

Objektorientierte Modellierung, Spezifikation und Implementierung (OMSI) I

Praktikum

Prof. Joachim Fischer / Dr Klaus Ahrens

Dr. Markus Scheidgen

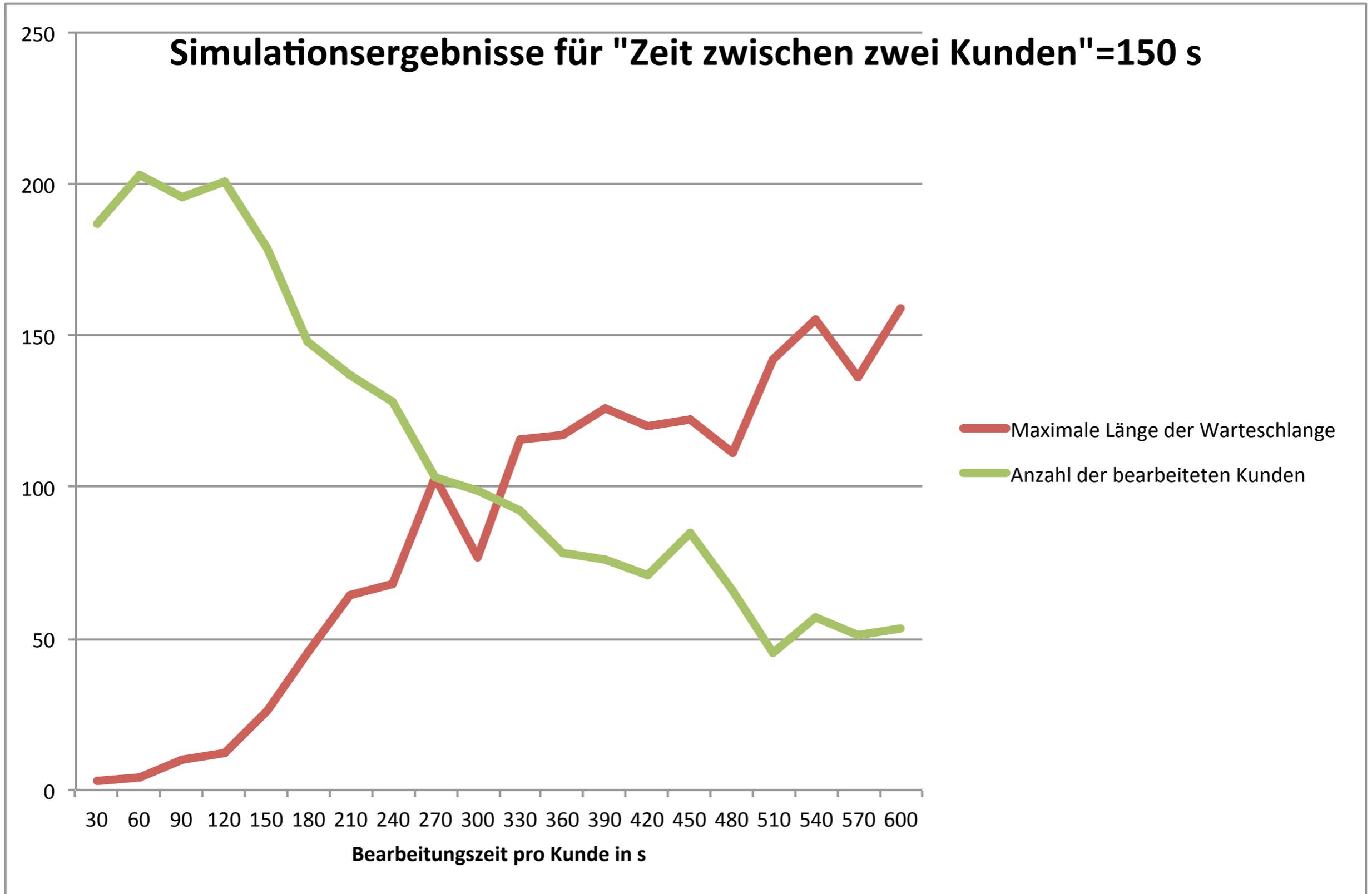
{fischer,ahrens,scheidge}@informatik.hu-berlin.de

LFE Systemanalyse, III.310

Aufgaben zur Veranstaltung

- ▶ <http://www.informatik.hu-berlin.de/forschung/gebiete/sam/Lehre/omsi>
- ▶ Fragen: per Mail über Goya

Lösung Aufgabe 3



Aufgabe 4

Implementieren Sie eine **M/M/c Warteschlange** mit **ODEMx**

Aufgabe 4 – Ziele

- ▶ Folgere ODEMX Konzepte/Klassen sollen geübt werden
 - `odemx::base::Simulation`
 - `odemx::base::Process`
 - `odemx::synchronization::Bin`
 - `odemx::random::Negexp`

M/M/c Warteschlange (1)

- ▶ klassische Warteschlange aus der Stochastik (speziell Warteschlangentheorie)
- ▶ mathematisch schwer zu bestimmende Eigenschaften
➔ Simulation
- ▶ Klassifizierung nach D.G.Kendall
 - A/S/c/N/K/D
 - ◆ Arrival
 - ◆ job **S**ize
 - ◆ count of servers
 - ◆ **K**apacity = ∞
 - ◆ population (**N**) = ∞
 - ◆ queueing **D**iscipline = *FIFO*



M/M/c Warteschlange (2)

$$P(x) = \lambda e^{-\lambda x}$$

$$\text{CDF } P(X \leq x) = 1 - e^{-\lambda x}$$

$$\text{Erwartungswert} = 1/\lambda$$

► M

➔ Markov Eigenschaft

➔ Ankünfte pro Zeitraum Poisson verteilt

➔ Abstand zwischen zwei Ankünften exponential verteilt

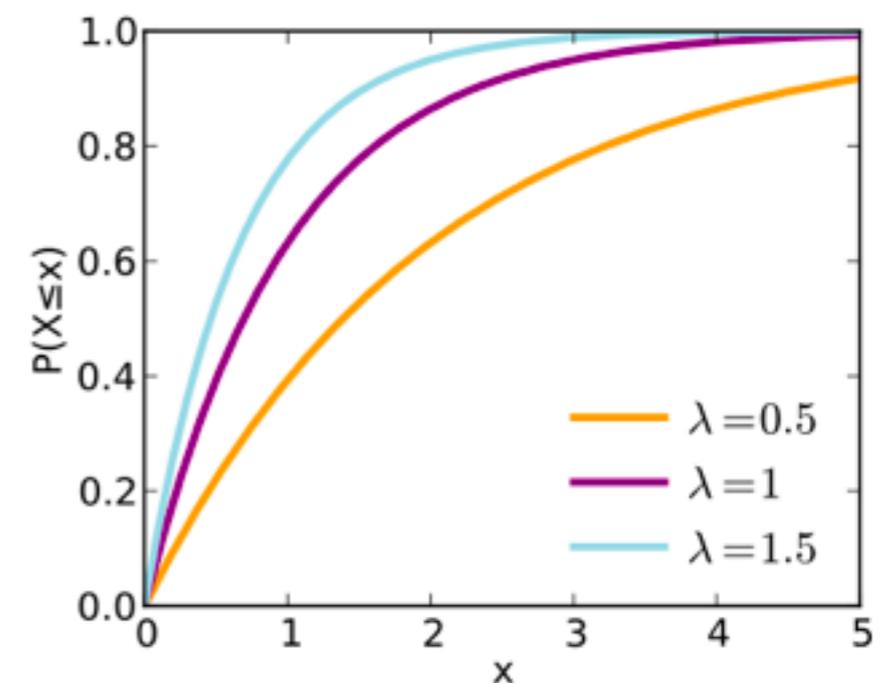
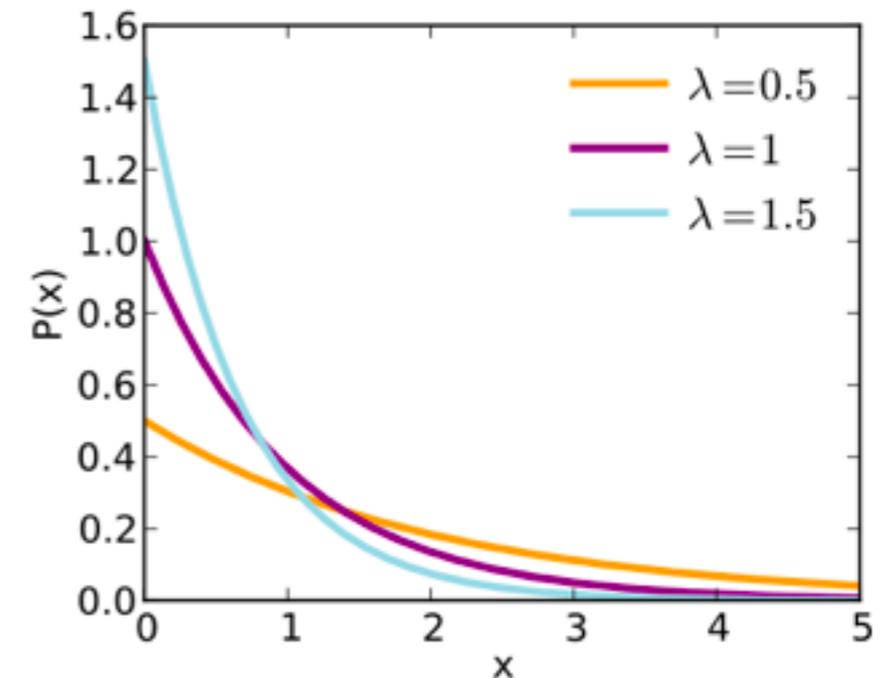
► M

➔ Markov Eigenschaft

➔ Bearbeitungszeit Exponential verteilt

► 1

➔ es gibt genau einen Schalter/Server



M/M/c Warteschlange (3)

- ▶ gegeben mittlere Ankunftszeit a , mittlere Bearbeitungszeit m , Dauer der Öffnungszeit t , Anzahl der Bearbeiter c
- ▶ gesucht:
 - maximale Länge der Warteschlange
 - Anzahl der Bearbeiteten Kunden
 - Summe der Lehllaufzeiten der Bearbeiter

ODEMx (1)

- ▶ <http://blog.hilbri.ch/wp-content/uploads/2008/04/odemx.pdf>
- ▶ Make & Co: siehe 2. Praktikumsveranstaltung
- ▶ Eclipse ohne cmake -G "Eclipse CDT4..."
- ▶ Namensräume beachten

ODEMx (2)

- ▶ `odemx::base::Simulation`
 - `virtual initSimulation() = 0`
 - `runUntil(SimTime)`
 - `SimTime getTime()`
- ▶ `odemx::base::Process(Simulation&, Label)`
 - `virtual int main()`
 - `activate()`
 - `holdFor(SimTime)`
- ▶ `odemx::synchronization::Bin(Simulation&, Label, int)`
 - `give(int)`
 - `take(int)` - blockierend, wenn nicht genug Tokens zur Verfügung stehen
- ▶ `odemx::random::NegativeExponential(Simulation&, Label, double)`
 - `double sample()`
- ▶ `odemx::data::Label -> std::String`
- ▶ `odemx::base::SimTime -> some integer type`

Aufgabenstellung

- Implementieren Sie Ihr Modell der M/M/1 Warteschlange aus der Lösung der letzten Aufgabe in ODEMX. Verwenden Sie dabei die ODEMX Klassen Simulation, Prozess, Bin und NegativeExponential.
- Führen Sie die Parameterstudie der letzten Aufgabe auf der neuen Implementierung durch.
- Vergleichen Sie die Ergebnisse aus beiden Aufgaben. Nennen Sie Gründe für Abweichungen. Fakultativ: wandeln Sie die Lösung der letzten Aufgabe so ab, dass genau die gleichen Simulationsergebnisse geliefert werden.
- Erweitern Sie Ihre Implementierung, so dass eine beliebige Anzahl an Bankangestellten Kunden abarbeiten. Wiederholen Sie die Parameterstudie mit einem zusätzlichen Parameter $c=1,..5$. Ermitteln Sie die Gesamtleerlaufzeit der Bankmitarbeiter. Was sind abhängig von a und m eine gute Anzahl von Bankmitarbeitern

Abgabe

► Archiv mit

- Verzeichnis mit C++ Projekt
 - ◆ mm1.h, mm1.cpp (6 Punkte)
 - ◆ Ihr Programm zur Parameterstudie (1 Punkte)
 - ◆ Makefile (inkl. run-Task)
- PDF mit Erklärungen (3 Punkte)

▶ nächstes Praktikum, nächste Woche

Fragen?