

Fakultät:
Informatik/Mathematik

Thema: Entwicklung einer performanten
zentralen Produktionsplanung zu
Vergleichs- und Integrationszwecken

Verfasser: Pascal Schumann
Betreuer: Prof. Dr. rer. pol. Torsten Munkelt
M.Sc. Martin Krockert

Einführung:

Im Rahmen zweier aktueller Forschungsprojekte entwickeln zwei wissenschaftliche Mitarbeiter ein Konzept und einen Prototyp für eine sich selbst organisierende Produktion (SSOP). Die Untersuchung der SSOP hinsichtlich ihrer Qualität und ihrer Performanz erfolgt anhand eines parametrisierbaren ereignisdiskreten Simulationsmodells einer Produktion. Um die SSOP mit einer zentral geplanten Produktion zu vergleichen und die zentral geplante Produktion mit der SSOP zu kombinieren, ist eine performante zentrale Produktionsplanung (ZPP) zu entwickeln, die auf dem oben erwähnten Simulationsmodell aufsetzt, auf denselben Datenstrukturen wie die SSOP basiert und in die SSOP integriert werden kann.

Anforderungen:

Die ZPP muss den MRP-Lauf (MRP II) abbilden. Die ZPP muss eine Neuplanung unter Verrechnung von Rückmeldungen durchführen. Die ZPP muss die Simulation und die Datenstrukturen der SSOP nutzen. Die ZPP muss eine gute Struktur durch Befolgung der Regeln der Objektgymnastik aufweisen. Die ZPP muss in die SSOP integrierbar sein. Es muss eine empirischen Studie zur Überprüfung der Performanz durchgeführt werden.

Anforderungen an den MRP-Lauf:

Der MRP-Lauf muss eine Stücklistenauflösung mit Produktionsauftrags- und Bestellanlage aufweisen. Die ZPP muss eine Schnittstelle für die Losgrößenbildung online und offline anbieten. Online bedeutet während der Planung. Offline bedeutet nach der Planung. Der MRP-Lauf muss die Durchlaufterminierungen nach dem Schema rückwärts-vorwärts-rückwärts durchführen. Die ZPP muss eine Schnittstelle anbieten, um die Durchlaufterminierungen offline durchzuführen. Der MRP-Lauf muss eine Maschinenbelegung mit dem Giffler-Thompson-Algorithmus und der Schlupfzeitregel als Prioritätsregel durchführen.

Umsetzung:

Die ZPP ist gemäß Anforderungen implementiert. Abb. 1 zeigt die Architektur der ZPP. Der Nutzer startet die Komponente ZppSimulator, der ZppSimulator führt Simulationszyklen aus. Die Agenten Kundenauftragsersteller, Mrp2, Rückmeldungsersteller greifen lesend und schreibend auf die Datenbank Transaktionsdaten zu. Der Agent Rückmeldungsanwender greift lesend und schreibend auf die Datenbank Transaktionsdaten und auf die Datenbank Transaktionsdatenarchiv zu, was nötig ist, um abgeschlossene Transaktionsdaten zu archivieren. Nur der Agent Mrp2 liest Masterdaten. Alle drei Datenbanken sind gecached. Eine Simulation umgibt den MRP-Lauf. Ein Simulationszyklus der ZPP sieht wie folgt aus:

1. Generierung der Kundenaufträge
2. MRP II:
 - 2.1 Materialbedarfsplanung (Stücklistenauflösung und Sekundärbedarfsrechnung)
 - 2.2 Kapazitätsplanung gegen unbegrenzte Kapazität: Durchlaufterminierungen
 - 2.3 Kapazitätsplanung gegen begrenzte Kapazität: Maschinenbelegung
3. Simulation der Produktion und Rückmeldungen der Operationen (Zeiten)
4. Anwendung der Rückmeldungen

Empirische Studie zur Überprüfung der Performance der ZPP:

Abb. 2 zeigt die Laufzeit zwischen 100 und 1000 Kundenaufträgen wegen der Laufzeitentwicklung bei wachsender Kundenauftragszahl. Abb. 3 zeigt den Speicherbedarf über die Simulationszeit des vorhergenannten Falls zur Bestimmung des max. Speicherbedarfs. Abb. 4 zeigt die Laufzeit pro Simulationszyklus bei 500 Kundenaufträge wegen Komponententeilung. Die maximale Simulationsdauer ist 20160min. Ein Simulationszyklus ist 1440min lang (mind. 14 Zyklen). Den Messungen nutzen die Stücklisten und Operationen der Trucks (2 Stück) aus der SSOP wegen der Vergleichbarkeit zwischen ZPP und SSOP. Die Trucks haben eine 3-stufige Stückliste mit mind. 25 Operationen.

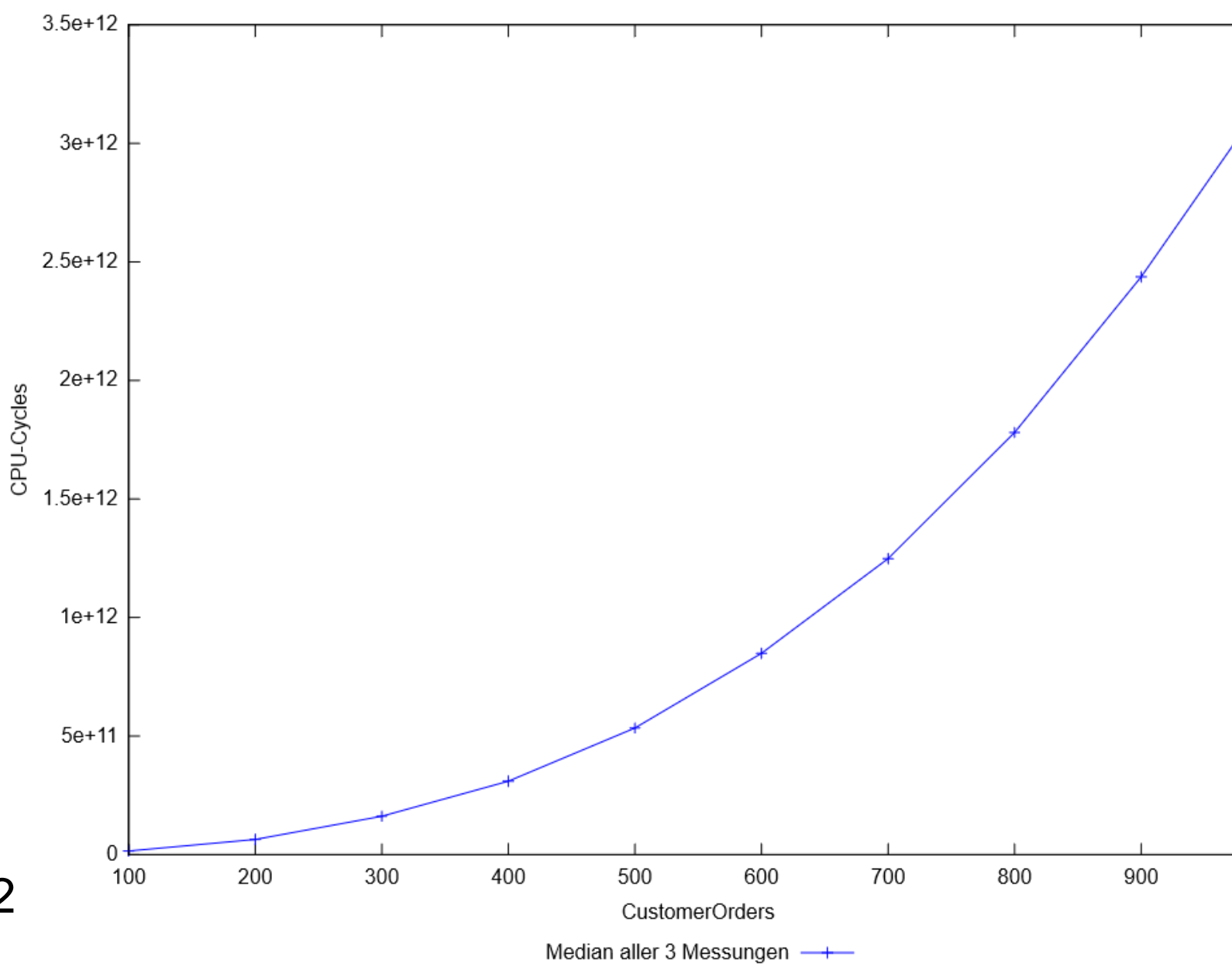


Abb. 2

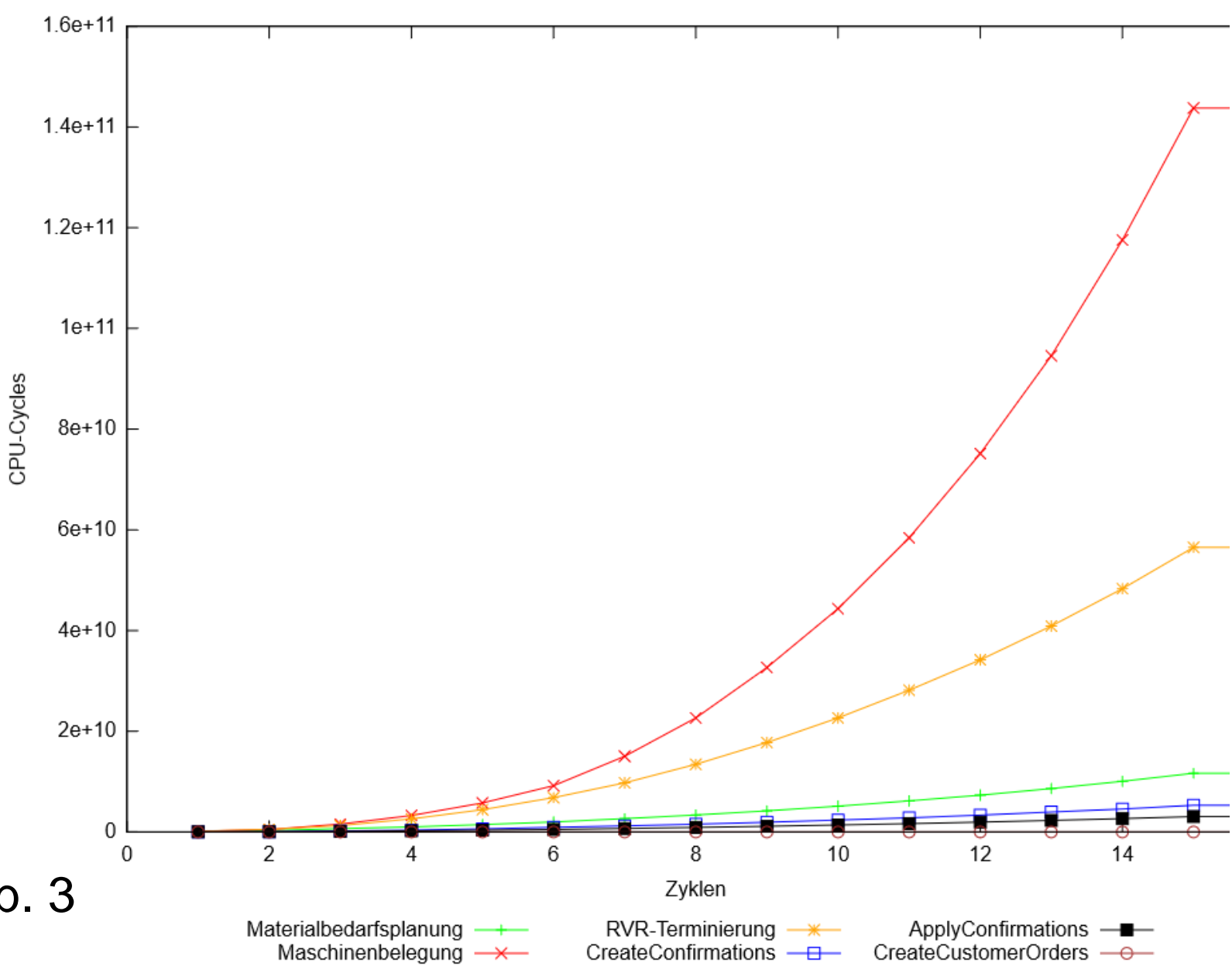


Abb. 3

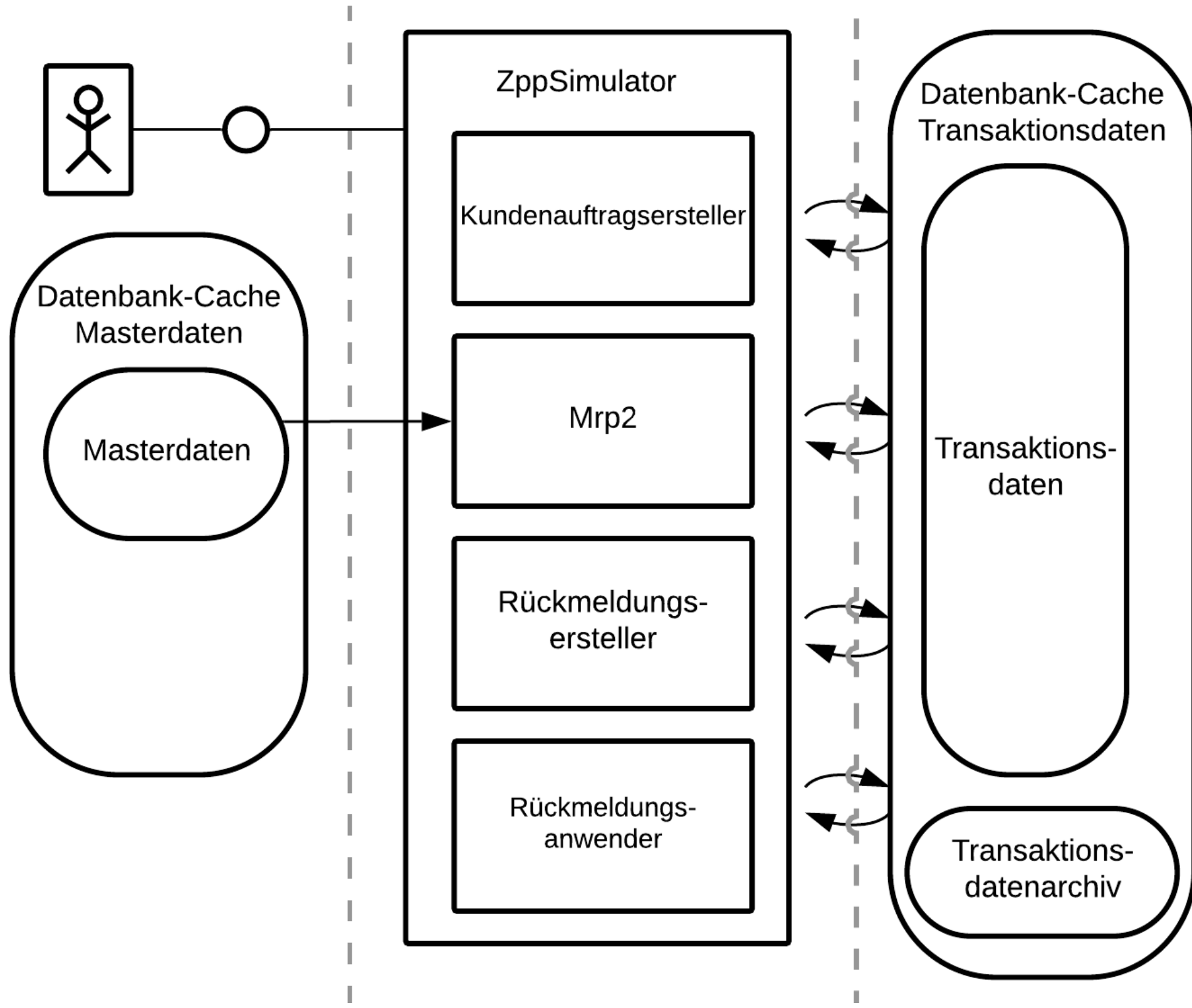


Abb. 1: Architektur der ZPP als Component/Block Diagramm (TAM)

Objektgymnastik:

Jeff Bay beschreibt neun Regeln für eine bessere Struktur des Quellcodes:

1. Verwende höchstens eine Einrückungsebene pro Methode.
2. Nutze nicht das Else-Schlüsselwort.
3. Alle Primitive und Zeichenketten kapseln.
4. Nutze nur einen Punkt pro Anweisung (statement).
5. Bezeichner nicht abkürzen.
6. Alle Entitäten kleinhalten.
7. Eine Klasse soll höchstens zwei Instanzvariablen verwalten.
8. Container (Collection) als Instanzvariable in eine eigene Klasse auslagern.
9. Nutze keine Getters, Setters und Instanzvariable außerhalb der Klasse.

Maschinenbelegung:

Die Maschinenbelegung ist ein NP-schweres Optimierungsproblem, daher verwendet man eine Heuristik. Eine Heuristik findet eine gute Lösung; sie kann aber keine Aussage darüber treffen, wie weit das Ergebnis vom Optimum entfernt ist. Diese Arbeit verwendet den Giffler-Thompson-Algorithmus. Der Giffler-Thompson-Algorithmus benötigt eine Prioritätsregel. Diese Arbeit verwendet die Schlupfzeitregel als Prioritätsregel.

Verifikation der ZPP:

Die ZPP enthält in einem Paket Integrations- und Unittests für die Bestandteile des MRP-Laufs, die Rückmeldungen der Operationen, die Anwendung der Rückmeldungen und die Struktur zur Einhaltung der Regeln der Objektgymnastik. Nach jedem Push von Commits (Git) führt ein Dienst einen Test-Job aus, der die Tests in einer virtuellen Maschine innerhalb einer gehosteten Umgebung ausführt (Cloud).

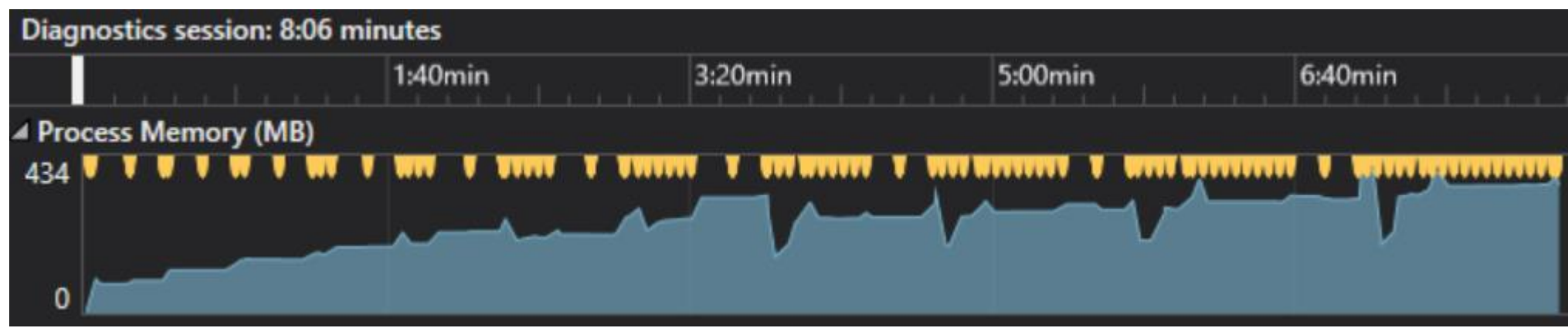


Abb. 4: Speicherbedarf über eine Simulation

Fazit

Das Ergebnis dieser Arbeit ist eine leistungsfähige ZPP mit einer Laufzeit deutlich unter 2 Stunden für eine Simulation. Die Laufzeit der ZPP liegt annähernd in $O(n)$ in einer ausgeglichenen Kapazitätssituation (< 500 Kundenaufträge). Eine umfangreiche Verifikation der ZPP weist auf eine funktionelle Korrektheit hin. Eine Validierung der ZPP (Feldversuche) ist noch ausstehend. Diese Arbeit legt die Grundlage für die Entwicklung einer hybriden Planungslösungen aus der ZPP und der SSOP.