

# Pinpoint: Problem Determination in Large, Dynamic Internet Services

Paper von M. Chen, E. Kiciman, E. Fratkin, A. Fox und E. Brewer



Thomas Kollbach  
kollbach@informatik.hu-berlin.de



# Inhalt

Problemstellung

Pinpoint-Framework

Evaluation und Ergebnisse

Zukunft und Fazit

# Problemstellung

- Diagnose von Fehlerursachen in modernen Internetdiensten
  - **Groß**  
Zusammenspiel sehr vieler Hard- und Softwarekomponenten
  - **Komplex**  
Viele unterschiedliche Komponenten (Web-Server, Load-Balancer, Datenbanken, etc.)
  - **Dynamisch**  
Redundant ausgelegt, zwei gleiche Anfragen können völlig verschiedene Komponenten beanspruchen

# Statische Analysemethoden

- Meist verwendeter Ansatz
- Basiert auf statischen Abhängigkeitsmodellen

## **Nachteile**

- Benötigt viel Wissen über das System
- Kann häufig nicht mit der Systementwicklung Schritt halten
- Einzelne redundante Komponenten können nicht identifiziert werden

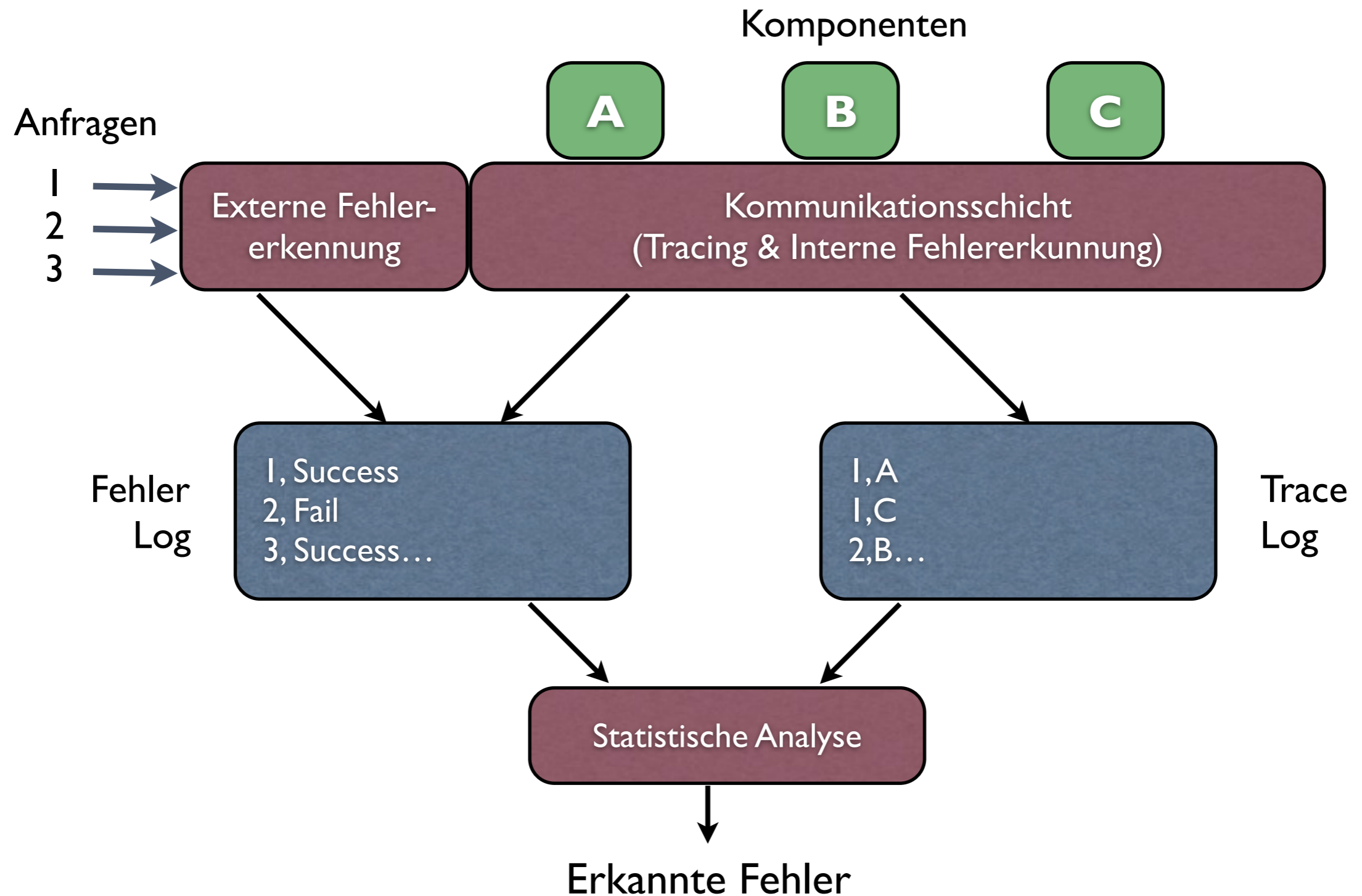
# Dynamische Methode

- Bindet sich in den Anfrageprozess ein
  - Aufzeichnung der verwendeten Komponenten
  - Fehler oder Erfolg wird aufgezeichnet
  - Statistische Analyse zum Finden der Fehlerquelle

## **Vorteile**

- Passt sich dem System bei Veränderungen an
- Kann spezifische Komponenten identifizieren
- Vollständig automatisiert
- Black-Box Ansatz

# Pinpoint Framework

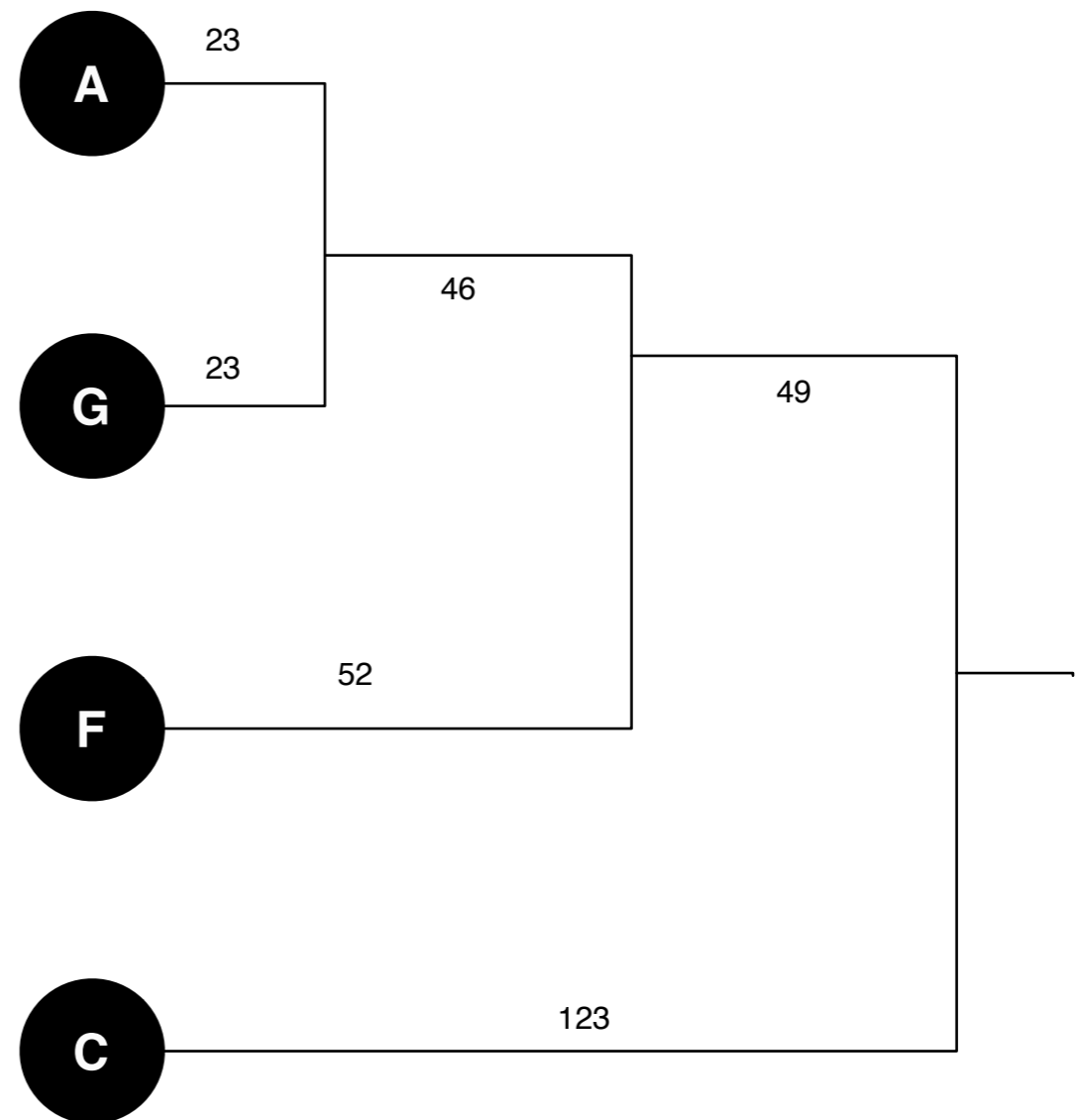


# Pinpoint (2)

- Prototyp
- Diagnostiziert PetStore J2EE Web-Dienst
- Interne und externe Fehlererkennung
  - Layer-7 Sniffer (z.B. HTTP-Fehler oder TCP-Timeouts)
  - Zusätzlich Einbau in J2EE-Anwendung

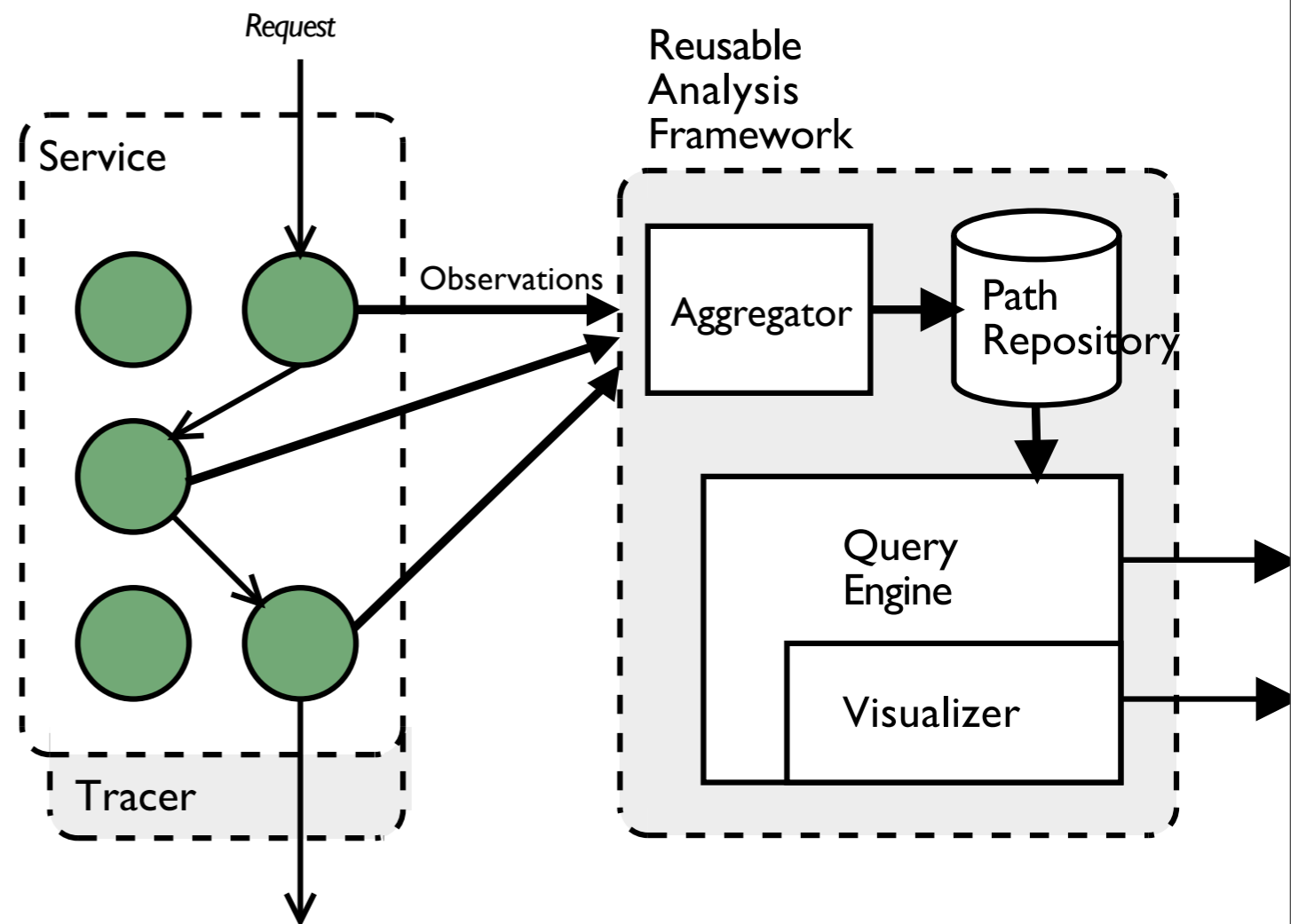
# Data Clustering

- Unweighted Pair-Group Method using Arithmetic Averages (UPGMA)
- Ähnlichkeitsmetrik Jaccard-Koeffizient als (Abstand im UPGMA)

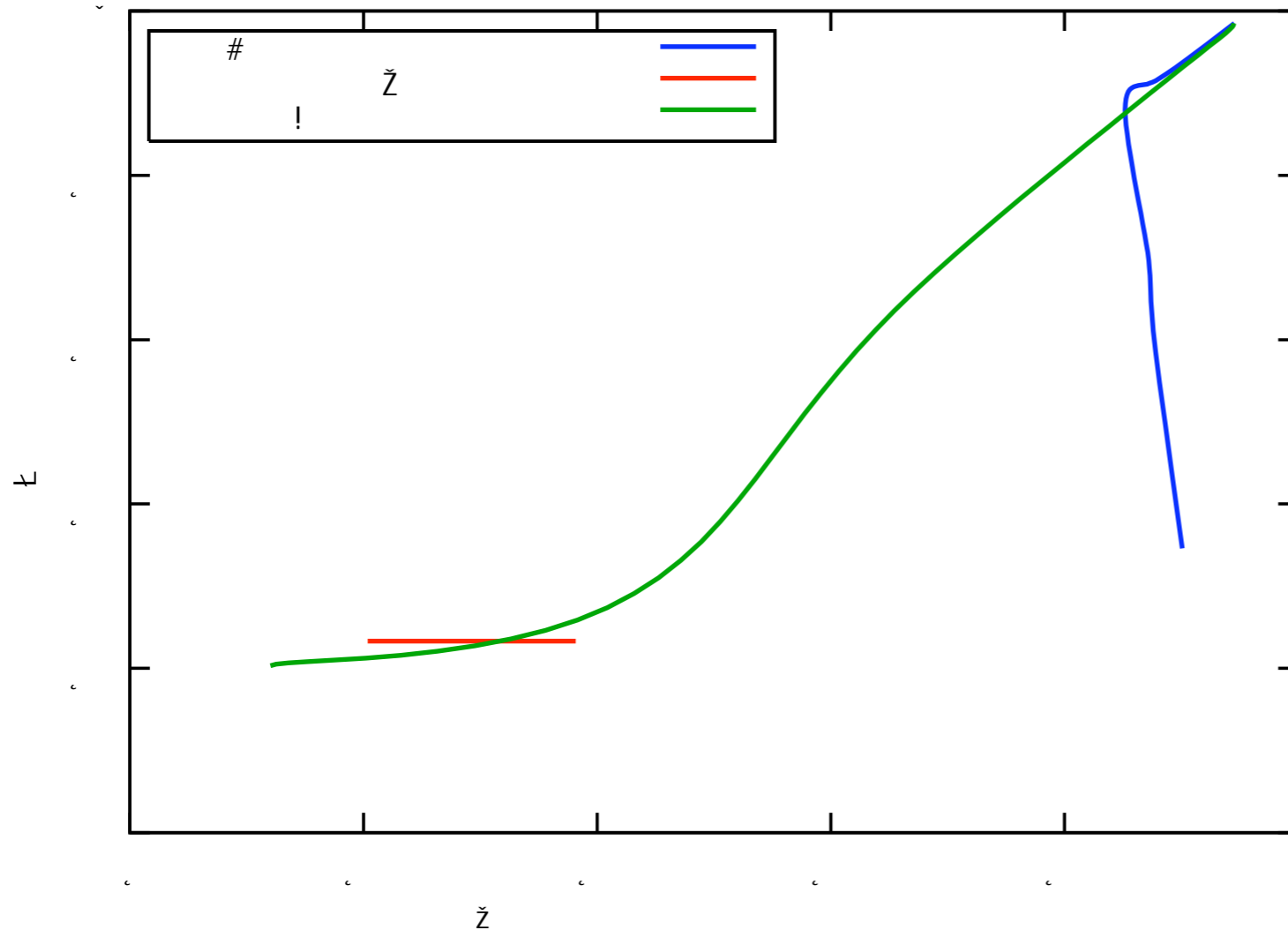


# Evaluation

- 133 Test-Sessions mit zufälligen, künstlich induzierten Fehlertypen
- Browser Emulator emuliert surfende Benutzer
- Fehlertypen
  - Erwartete Fehler  
IOException, RuntimeException
  - Unerwartete Fehler  
Endlosschleifen, Null-Aufrufe



# Ergebnisse



**Summary accuracy vs. false positive rate over all tests**

- *Accuracy*  
Alle Komponenten die den Fehler verursachen wurden korrekt identifiziert
- *Precision*  
Verhältnis zwischen vorhergesagten fehlerhaften Komponenten und tatsächlich fehlerhaften Komponenten

# Ergebnisse (2)

- Probleme
  - Eng gekoppelte Komponenten können nicht unterschieden werden
  - Basiert auf der Annahme, dass Anfragen unabhängig sind
  - Keine Erkennung von abnehmender Performance

# Fazit & Zukunft

- Deutliche Verbesserung gegenüber herkömmlichen Fehlererkennungssystemen
- Black Box Ansatz
  - Theoretisch gut → Einfache Integration
  - Bei Pinpoint nicht ganz erfüllt
- Laufzeitanalyse für verteilte Systeme wird aktiv weiterentwickelt

# Quellen

M. Chen, E. Kiciman, E. Fratkin, E. Brewer and A. Fox. *Pinpoint: Problem Determination in Large, Dynamic Internet Services*. In Symposium on Dependable Networks and Systems (IPDS Track). 2002.

M. Chen, E. Kiciman, A. Accardi, A. Fox, E. Brewer. *Using Runtime Paths for Macroanalysis*. In Proceedings of the 9th conference on Hot Topics in Operating Systems. S. 14ff.

T. Fawcett. *Roc graphs: Notes and practical considerations for data mining researchers*. Technical report hpl-2003-4, HP Laboratories, Palo Alto, CA, USA, January 2003.

# Vielen Dank für die Aufmerksamkeit!

Diese Präsentation steht unter einer  
Creative Commons Attribution-NonCommercial-ShareAlike 2.0 Germany License  
<http://creativecommons.org/licenses/by-nc-sa/2.0/de/>

Folien verfügbar unter  
<http://thomas.kollba.ch/files/folien/pinpoint.pdf>