SCALABILITY AND USABILITY IN MAS

Christian Hüning (christian.huening@haw-hamburg.de)

Agenda

1. Einleitung
2. Anforderungen
3. Paper1 – PDES-MAS
4. Paper2 – Repast HPC
5. Paper3 – Vigueras
6. Ausblick & nächste Schritte

Einleitung

- Generische Skalierbarkeit von MAS Systemen weiter ungelöst
- Große Simulationen erforderlich:
  - Prognosen in gekoppelten sozio-XYZ Systemen
  - Masseneffekte sichtbar machen (Yamamoto et al., 2008)
- Arbeiten zu diesem Thema älteren Datums (<= 2008) oder mit eher speziellem Fokus (z.B. Vigueras, 2011)
- Skalierende Systeme teils schwer zugänglich bzgl.
  - Datenimport
  - Umsetzung eines Modells
  - Auswertung
  - Wiederverwendbarkeit

Anforderungen an modernes MAS

- Modularity and Reusability
  - Submodelle sollen wiederverwendbar sein (z.B. Waldwachstum)
- Information Integration
  - Daten verschiedener Quellen integrieren
  - Probleme: Datenmenge, Heterogenität, Qualität, Lücken…
- Scalability
  - System soll mit zusätzlicher Hardware idealerweise um konstanten Faktor schneller werden
  - Communication-to-computation Verhältnis
- Ease of Use
  - Entwicklung neuer Modelle durch Services unterstützt
  - DSL oder grafische Tools zur Modellentwicklung durch Domänenexperten
- Visualization
  - Effiziente, Echtzeit Visualisierung von Teilen der Simulation

Motivation Scale

- Bestimmte Systeme lassen erst ab gewisser Größe sinnvoll untersuchen:
  - Ökonomie
  - Mikro-Biologie
  - Klima
- Technische Anforderungen:
  - Simulation(sraum) zerteilbar & verteilbar
  - Load Balancing / Re-Partitionierung (statisch & dynamisch)
  - Optimiertes Messaging
- Agentenzahl hat Auswirkung auf Simulationsergebnisse
  - Gezeigt von ZASE (Yamamoto et al. 2008) anhand einer Auktionssimulation

Motivation Scale – ZASE (Yamamoto et al. 2008)

- Höhere Agentenzahl erzeugt andere Ergebnisse:

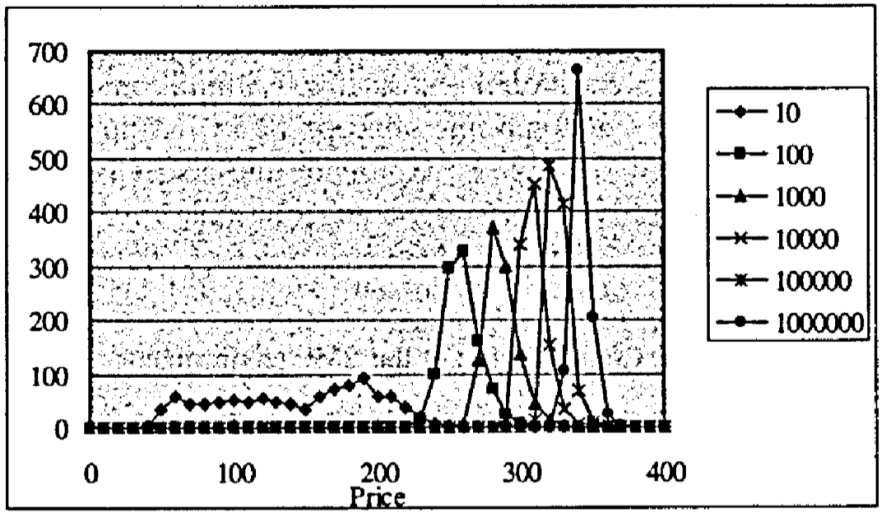

Fig. 5. Distribution of the final bid prices

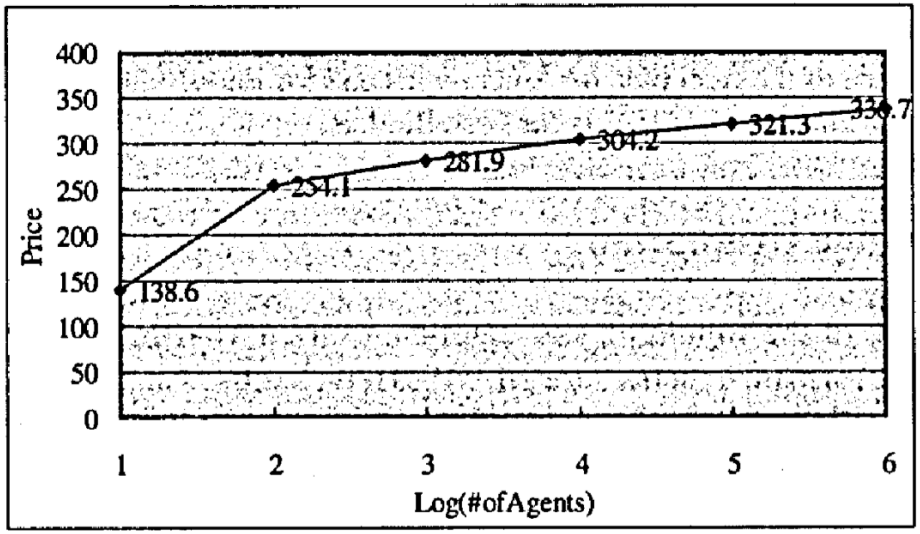

Fig. 4. Final bid prices

Paper 1 – PDES-MAS (Suryanarayanan et al. 2013)

- PDES: Parallel Discrete Event Simulation
- MAS als Logische Prozesse (LP)
- Agent LPs (ALP) modellieren Agentenverhalten
- Communication LPs (CLP) modellieren Kommunikation und Interaktion
- Shared State Variables (SSV) in den CLPs

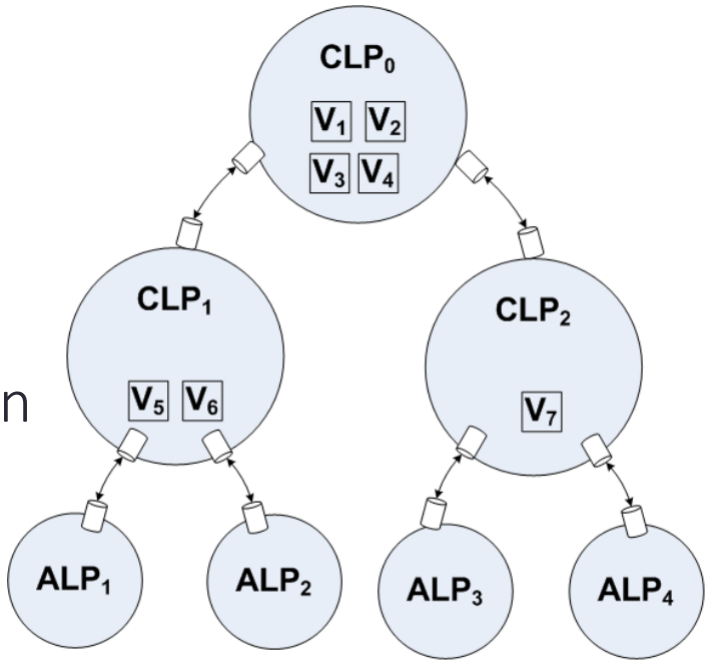

Fig. 1: A tree of 3 CLPs and 4 ALPs.

Paper 1 – PDES-MAS (Suryanarayanan et al. 2013)

- Routing ähnlich wie im DHT
- CLP Baum hat feste Größe
- SSVs werden dynamisch re-partitioniert je nach Zugriffsmuster (State Migration)
- ALPs werden RoundRobin ausgeführt & verwalten Local Virtual Time
- Rollbacks werden anhand der LVT durchgeführt

CLP

SSVs	Port
5,6	L
7	R
1,2,3,4	H

CLP	Port
0	H
1	L
2	R

ALPs	Port
1, 2	L
3, 4	R

(a) Routing table of CLP id 0.

CLP

SSV Values — Blocked / Not Blocked — Range Period List

SSV	LT_1	LT_2	LT_3	LT_4	LT_5
1	0				3
2	2		10		
3	5				
4	1		4		

Ports	LT_1	LT_2	LT_3	LT_4	LT_5
H	[0,5] RQ{0,2}, RQ{10,15}		[0,10] RQ{0,2}		[3,10] RQ{10,15}
U	[0,0] RQ{10,15}, RQ{0,2}			[0,0] RQ{0,2}, RQ{10,15}	
L	[21,25]	[21,27] RQ{10,15},RQ{0,2}			
R	[50,55] RQ{10,15},RQ{0,2}				

(b) An example of a Range Period List (at the right) of CLP 0 at ports H, L, U, R evolved over time.

Fig. 4: Routing information in PDES-MAS

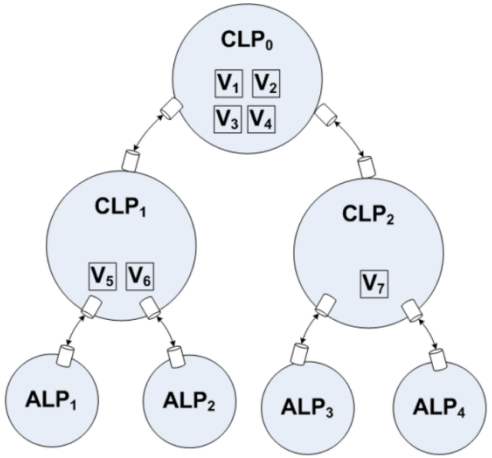

Fig. 1: A tree of 3 CLPs and 4 ALPs.

Paper 1 – PDES-MAS (Suryanarayanan et al. 2013)

Skalierbarkeit

- Vermutung: > ALP → Höhere Zugriffszahlen und Datendichte → abgefangen durch State Migration
- Hardware: BlueBEAR Cluster, Nodes mit 2x4Core 2.6Ghz, 8GB, Infiniband Netzwerk

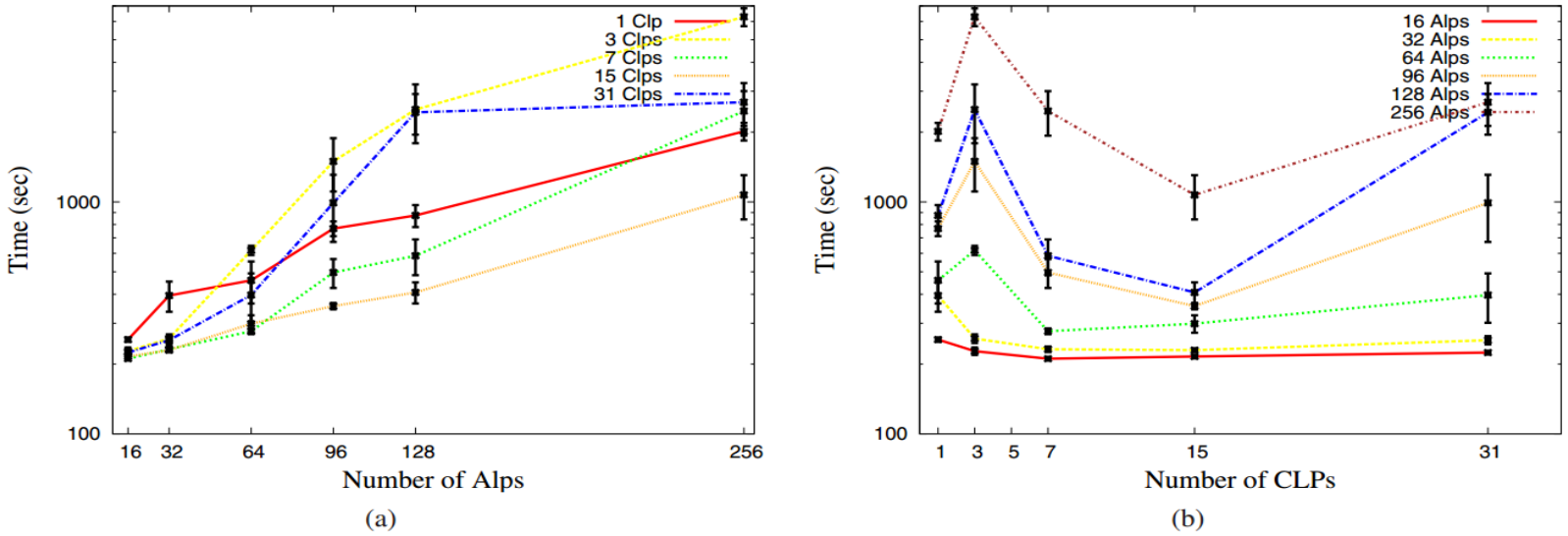

Fig. 8: Average Simulation Time for varying Number of ALPs and Number of CLPs with their standard deviations.

Paper 1 – PDES-MAS (Suryanarayanan et al. 2013)

Diskussion

- Idee ähnlich TupleSpaces (XAP o.ä.)
- Führt „nur“ Sense-Think-Act Agenten aus
- SSVs erlauben nur primitive Datentypen
- Datenimport →Abbildung ggf. komplex und aufwendig?!
- Skalierbarkeit → Ergebnisse gut, aber Hardware-Config unklar

- Gutes Verfahren zur Messung von Skalierbarkeit
- Wollen „Proxy Variables“ testen → „Agent Shadowing“

Paper 2 – Repast HPC (Collier et al., 2013)

- Basiert auf RepastJ bzw. Repast Symphony
- Motivation ähnlich der von MARS: Große Simulationen auf vielen Knoten, statt viele Durchläufe kleiner Simulationen.
- Optimization Simulation vs. Large-Scale model simulation
- Nutzt Logo-Sprache (ReLogo):
  - Agenten sind Turtles oder Patches
  - Observer
  - Interaktionen zwischen benachbarten Turtles/Patches

Paper 2 – Repast HPC (Collier et al., 2013)

- Agenten, Kontexte und Projektionen:
  - Agenten sind Agenten
  - Kontexte als Menge von Agenten
  - Projektionen nutzen Kontexte um Environments zu modellieren
- Kontexte und Projektionen erlauben Wiederverwendung
- Verteilung mittels Influence Sphere
  - Prozesse möglichst nach Lokalitätsverhalten der Agenten einteilen

Paper 2 – Repast HPC (Collier et al., 2013)

Verteilung & Kommunikation

- „Shared Projections“
- Buffer mit nicht-lokalen Nachbaragenten
- Änderungen zunächst lokal, Betroffene Prozesse werden dann informiert

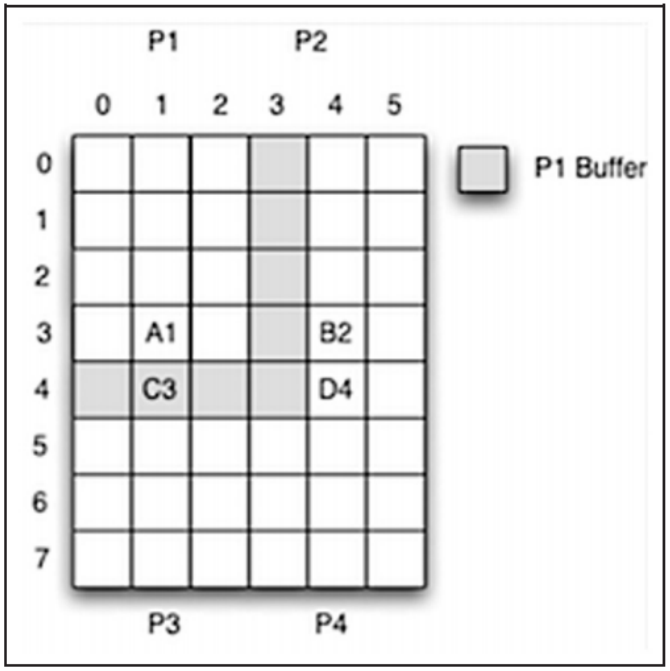

Figure 1. Shared grid buffer. Note that the grid section starting at row 0, column 0 and extending through row 3, column 2 runs on process P1. Section 0,3–3,5 runs on process P2 and so on for the other processes.

Paper 2 – Repast HPC (Collier et al., 2013)

Performanz & Skalierbarkeit

- Hardware: IBM Blue Gene/P 1048 & 40.000 Kerne

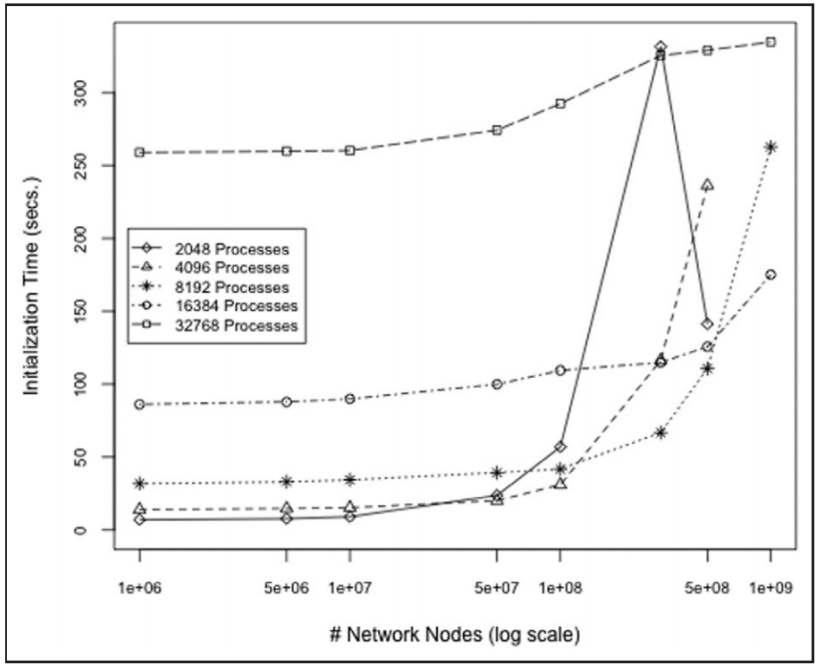

Figure 2. Initialization time by process count.

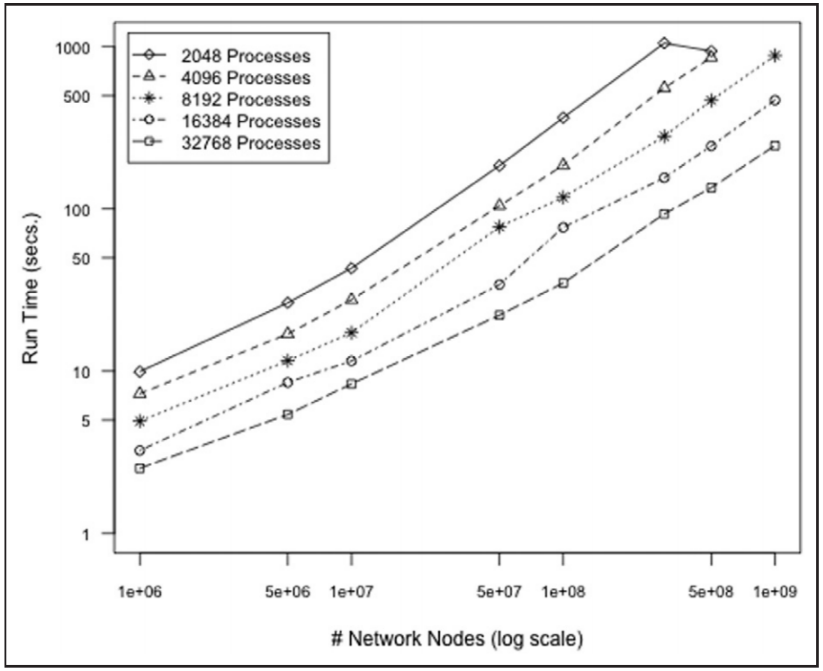

Figure 3. Runtime by process count.

Paper 2 – Repast HPC (Collier & North, 2013)

Diskussion

- Environment „nur“ 2D (Network, Grid, Continuous Space)
- Logo-Sprache birgt Einschränkungen bzgl. der Modelle
- Auswertung / Visualisierung nur über Textausgabe in Datei
- Datenimport / Integration nicht erwähnt
- Kommunikations- und Synchronisationsmechanismus erfordert Benutzereingriff → nicht transparent

Paper 3 – Vigueras (Vigueras et al. 2011)

- Entwickelt für interaktive Simulationen
- Massiv verteilt und asynchron
- 1 ActionServer je Region
- 1 ClientProcess je AS
- Semantische Datenbank als Kopie in jeder Station
- Synchronisation nur wenn nötig
- Jeder AS, CP und VCP auf eigenem Rechner

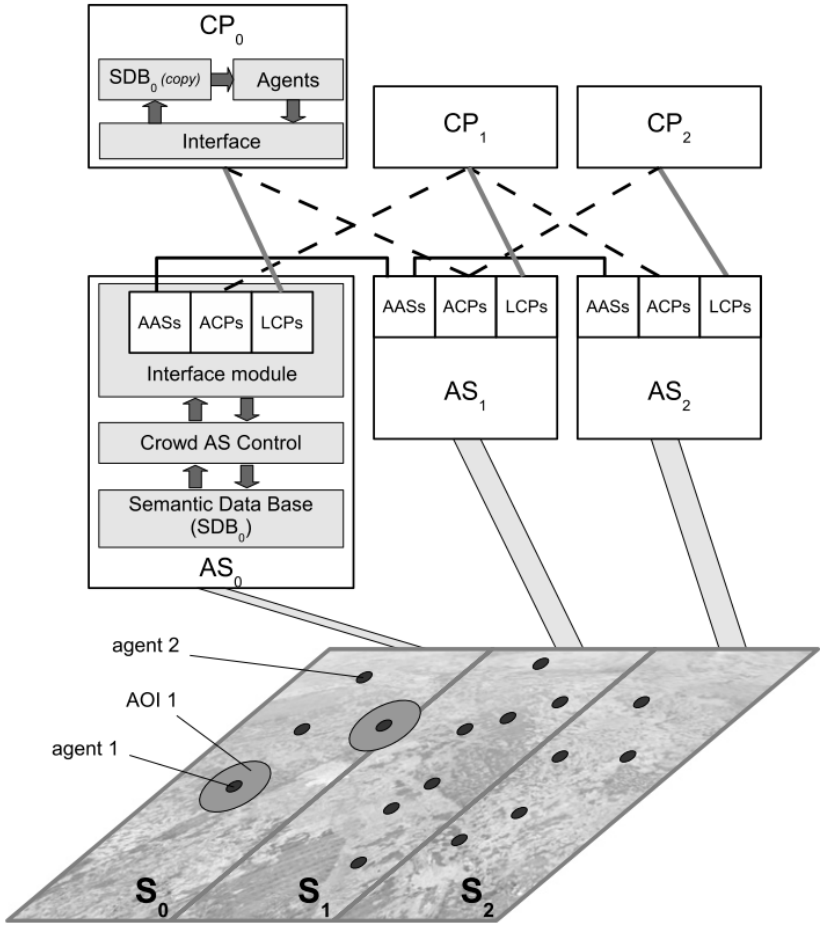

Fig. 1. General scheme of the distributed multiagent system architecture.

Paper 3 – Vigueras (Vigueras et al. 2011)

- Visualisierung ebenfalls verteilt
- VisualClientProcesses als Kameras mit Blick in die Simulation
- Skaliert sehr gut mit mehr Hardware

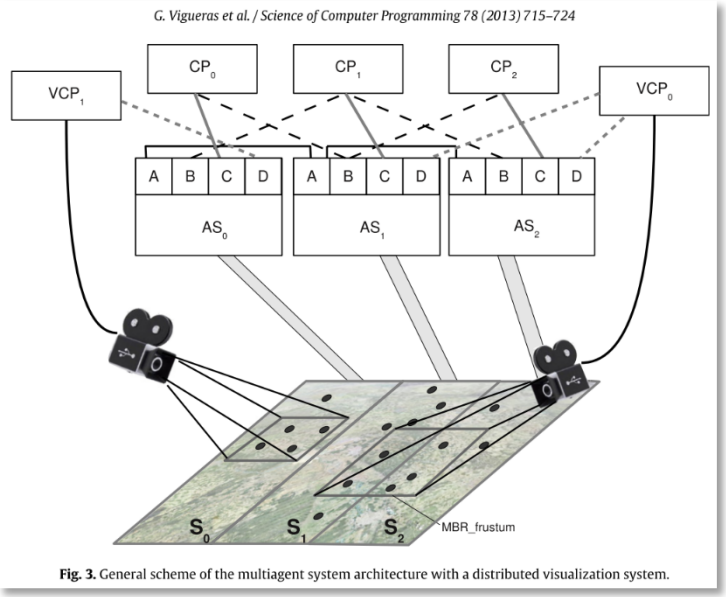

Fig. 3. General scheme of the multiagent system architecture with a distributed visualization system.

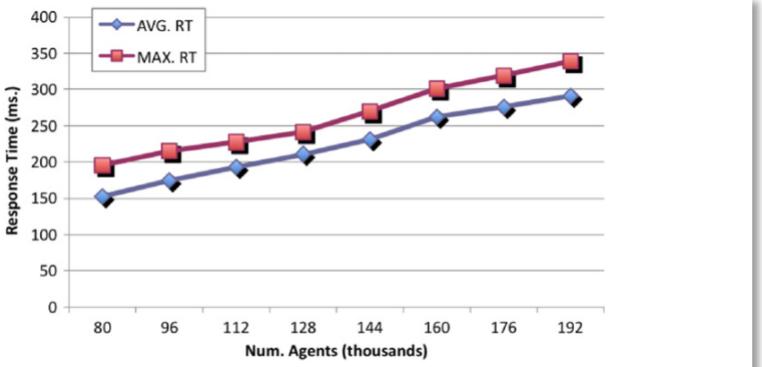

Fig. 5. Response times of the proposed architecture when implemented on a cluster of ninety six computing nodes.

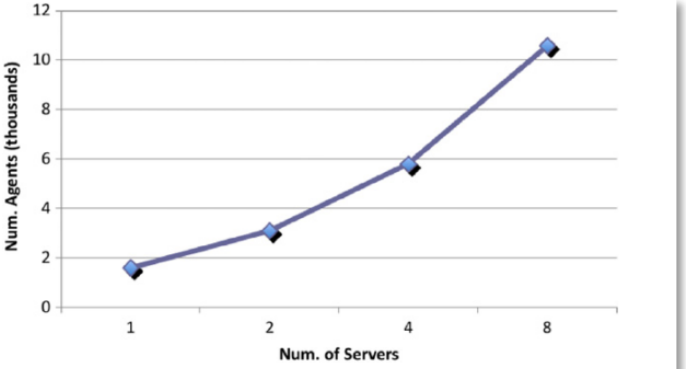

Fig. 6. Speedup achieved by the proposed architecture on a set of interconnected PCs.

Ausblick & nächste Schritte

- Vertical Slice ca. 80 % fertig
- Verteilung und Kommunikation für Agenten implementiert, erste Tests vielversprechend → Bestätigen Ergebnisse von Repast HPC
- Information Integration Systeme implementiert
- Prototyp der MARS DSL inkl. WebEditor fertig

Ausblick & nächste Schritte

ToDo:

- Erste Version mit Beispielszenario fertigstellen
- Partitionierungslösung entwickeln
- Daten aus Information Integration Lösung in LIFE nutzen
- Visualisierung anbinden und testen
- Lösungen für Zeitmanagement und Ausführungssteuerung testen
- Eigene Lösung mit Repast Symphony / HPC vergleichen

Vielen Dank für eure Aufmerksamkeit!

Fragen?

Quellen

- (Yamamoto et al., 2008) : A platform for massive agent-based simulation and its evaluation“, Massively Multi-Agent Technology. In Lecture Notes in Computer Science, Vol. 5043, pp 1-12, Jan. 2008.
- (Collier & North, 2013) : N. Collier und M. North, „Parallel agent-based simulation with Repast for High Performance Computing“, SIMULATION, Bd. 89, Nr. 10, S. 1215-1235, Okt. 2013.
- (Vigueras et al. 2011) : G. Vigueras, J. M. Orduña, M. Lozano, und Y. Jégou, „A scalable multiagent system architecture for interactive applications“, Science of Computer Programming, 2011.
- (Suryanarayanan et al. 2013) : V. Suryanarayanan, G. Theodoropoulos, und M. Lees, „PDES-MAS: Distributed Simulation of Multi-agent Systems“, Procedia Computer Science, Bd. 18, S. 671-681, 2013.