

Abstract

Space based computing is an alternative in comparison to the traditional message oriented approach for communication in distributed systems. eXtensible Virtual Shared Memory [XVSM] follows this paradigm. It conjuncts the positive of other Virtual Shared Memory [VSM] implementations like Corso or Linda and improves them.

The aim of this work is to develop an API for the windows platform which provides the concept of a virtual shared memory. It is a language binding in C# for an XVSM kernel and provides .Net developers a middleware to write distributed application in an easy way. It can be an alternative to other message oriented middleware APIs, like Windows Communication Foundation [WCF].

The API provides write and read access to arbitrary objects, which are managed in containers. This container and its objects will be published in a global shared virtual space which any user can gain to access. All containers have an own access principle, for example FIFO. Thereby transactional security will be guaranteed. Furthermore, notifications can be used to indicate changes of space objects. The API is open and configurable, for the possibility of extensions. So the API can be the base for arbitrary complex distributed applications.

Kurzfassung

Space Based Computing ist eine Alternative zu dem traditionellen nachrichtenorientierten Ansatz für die Kommunikation in verteilten Systemen. eXtensible Virtual Shared Memory [XVSM] folgt diesem Paradigma. Es orientiert sich an bereits vorhandenen Virtual Shared Memory [VSM] System wie beispielsweise Corso oder Linda und erweitert diese.

Das Ziel dieser Arbeit ist es ein API für die Windows Plattform zu entwickeln, was das Konzept des virtuell gemeinsamen Speichers anbietet. Es stellt eine Sprachanbindung in C# für einen XVSM Kernel dar und bietet somit .NET Entwicklern eine Middleware mit der man schnell und einfach verteilte Anwendungen schreiben kann. Diese Middleware kann konzeptionell ein alternativer Ansatz zu anderen nachrichtenorientierten Middleware APIs sein wie z.B., die Windows Communication Foundation [WCF]. Hierbei sei aber jedoch darauf hingewiesen, dass man XVSM/XtSpaces in seinem derzeitigen Status nicht annähernd mit der Mächtigkeit der WCF vergleichen kann. Es stellt aber konzeptionell einen interessanten anderen Ansatz dar.

Im Speziellen bietet das API Schreib- und Lesezugriff auf beliebige Objekte, die über Container verwaltet werden. Diese Container und deren Objekte werden in einem globalen virtuellen Raum allen Benutzern zu Verfügung gestellt. Alle Container besitzen Ihnen eigene Zugriffsprinzipien wie z.B. FIFO. Bei dem Zugriff auf Container wird transaktionale Sicherheit gewährleistet. Notifikationen können benutzt werden um Änderungen und Zugriffe auf den Container zu signalisieren. Das API ist konfigurierbar und hat offene Programmierschnittstellen. Somit ist es flexibel und erweiterbar.

Table of Content

Abstract	1
Kurzfassung.....	2
Danksagung.....	Fehler! Textmarke nicht definiert.
Table of Content.....	3
1 Introduction	6
1.1 Motivation & Goal	6
1.2 Structure of the thesis	7
2 Technical Background.....	8
2.1 Distributed Systems.....	8
2.2 Space Based Computing.....	10
2.2.1 Evaluation.....	10
2.2.2 Virtual Shared Memory Paradigm	11
3 Virtual Shared Memory Implementations	12
3.1 Linda.....	12
3.1.1 The Concept	12
3.1.2 Components.....	14
3.1.3 The API	15
3.2 JavaSpaces.....	18
3.2.1 The Concept	18
3.2.2 Components.....	20
3.2.3 The API	21
3.2.4 Comparison	23
3.3 Corso	24
3.3.1 The Concept	24
3.3.2 Components.....	24
3.3.3 The API	28
3.3.4 Comparison	28
4 XVSM	30
4.1 The Concept	30
4.2 Components.....	31
4.3 API	35
4.4 Comparison	35

Table of Content

5 XtSpaces.....	36
5.1 API Design Deliberations.....	36
5.2 Design and Architecture.....	38
5.3 XtSpaces Layers.....	41
5.3.1 XVSM Kernel.....	41
5.3.2 XVSM Kernel Wrapper.....	48
5.3.2.1 Managed and Unmanaged Code.....	48
5.3.2.2 Platform Invoke.....	49
5.3.2.3 Implementation.....	50
5.3.3 XVSM Kernel Provider.....	52
5.3.3.1 Contract.....	53
5.3.3.2 Implementation.....	56
5.3.4 XVSM Kernel Provider Extensions.....	56
5.3.4.1 Design.....	57
5.3.4.2 Connection Manager.....	58
5.3.4.3 Converter.....	59
5.3.4.4 Lookup Manager.....	61
5.3.4.5 Message Queue Manager.....	63
5.3.4.6 Notification Manager.....	67
5.4.5 XtSpaces API.....	71
5.4.5.1 XtSpace.....	71
5.4.5.2 XtSpace Container.....	73
5.4.5.3 Notification.....	77
5.4.5.4 Transactions.....	78
5.4.5.5 Additional Internal Classes.....	78
5.4 Examples.....	82
5.4.1 Simple Queue.....	82
5.4.2 Producer – Consumer.....	83
5.4.3 TicTacToe.....	85
5.5 Configuration & Deployment.....	87
6 Conclusion.....	89
References.....	91
Abbreviations.....	93
Appendix.....	94

Table of Content

Figure List	99
-------------------	----