

EINFÜHRUNG IN MESSAGE PASSING IN VERTEILTEN SYSTEMEN

Florian Willich

Hochschule für Technik und Wirtschaft Berlin
Kurs: Verteilte Systeme
Dozentin: Prof. Dr. Christin Schmidt

19. Juni 2015

Table of Contents

Einführung

theoretische Anforderungen

synchroner und asynchroner Nachrichtenaustausch

Definition

Die Programmiersprache Erlang

Demonstration

References

Was ist Message Passing?

Was ist Message Passing?

Was ist Message Passing in verteilten Systemen?

(Tanenbaum & Steen, 2007)

- ▶ Verbindung

- ▶ Verbindung
- ▶ Befähigung

- ▶ Verbindung
- ▶ Befähigung
- ▶ Vollständigkeit

- ▶ Verbindung
- ▶ Befähigung
- ▶ Vollständigkeit
- ▶ Verständlichkeit

- ▶ Verbindung
- ▶ Befähigung
- ▶ Vollständigkeit
- ▶ Verständlichkeit

Ausführbarkeit ist nicht teil des Message Passing Modells!

- ▶ **synchron**

Kommunikations-Primitive werden direkt aufgerufen.

- ▶ **synchron**

Kommunikations-Primitive werden direkt aufgerufen.

- ▶ **asynchron**

Eine Dienstschicht (Middleware) wird eingeführt (z.B. Message Queues).

(Tanenbaum & Steen, 2007)

Message Passing in verteilten Systemen ist ein Modell um Nachrichten innerhalb eines Prozesspaares auszutauschen. Es legt fest wie die Verbindung hergestellt, Nachrichten gesendet, empfangen und verstanden werden. Wie dies geschieht hängt stark von den genutzten Standards und der Implementierung ab. Das zum Einsatz kommende Modell kann sich dabei extrem von anderen hinsichtlich der bereitgestellten Funktionalität unterscheiden.

Message Passing in verteilten Systemen ist ein Modell um Nachrichten innerhalb eines Prozesspaares auszutauschen. Es legt fest wie die Verbindung hergestellt, Nachrichten gesendet, empfangen und verstanden werden. Wie dies geschieht hängt stark von den genutzten Standards und der Implementierung ab. Das zum Einsatz kommende Modell kann sich dabei extrem von anderen hinsichtlich der bereitgestellten Funktionalität unterscheiden.

Die physikalischen Gegebenheiten sowie die Ausführung der gewünschten Instruktionen ist nicht teil des Message Passing Modells.

- ▶ funktional

Die Programmiersprache Erlang

- ▶ funktional
- ▶ deklarativ

- ▶ funktional
- ▶ deklarativ
- ▶ konzipiert für nebenläufige, sehr große Programme die im Dauerbetrieb und in Echtzeit in einem verteilten System laufen

(Armstrong, 1996)

(Däcker, 2000)

- ▶ Funktion `spawn`(*Module, Exported Function, List of Arguments*)

- ▶ Funktion `spawn`(*Module, Exported Function, List of Arguments*)
- ▶ Konstrukt `receive`

- ▶ Funktion `spawn`(*Module, Exported Function, List of Arguments*)
- ▶ Konstrukt `receive`
- ▶ Operator `!`

- ▶ Funktion `spawn`(*Module, Exported Function, List of Arguments*)
- ▶ Konstrukt `receive`
- ▶ Operator `!`
- ▶ Funktion `self()`

(Ericsson AB, 2015)

(Hebert, 2013)

- ▶ Verbindung: Erlang Laufzeitsystem stellt Verbindung her (Operator !)

Message Passing Modell von Erlang

- ▶ Verbindung: Erlang Laufzeitsystem stellt Verbindung her (Operator !)
- ▶ Befähigung: Message Queues (Konstrukt `receive`)

Message Passing Modell von Erlang

- ▶ Verbindung: Erlang Laufzeitsystem stellt Verbindung her (Operator !)
- ▶ Befähigung: Message Queues (Konstrukt `receive`)
- ▶ Vollständigkeit: Bewerkstelligt das Erlang Laufzeitsystem

Message Passing Modell von Erlang

- ▶ Verbindung: Erlang Laufzeitsystem stellt Verbindung her (Operator !)
- ▶ Befähigung: Message Queues (Konstrukt `receive`)
- ▶ Vollständigkeit: Bewerkstelligt das Erlang Laufzeitsystem
- ▶ Verständlichkeit: Erlang arbeitet mit Pattern Matching

Diffie-Hellman Schlüsselaustauschalgorithmus implementiert in Erlang
mit Hilfe von Message Passing

Auch zu finden unter: https://github.com/c-bebop/message_passing

References

Armstrong, Joe. 1996.

Erlang - A survey of the language and its industrial applications. In: Proceedings of the symposium on industrial applications of Prolog (INAP96).

<http://www.erlang.se/publications/inap96.ps>.

The 9th Exhibitions and Symposium on Industrial Applications of Prolog. 16-18, October 1996. Hino, Tokyo Japan. [Online. Accessed 5th May 2015].

Däcker, Bjarne. 2000.

Concurrent functional programming for telecommunications: A case study of technology introduction.

<http://www.erlang.se/publications/bjarnelic.ps>.

Licentiate Thesis, Department of Teleinformatics. TRITA-IT AVH 00:08, ISSN 1403-5286. Royal Institute of Technology, Stockholm, Sweden. [Online. Accessed 20th May 2015].

Ericsson AB. 2015.

Erlang/OTP System Documentation 6.4.

<http://www.erlang.org/doc/pdf/otp-system-documentation.pdf>.

[Online. Accessed 31th May 2015].

Hebert, Fred. 2013.

Learn You Some Erlang for Great Good! : A Beginner's Guide.

<http://learnyousomeerlang.com/content>.

No Starch Press. [Online. Accessed 20th May 2015].

Tanenbaum, Andrew S., & Steen, Maarten Van. 2007.

Distributed Systems: Principles and Paradigms (Second Edition).

Pearson Prentice Hall.

ISBN 0-13-239227-5.