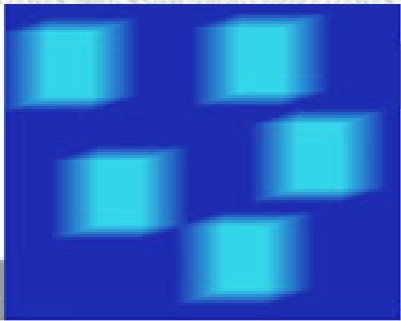


RSerPool – Reliable Server Pooling



Technik der Rechnernetze

Prof. Dr.-Ing. Erwin P. Rathgeb

Dipl.-Inf. Thomas Dreibholz

Alfried Krupp von Bohlen und Halbach-Stiftungsprofessur

Technik der Rechnernetze

Institut für Experimentelle Mathematik und

Institut für Informatik und Wirtschaftsinformatik

Universität Duisburg-Essen (Campus Essen)

■ Motivation

- Wie werden Dienste im Netz unterstützt?

■ Einführung

- Was ist Reliable Server Pooling?
- Stand der IETF-Standardisierung
- Anwendungsszenarien

■ Leistungsuntersuchungen und Optimierung

- Handlespace Management
- Pool Policies und Lastverteilung

■ Zusammenfassung und Ausblick

- Unsere Forschungsaktivitäten und Ergebnisse
- Unsere Standardisierungsaktivitäten

Dienste in Telekommunikationsnetzen – Beispiel ATM

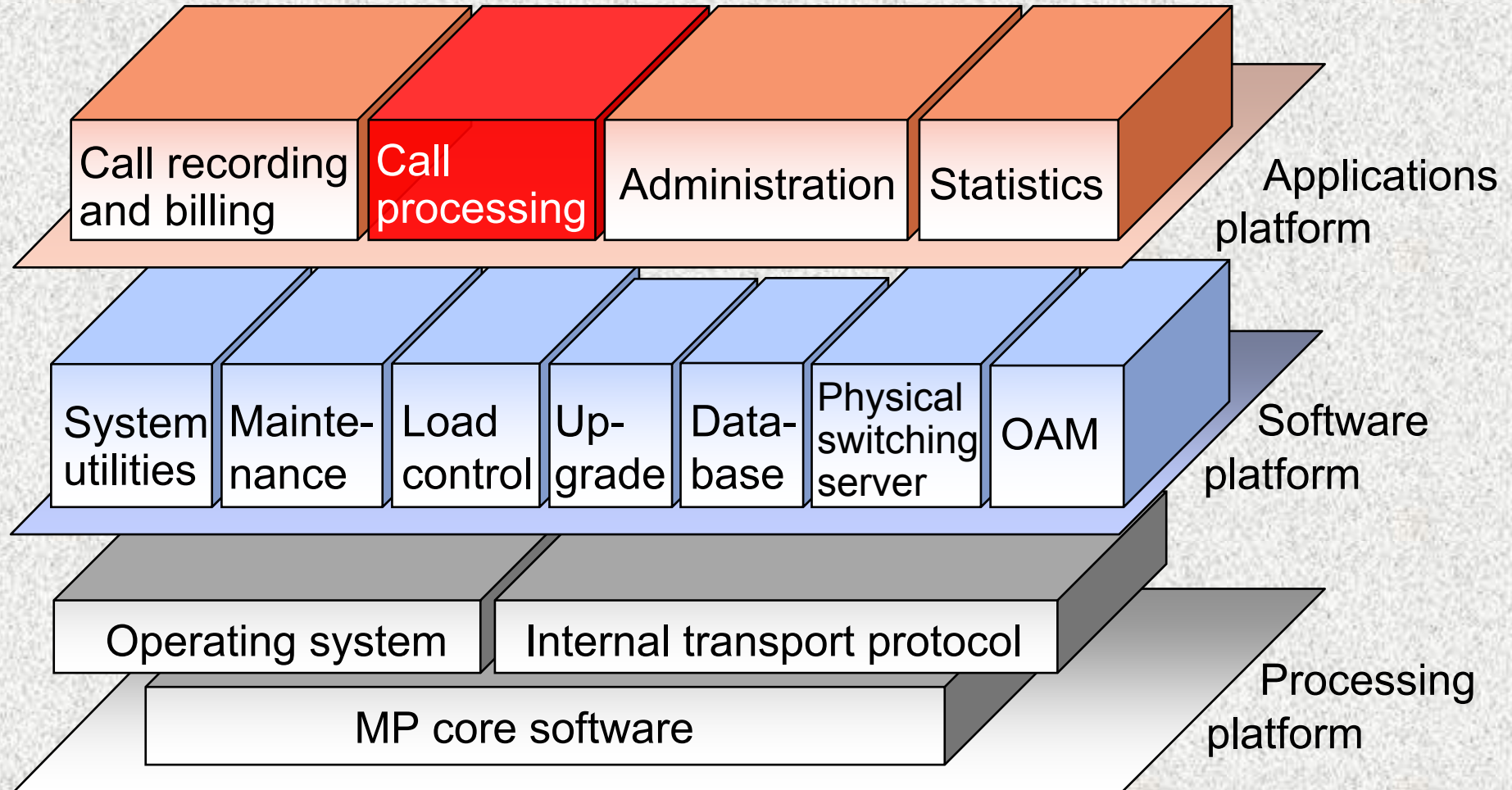


- Durchgängige Standardisierung
 - Teledienste
 - Dienstmerkmale
- Dienstverfügbarkeit durch hoch zuverlässige Netzkomponenten
 - Alle zentralen Teile voll redundant
 - Fehlertolerante Softwarearchitektur
 - Upgrade ohne Betriebsunterbrechung
- Integration der Dienstsoftware in die leistungsfähige Knotensteuerung
 - Verteilte Multiprozessorplattform, über 100 Prozessoren maximal
 - Skalierbar bis über 2,5 MBHCA



MBHCA: Million Busy Hour Call Attempts
ASIC: Application Specific Integrated Circuit

Dienste in Telekommunikationsnetzen – Integration in die Steuersoftware



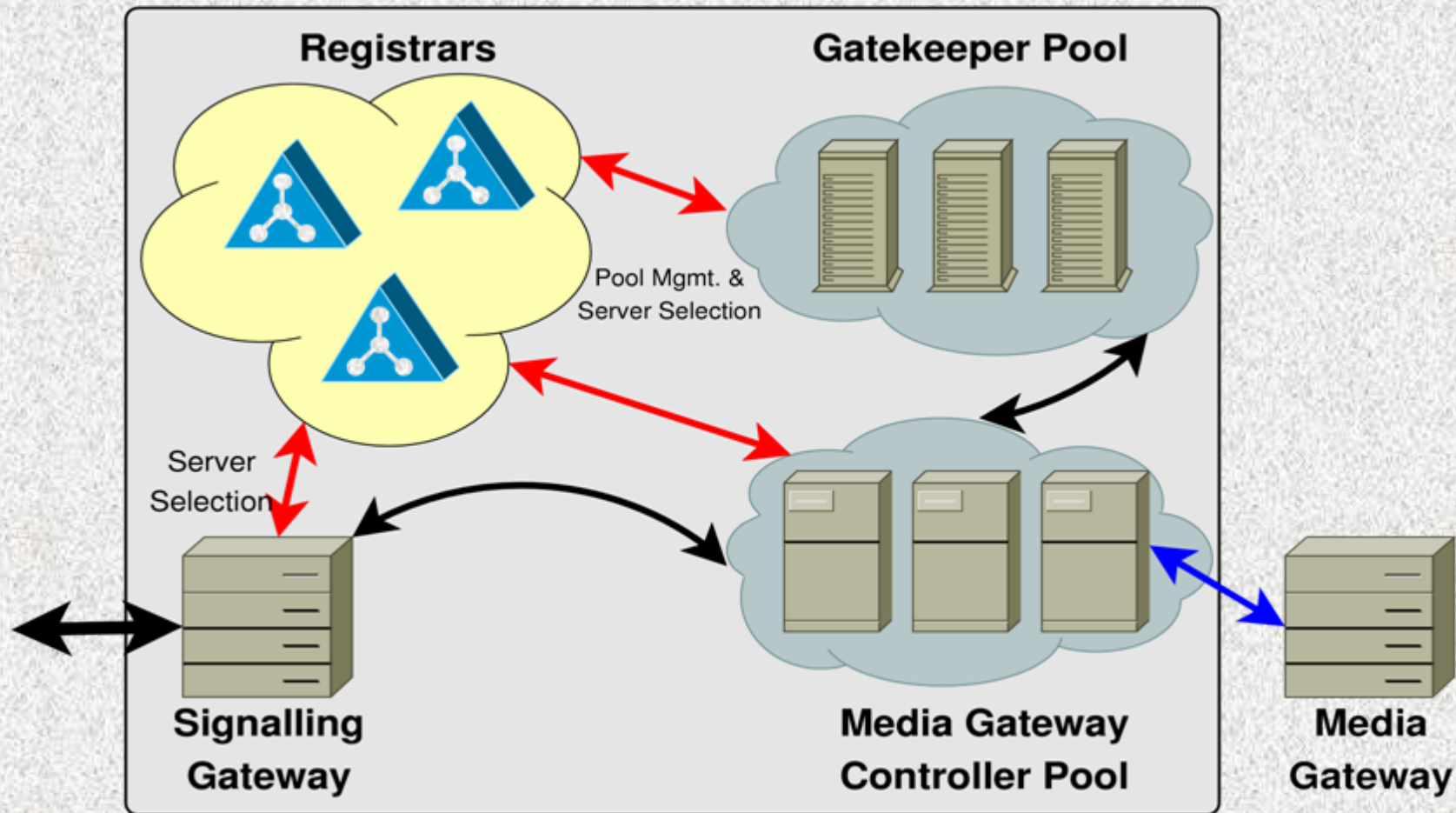
- Nur „Standardisierung“ von Basisprotokollen
 - Flexible Einführung von neuen Anwendungen ohne Netzunterstützung
 - Peer-to-Peer, z.B. Skype
 - Open Source, z.B. Asterisk
 - Evtl. Einschränkungen bei der Interoperabilität
- „Dienstverfügbarkeit“ durch Kooperation verteilter Komponenten
 - Routing, ...
 - Evtl. Einschränkungen bei der Echtzeitfähigkeit
- Begrenzte Prozessorkapazität der Knotensteuerung
 - Integration von Dienstfunktionalität nur begrenzt möglich

Reliable Server Pooling (RSerPool) – Motivation und Eigenschaften

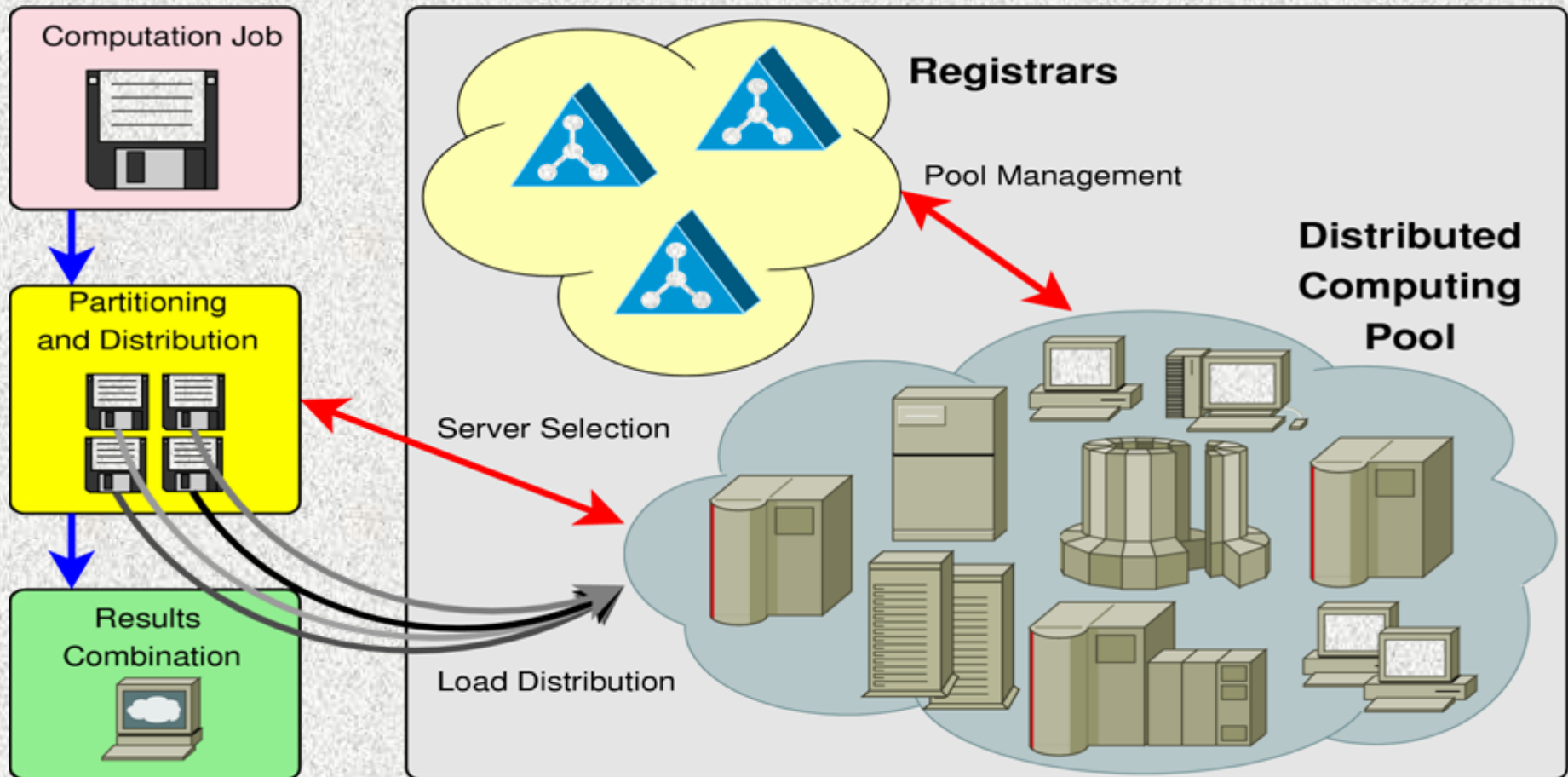


- **RFC 3237 Requirements for Reliable Server Pooling**
 - 12 weitere Dokumente der RSerPool WG im Draft-Status
- **Primäre Motivation: Signalling Transport (SS7)**
 - Ansatz: Verfügbarkeit durch redundante Komponenten und nicht durch Redundanz in den Komponenten
- **Eigenschaften von Reliable Server Pooling (IETF RSerPool WG)**
 - Lightweight Auch in ressourcenarmen Geräten einsetzbar
 - Realtime Schnelle Reaktion im Fehlerfall
 - Scalable
 - Extendable
 - Simple Autokonfiguration
- **Weitere Anwendungsszenarien**
 - Lastverteilung für Webserver (diskutiert beim 63. IETF-Meeting)
 - SCTP Mobility Management [LCN2003]

Anwendungsszenarien für RSerPool – SigTran-Beispiel [draft-ietf-rserpool-arch-12.txt]



Anwendungsszenarien für RSerPool – Verteiltes Rechnen [draft-dreibholz-rserpool-distcomp-01.txt]



Reliable Server Pooling (RSerPool) – Komponenten



■ Pool Element (PE)

- Server, der einen oder mehrere Dienste zur Verfügung stellt

■ Pool

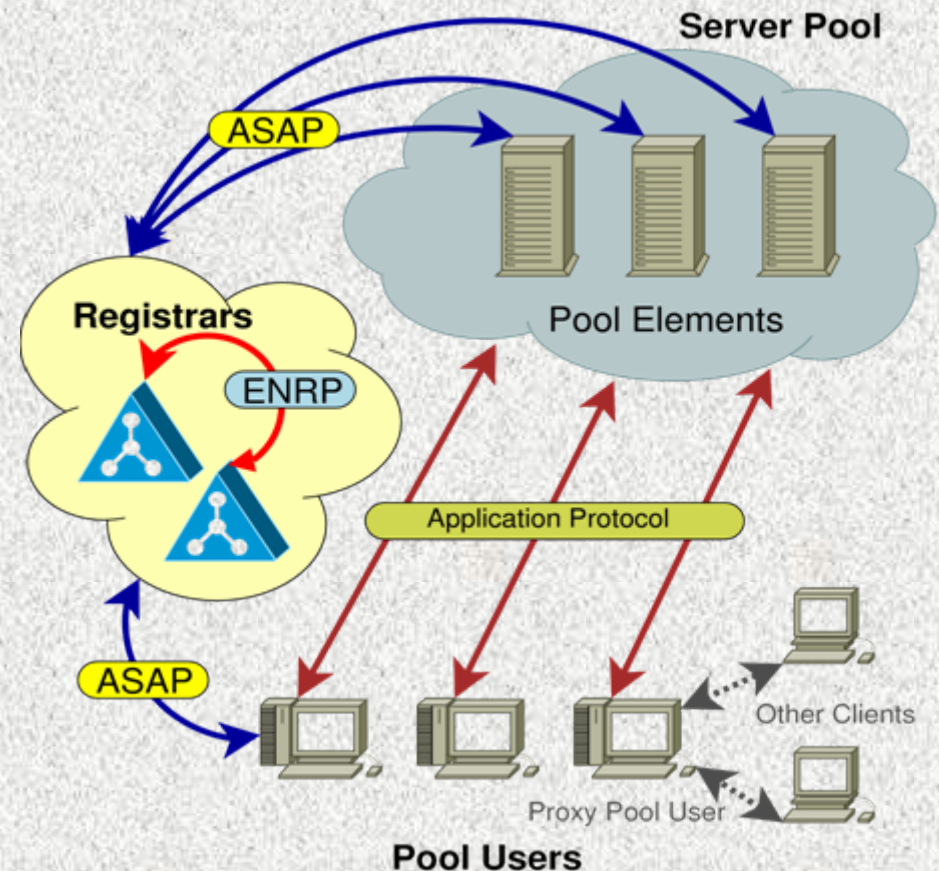
- Menge von PEs, die einen bestimmten Dienst zur Verfügung stellen
- Ein Pool wird durch eine eindeutige Kennung adressiert (Pool Handle)
- Die Menge der verfügbaren Pools wird als “Handlespace” bezeichnet

■ Registrar (PR)

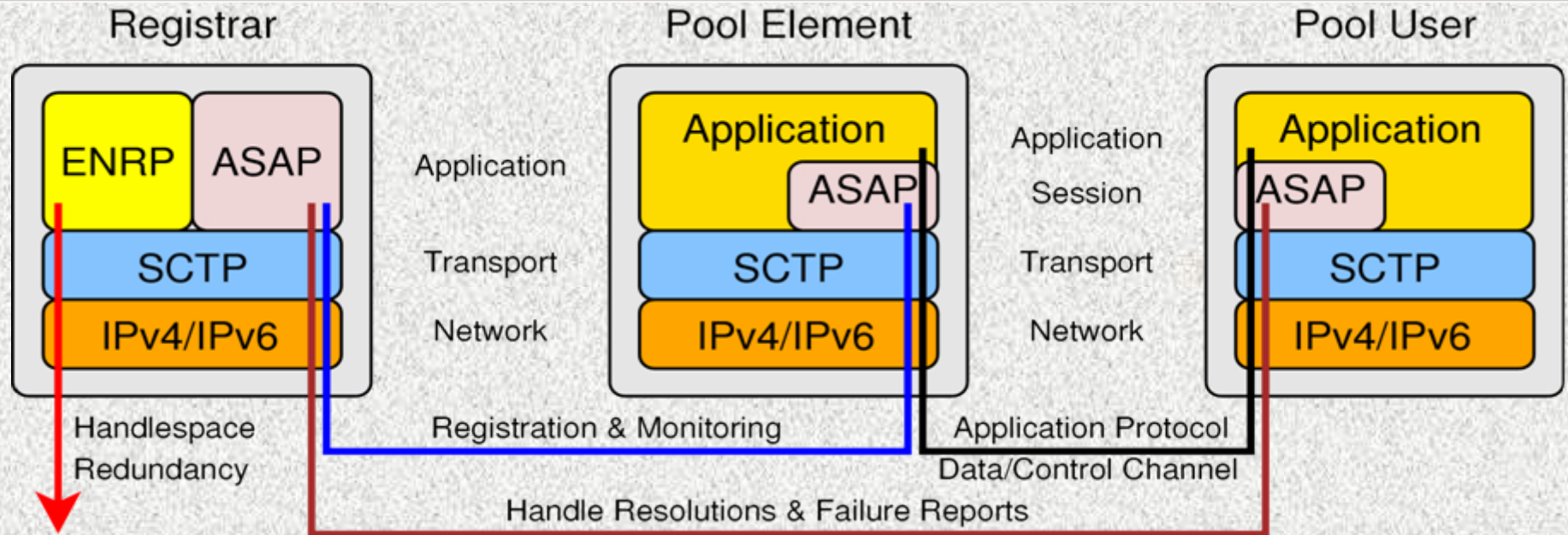
- Registriert und überwacht die PEs
- Gibt auf Anfrage die Identitäten geeigneter PEs an die Nutzer weiter

■ Pool User (PU):

- Nutzer eines Pools



Reliable Server Pooling (RSerPool) – Protokolle



■ Aggregate Server Access Protocol (ASAP)

- PR $\Leftrightarrow$ PE: Registrierung und Überwachung
- PR $\Leftrightarrow$ PU: Auflösung von Dienstanfragen, Fehlerberichte

■ Endpoint Handlespace Redundancy Protocol (ENRP)

- PR $\Leftrightarrow$ PR: Synchronisation der Zustandsinformation

Reliable Server Pooling (RSerPool) – Pool Member Selection Policies



■ Was ist eine Pool Policy?

- Eine Regel nach der die Auswahl der PEs erfolgt
- Definiert in unserer WG Draft [draft-ietf-rserpool-policies-01.txt]

■ Nicht-adaptive Policies

- Round Robin (RR) / Weighted Round Robin (WRR)
- Random (RAND) / Weighted Random (WRAND)

■ Adaptive Policies

- Least Used (LU)
- Lastdefinition ist anwendungsspezifisch, z.B.
 - CPU-Last, Nutzeranzahl, Verzögerung, ...

■ Anwendung

- Registrar: Erstellt auf Anfrage der PUs eine PE-Liste
- Pool User: Selektiert ein PE aus der Liste
- Beides erfolgt anhand der Policies (Pool-spezifisch)

Reliable Server Pooling (RSerPool) – Handlespace Management



■ Handlespace

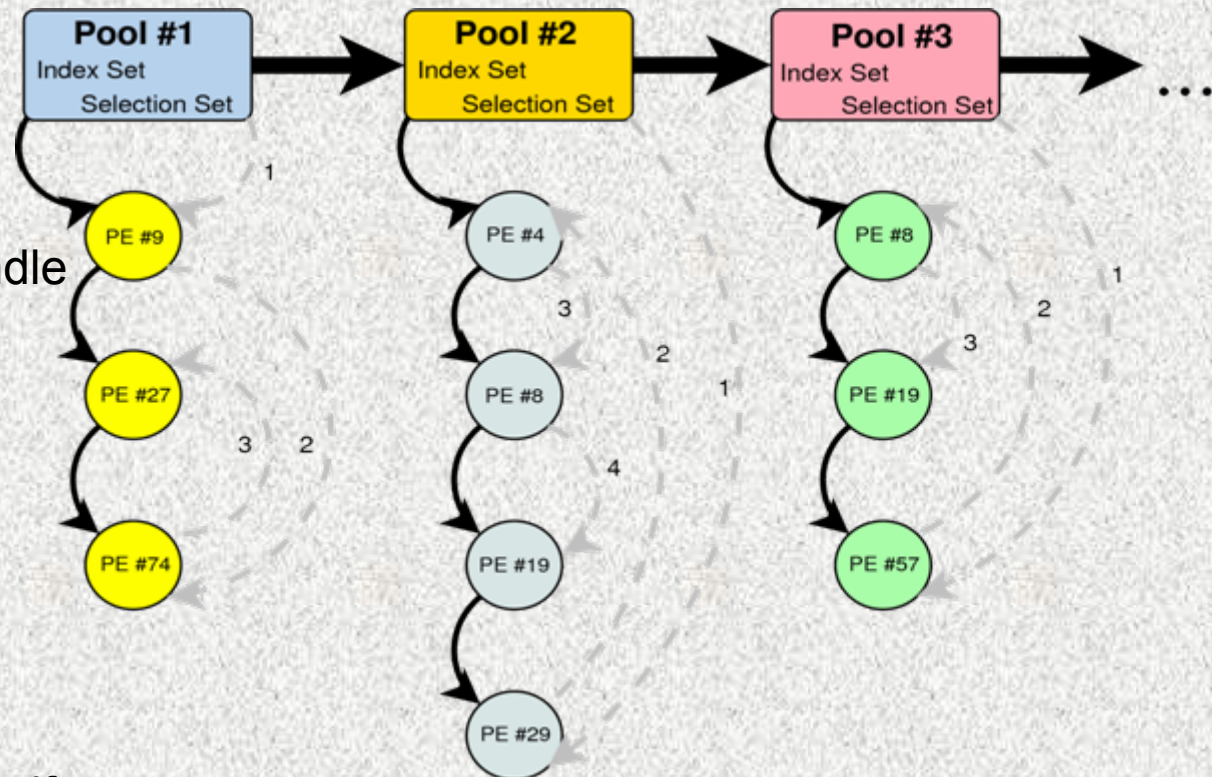
- Menge von Pools
 - Sortiert nach Pool Handle

■ Pool

- Menge von PE Indizes
 - Sortiert nach PE ID
- PE Auswahlmenge
 - Nach Sortiervorschrift
- Auswahlverfahren

■ Implementierung der Policies

- Definition der Sortiervorschrift
- Definition des Auswahlverfahrens



Reliable Server Pooling (RSerPool) – Handlespace Management Performance

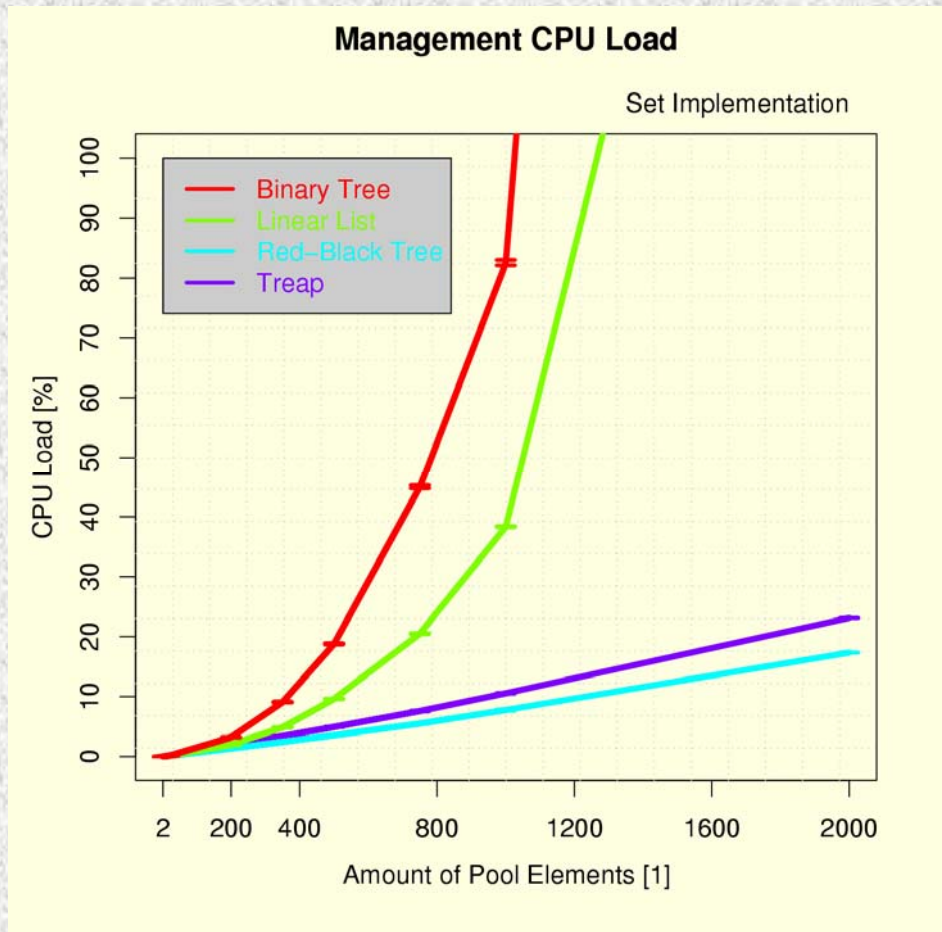


■ Parameter

- Anzahl der PEs variiert
- 2 Pools
- 10 Selektionen per PE und Sek.
- PE Re-Registrierung: Alle 30s
- PE Uptime: 30min (Mittelwert)
- Synchronisation: Alle 5min
- Athlon 1.3GHz (Router CPU)

■ Ergebnisse

- Unbalancierte Bäume untauglich
- Für balancierte Bäume
 - CPU Last < 20% für 2000 PEs



Reliable Server Pooling (RSerPool) – Simulationsstudien

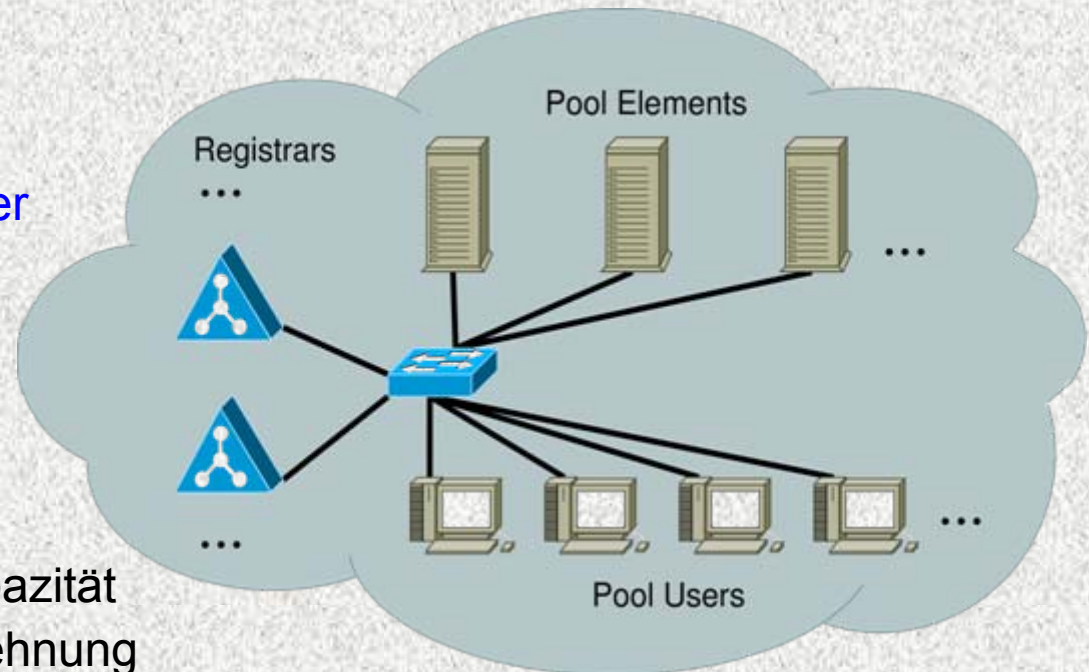


■ Simulationswerkzeug

- OMNeT++
- R Project zur Auswertung

■ Simulations-Szenario und Parameter

- m PRs
- n PUs
 - Anforderungsrate
 - Jobgröße
- k PEs
 - Homogene/heterogene Kapazität
 - Multi-Tasking mit/ohne Ablehnung von Anforderungen
- Verzögerungen:
 - Linkverzögerung



Dimensionierung:
Zielauslastung bei optimaler Verteilung ist
80% (60%) der Gesamtkapazität

Reliable Server Pooling (RSerPool) – Simulationsstudien: Kenngrößen

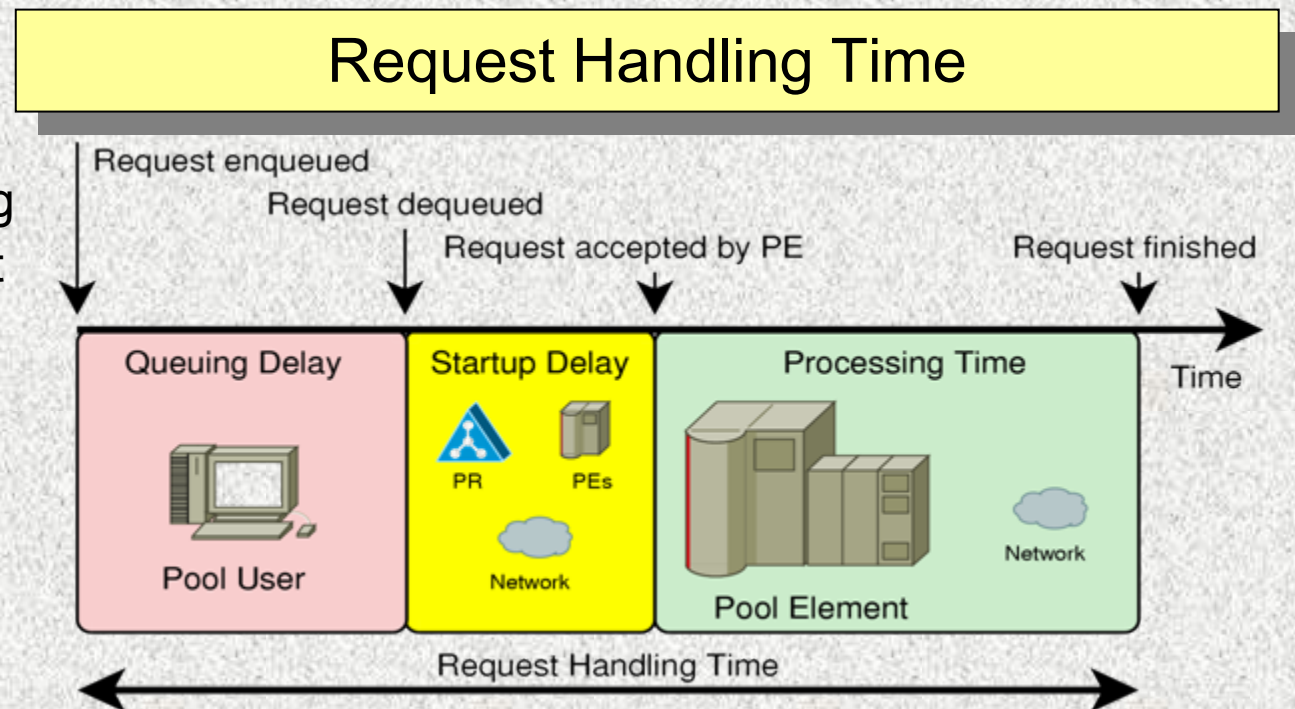


■ Aus Providersicht

- Mittlere Auslastung der Serverressourcen

■ Aus Anwendersicht

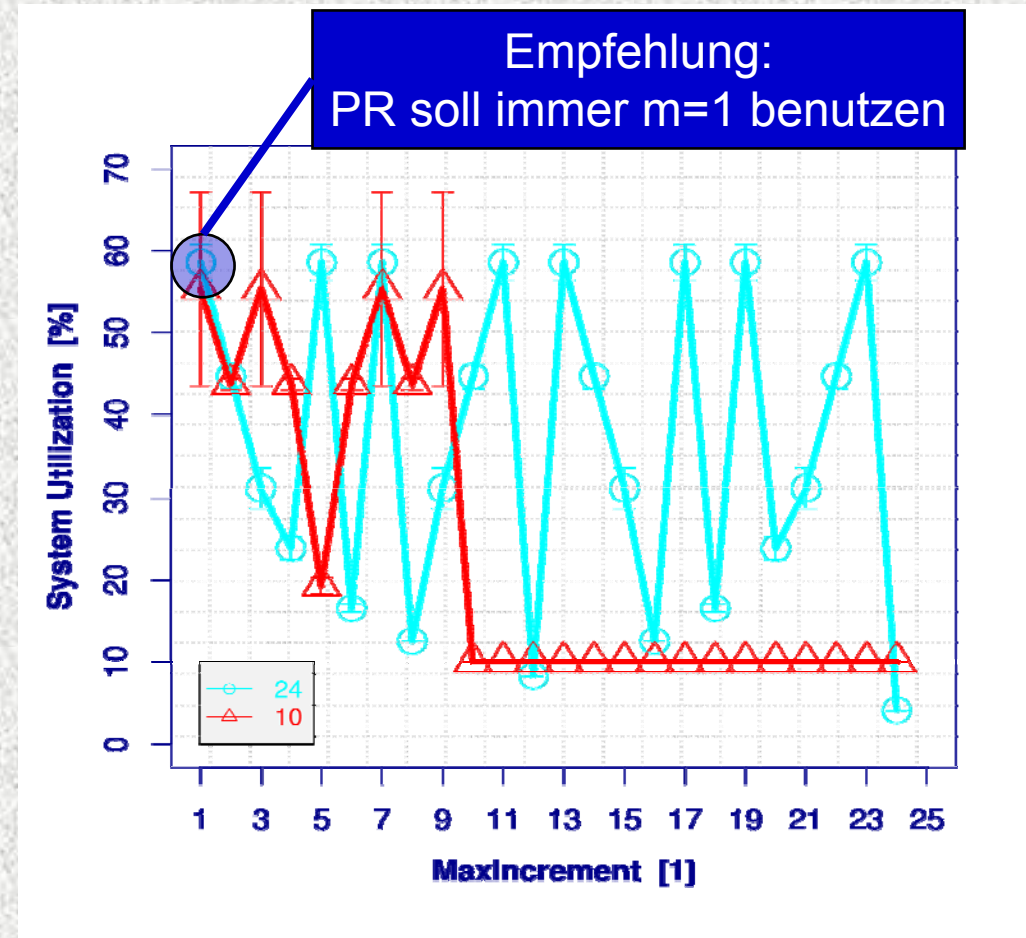
- Mittlere Antwortzeit
- Besteht aus
 - Wartezeit
 - Startverzögerung
 - Bearbeitungszeit



RSerPool-Simulationsstudien – Problematik der Round Robin Policy



- Anzahl der PEs im Pool n
 - Ändert sich dynamisch
- (Max.) Anzahl der PEs pro Liste MaxHResItems
 - Annahme: Die PE-Liste enthält alle PEs
 - $\text{MaxHResItems} \geq n$
- $\text{MaxIncrement } m$
 - Versatz der Startwerte für aufeinander folgende Listen
- Problematischer Fall
 - PUs wählen bei RR immer zuerst das erste PE aus der Liste
 - Anzahl der PEs n ist ein Vielfaches von $\text{MaxIncrement } m$
 - PE #1,
 - PE #(1+m),
 - PE #(1+2*m), ...
 - Worst Case:
 - $\text{MaxIncrement } m == \text{Poolgröße } n$
 - Es wird immer PE #1 gewählt

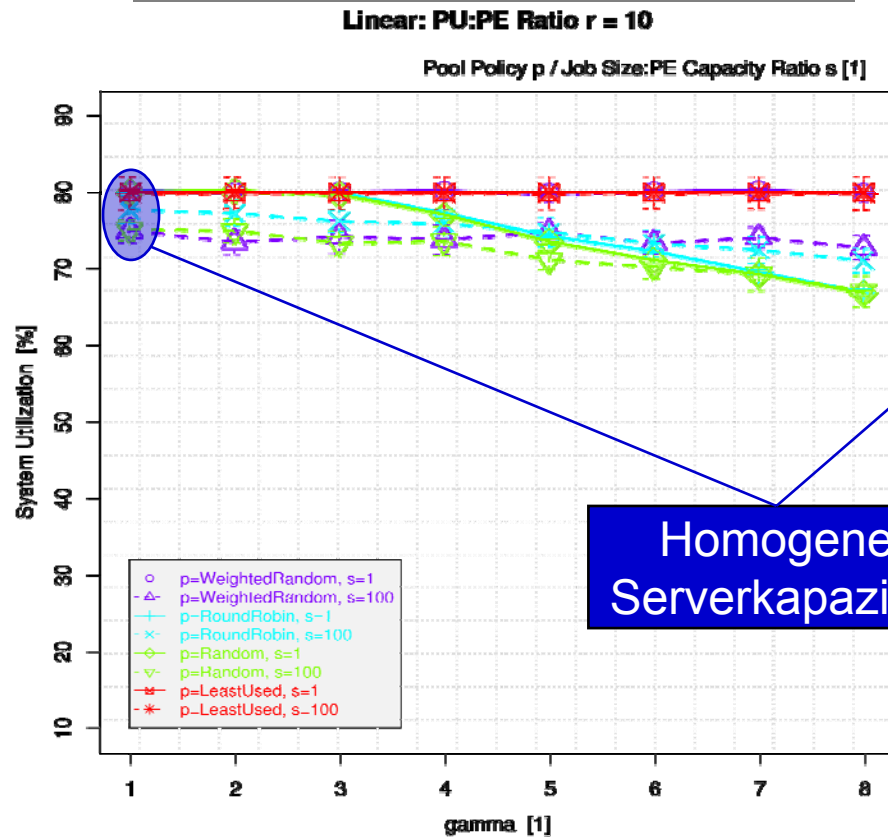


RSerPool-Simulationsstudien – Einfluss heterogener Serverkapazität

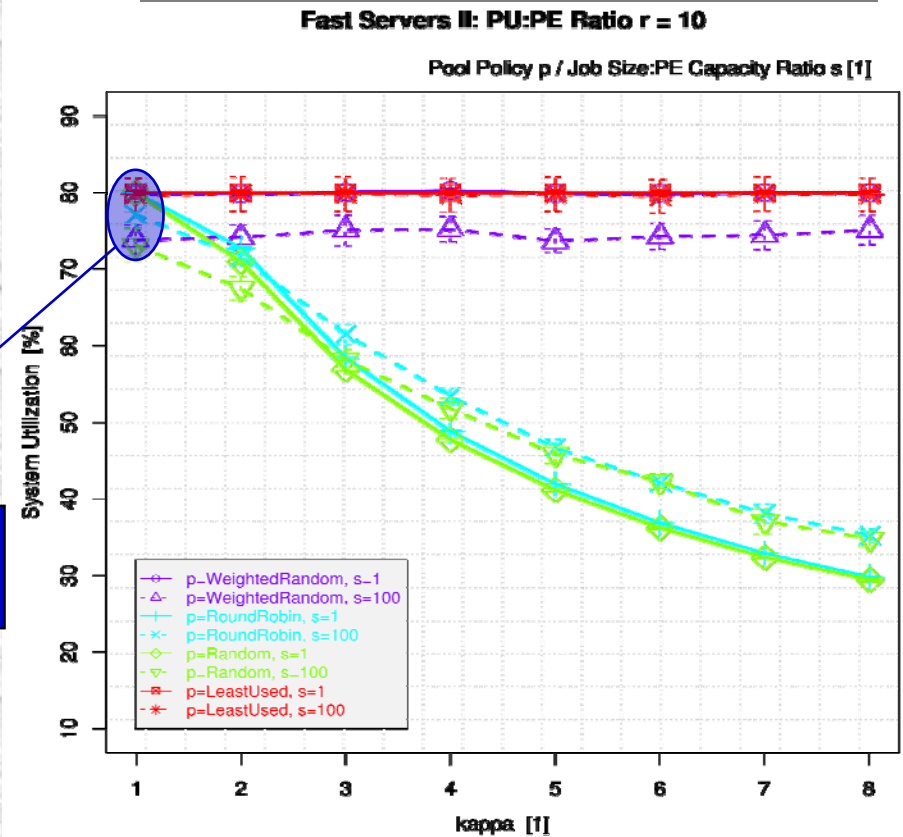


Linear skalierte Serverkapazität

1/3 schnelle Server



Homogene
Serverkapazität

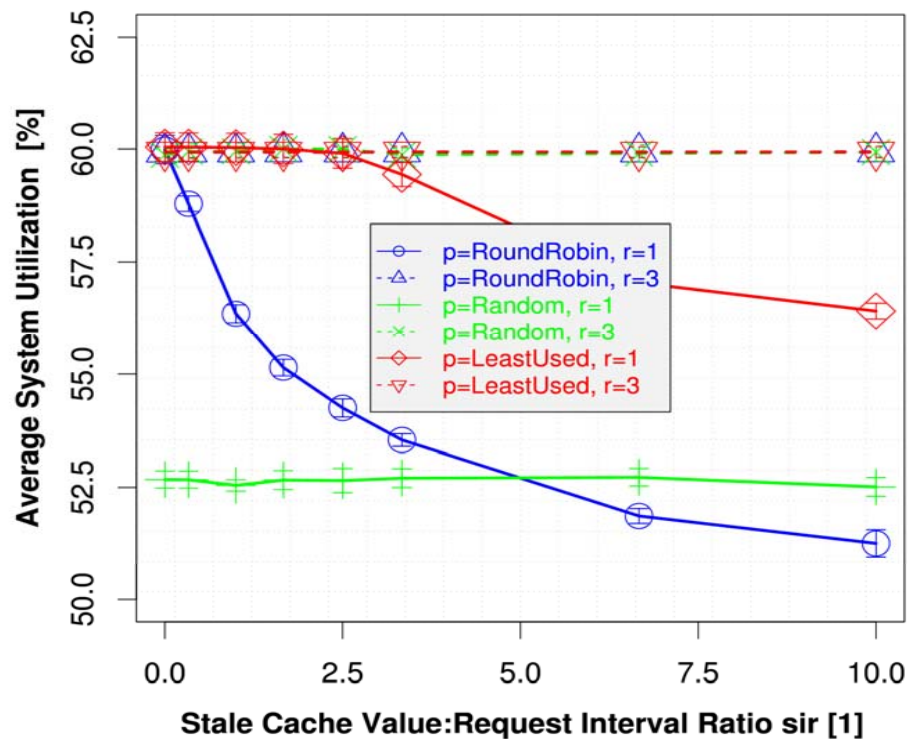


RSerPool-Simulationsstudien – Einfluss des PU-Cache für PE-Adressen



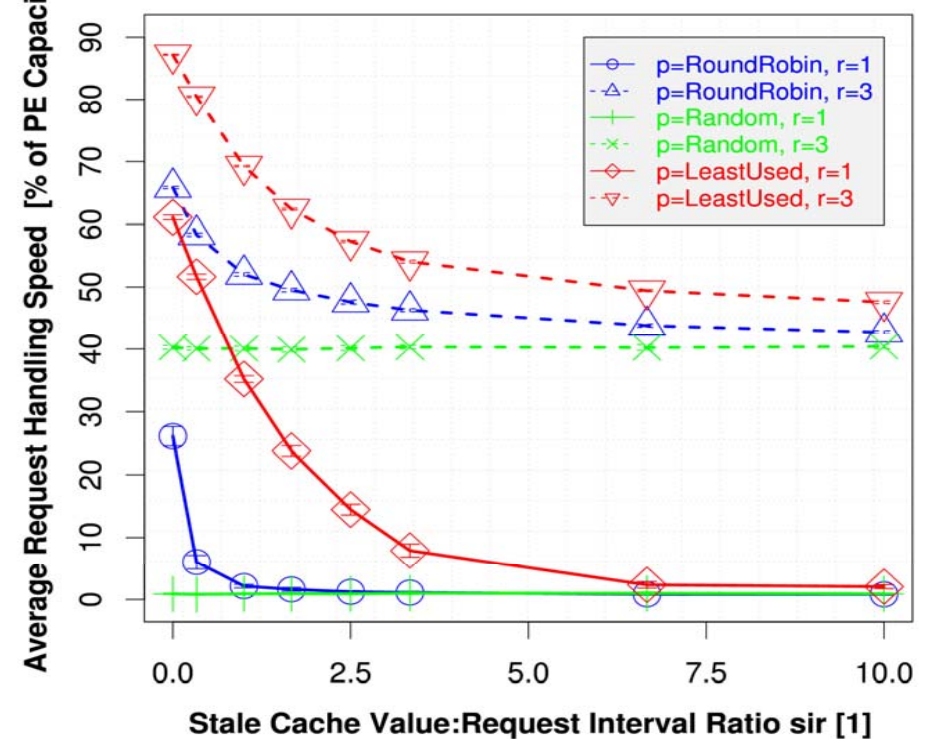
System Utilization

Pool Policy p / PU:PE Ratio r [1]



Request Handling Speed

Pool Policy p / PU:PE Ratio r [1]



Nutzung des PU-Cache für die PE-Identitäten ist kritisch und sollte vermieden werden

RSerPool-Simulationsstudien – Nutzen des PU-Cache für PE-Adressen



■ Bedingungen für Nutzen des PU-Cache

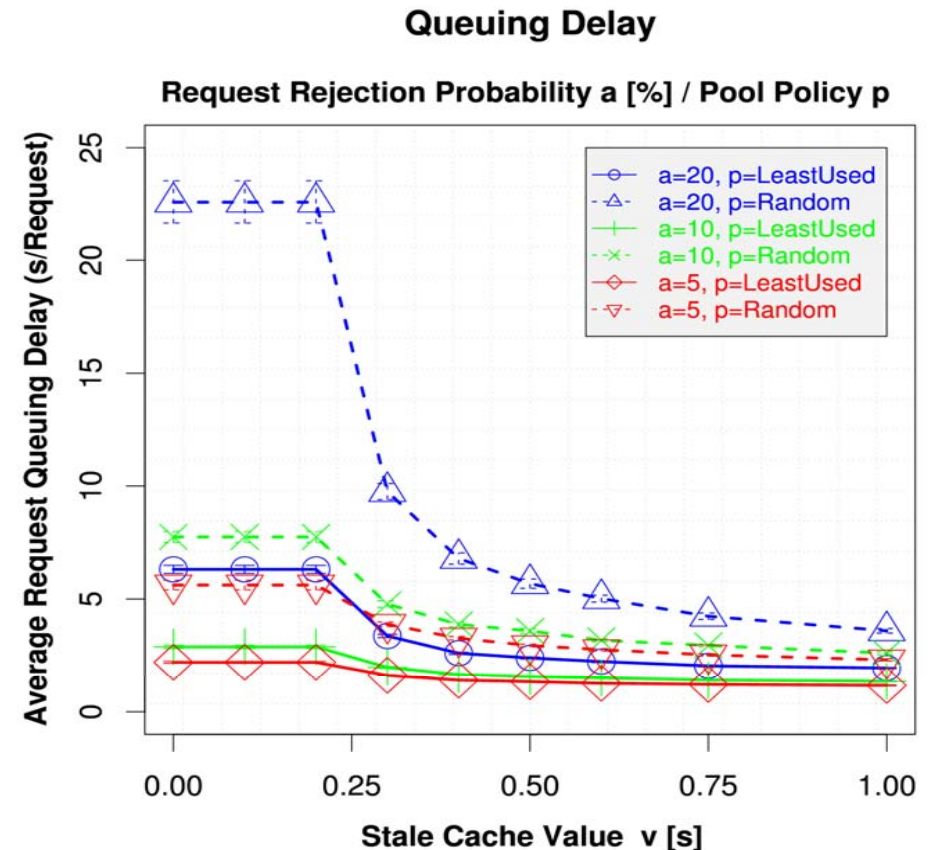
- Netzverögerung $\approx$ Bearbeitungszeit
 - PR-Anfragen werden reduziert
- PE lehnt Anfragen bei Überlast ab
 - Mehrere Versuche erforderlich

■ Beispiel

- Ablehnungswahrscheinlichkeit
 - $a = \{5\%, 10\%, 20\%\}$
- Stale Cache Value
 - Sollte genügend Wiederholungen erlauben (95% Erfolgschance)

■ Ergebnis

- Signifikante Reduktion der Wartezeit



Reliable Server Pooling (RSerPool) – Unsere Forschungsaktivitäten



- Begonnen als Kooperation mit Siemens, seit Oktober 2004 als DFG-Projekt
- Die Prototypimplementierung rsplib
 - Abgeschlossene Untersuchungen:
 - Unterstützung für SCTP-basierte Mobilität [LCN2003]
 - Realisierung eines Distributed Computing Szenarios [LCN2002] [LCA2003]
 - Laufende Arbeiten:
 - Validierung der Ergebnisse aus den Simulationsstudien
 - Untersuchungen im PLANET LAB
- Das Simulationsmodell rspsim
 - Abgeschlossene Untersuchungen:
 - Pool Policy Performance [LCN2005] [Tencon2005] [ICN2005]
 - Failover Handling [EuroMicro2005] [LCN2002]
 - Handlespace Management [Contel2005]
 - Laufende Arbeiten:
 - Pool Policy Performance in unterschiedlichen Ausfallszenarien
 - Untersuchung der Funktionalität und Skalierbarkeit von ENRP

Reliable Server Pooling (RSerPool) – Unsere Standardisierungsaktivitäten



■ Beteiligung an 3 Working Group Drafts

- draft-ietf-rserpool-policies-01.txt
 - Ergebnisse der Untersuchungen zu den Pool Policies
- draft-ietf-rserpool-mib-01.txt
 - SNMP MIB-Beschreibung der RSerPool-Komponenten
- draft-ietf-rserpool-api-00.txt
 - RSerPool PU/PE API



■ Mehrere individuelle Beiträge

■ Der rsplib Prototyp ist die weltweit erste RSerPool-Implementierung

- Open Source (GPL License)
- Referenzimplementierung der IETF RSerPool WG
- Erste erfolgreiche Interoperabilitätstest
 - Proprietäre Motorola-Implementierung (beim 60. IETF Meeting)

Reliable Server Pooling (RSerPool) – Weitere Informationen



Project Homepage:

<http://tdrwww.exp-math.uni-essen.de/dreibholz/rserpool/>

Thomas Dreibholz, dreibh@exp-math.uni-essen.de