erstellung der DSKP-, DSVP-Strukturen berücksichtigt. Diese Etappe wird mit den Mittel von VPSU-Benutzeroberfläche (BOF) auf APRME erfüllt und hat das Ziel – die Arbeitsumfang zu minimieren und mögliche Fehler der Strukturierung und Parametereingabe zu vermeiden. Die erstellte Struktur wird von dem Topologieanalysator bearbeitet, um die für die Gleichungsgenerierung notwendige topologische DSKP-, DSVP-Charakteristiken zu bekommen [1,2].

Die Gleichungsgenerierung wird in zwei Phasen erfüllt: das Ergebnis der ersten Phase ist die Matrix-Vektor-Darstellung der Gleichungssysteme in der Form, die alle numerische Lösungsverfahren brauchen; in der zweiten Phase wird geeignetes numerisches Verfahren ausgewählt, welches zur zusätzliche Topologie des diskretisierten DSKP, DSVP und zum diskreten Simulationsmodell führt; das Ergebnis ist die numerisch parallel lösbares Gleichungssystem in der Matrix-Vektor-Darstellung [1,2].

Topologieanalytoren und Gleichungsgeneratoren können an den APRME, SISD-, SIMD- und MIMD-Servern (Abb.1) implementiert werden.

Die numerische Lösung der Gleichungssysteme von parallelen DSKP-, DSVP-Modelle wird in den SIMD- oder MIMD-Gleichungslösern durchgeführt. Es gibt die komplizierte DSVP, für welche ist effektiv, die kombinierte SIMD/MIMD-Lösern mit der Verwendung von SPMD-Algorithmen zu bauen.

Debugging der parallelen Modelle als eine Lastart der VPSU-Ressourcen hat den veränderlichen Charakter – von der Erprobung der Modellfragmente auf dem APRME mit Hilfe von den SIMD- und MIMD-Emulatoren bis zu den Simulationsexperimenten auf den parallelen Zielrechnern. Eine Besonderheit dieser Arbeiten ist die aktive Nutzung der BOF-Mittel (des VPSU-Dialogsubsystems). Die lauffähige parallele Modelle werden auf den APRME und/oder auf den File-Servern gespeichert.

Die Durchführung der Simulationsexperimente mit den fertigen parallelen Modelle und die Visualisierung der Simulationsergebnisse gehören zu den planmäßigen VPSU-Lasten. Besonders intensiver Last entsteht für die VPSU-Komponenten, die die Echtzeitaufgaben mit der Anbindung von realen Geräten und/oder Trainingssimulatoren lösen.

In den Wartungstermine oder ereignisbedingt werden die VPSU-Testarbeiten durchgeführt; sie werden auch als die Systemaufträge spezifiziert.

Analyse der Lastquellen und Lastarten führt zur

**Definition1:** *VPSU-Last ist die Gesamtheit der Modellierungs- und Simulationsaufträge der betrachteten vier Arten, die von den APRME für die Erfüllung übertragen sind und in den File-Servern des Zuganges zu den parallelen Ressourcen gespeichert werden.*

**Problem der Lastbalancierung in der VPSU.** Das Entwicklungskonzept der parallelen Simulationssprache (1.Art der Modellspezifikation) geht davon aus, dass den Modellentwicklern die virtuelle parallele Ressourcen zur Verfügung gestellt werden [1]. Die Gesamtheit der virtuellen Modelle bildet virtueller VPSU-Last. Prozess der Modelldevirtualisierung beinhaltet oben genannten Etappen mit dem Endziel – gleichmäßige Verteilung des VPSU-Lastes zwischen vorhandenen Ressourcen oder Lastbalancierung in der VPSU. Diese Devirtualisierungsaufgabe kann man mit Hilfe von Lastmodell lösen [5,6].

**Definition2:** *Ein VPSU-Lastmodell ist die verbal-formale Beschreibung der Devirtualisierungseigenschaften der virtuellen parallelen Modelle der komplexen dynamischen Systeme.*

Diese Eigenschaften der virtuellen Modelle werden auf zwei Ebene analysiert: auf der Programmebene sind die virtuelle Lasten als untrennbare DSKP-, DSVP-Modelle zu betrachten; auf

der Prozessebene werden die parallele Modellprogramme als die Prozessmenge betrachtet, die den entsprechenden Ressourcen zugeordnet und für diese die Lastquellen sind.

Die Lastbalancierung in der verteilten parallelen Simulationsumgebung gehört zur Problematik der Lastverteilung und Lastbalancierung in den Rechensystemen [5,6]. Es wird vorgeschlagen, ein VPSU-Lastbalancierungssystem zu entwickeln und zu untersuchen.

**Definition3:** *Subsystem der Lastbalancierung (SUBLAST) in der verteilten parallelen Simulationsumgebung ist eine Gesamtheit der systemtechnischen, algorithmischen, programmtechnischen und informationstechnischen Mittel, die eine gleichmäßige Verteilung der Auslastung von parallel funktionierten VPSU-Komponenten realisieren und der effizienten Simulation der komplexen dynamischen Systeme beitragen.*

**SUBLAST-Funktionalität.** Entsprechend der Definition soll Lastbalancierungssystem die folgende Funktionen erfüllen:

- Erstellung der Lastmodelle aufgrund von DSKP-, DSVP-Modellspezifikationen.
- Effizienzanalyse der parallelen virtuellen Modelle von dynamischen Systemen (DS).
- Devirtualisierung der DS-Modelle unter der Berücksichtigung der exemplarischen VPSU.
- Statische Balancierung der parallelen Implementierungen von Topologieanalysatoren, Gleichungsgeneratoren und Gleichungslösern.
- Effizienzanalyse der Lastbalancierung während der Simulationsexperimenten mit den devirtualisierten Modelle, Vergleich mit den berechneten Effizienzdaten.
- Anmeldung der APRME-Aufträge, Organisation der Reihenfolge der Bearbeitung.
- Berechnung des VPSU-Auslastung auf der Programmebene, eine Korregierung der Verteilung von VPSU-Ressourcen abhängig von der Auslastung.

**Anforderungen auf die SUBLAST-Organisation.** Alle Mittel der Lastbalancierung sollen dem Modellentwickler als ein Werkzeug für rechnerunterstützte Modellerstellung dienen. Im Subsystem sollen Lastbalancierungsaufgaben auf den APRME sowie auf den File-Servern gelöst werden, d.h. die zentralisierte und dezentralisierte Balancierung durchgeführt werden. Abhängig von den Simulationsaufgaben soll SUBLAST die Balancierungskriterien verändern. Am meistens wird Subsystem die statische Lastbalancierung durchführen, soll aber die Mittel der dynamischen Balancierung für Modelle mit variablen Strukturen beinhalten. Integration mit der parallelen Simulationssprache, mit dem Datenaustauschsystem und mit der Benutzeroberfläche wird als eine Voraussetzung der SUBLAST-Implementierung betrachtet.

**SUBLAST -Systemsoftware** basiert sich auf den vorhandenen Balancierungsmittel der VPSU-Komponenten.

**SUBLAST - Simulationssoftware** ist als Bestandteil der VPSU-Simulationssoftware [2] aufgrund der Lastbalancierungsansätze [5,6] und der Funktionen des Subsystems zu entwickeln.

## **5. Zusammenfassung und Ausblick**

Die beiden vorgestellten VPSU-Subsysteme zum Datenaustausch und zur Lastbalancierung beinhalten mit der Benutzeroberfläche integrierte Module, die einen aktiven Dialog der Modellentwickler bzw. Modellbenutzer mit den VPSU-Ressourcen erlauben. Die Entwicklung der Subsysteme wird dem Konzept von parallelen Simulationssprachen angeglichen. Erste Ergebnisse der Implementierung und die Lösung der Testaufgaben zeigen, dass die vorgestellten Datenaustausch- und Lastbalancierungs-Subsysteme zur Erhöhung der Effizienz und Benutzerfreundlichkeit bei der parallelen Simulation von dynamischen Systemen beitragen.

## **6. Literatur**

1. L. Feldmann, V. Svjatnyj, V. Lapko, E.-D. Gilles, A. Reuter, K. Rothermel, M. Zeitz: Parallele Simulationstechnik.- Problems of Simulation and Computer-Aided Design of Dynamic Systems. Collected Volume of scientific papers. Donetsk State Technical University, Donetsk, 1999, 9 – 19.
2. V. Svjatnyj, E.-D. Gilles, M. Zeitz, A. Reuter, K. Rothermel: Simulationssoftware für eine parallele Simulationsumgebung für dynamische Systeme. 14. Symposium ASIM'2000, Tagungsband, SCS, 2000, 235 – 240.
3. W. Bär, R. Galasov, V. Lapko, O. Moldovanova, A. Pererva, D. Rasinkov, V. Svjatnyj: Das Simulations- und Service-Zentrum für automatisierte Grubenbewetterungsnetze. 14. Symposium ASIM'2000, Tagungsband, SCS, 2000, 223 – 228.
4. D. Nadeev, K. Rothermel, V.Svjatnyj, A. Solonin, I. Stepanov, M. Zeitz: Verteilte parallele Simulationsumgebung. - Problems of Simulation and Computer-Aided Design of Dynamic Systems. Collected Volume of scientific papers. Donetsk State Technical University, Donetsk, 2001, 229 – 233 (ukrainisch).
5. W. Becker: Dynamische adaptive Lastbalancierung für große, heterogen konkurrierende Anwendungen. IPVR der Universität Stuttgart, Dissertation, 1995.
6. R.Pollak: Auswirkungen verschiedener Informationsebenen auf die Effizienz der dynamischen Lastbalancierung. IPVR der Universität Stuttgart, Dissertation, 1999.

Дата надходження до редколегії: 13.12.2003 р.