UWS UNIVERSITY OF THE WEST of SCOTLAND

HOCHSCHULE DER MEDIEN

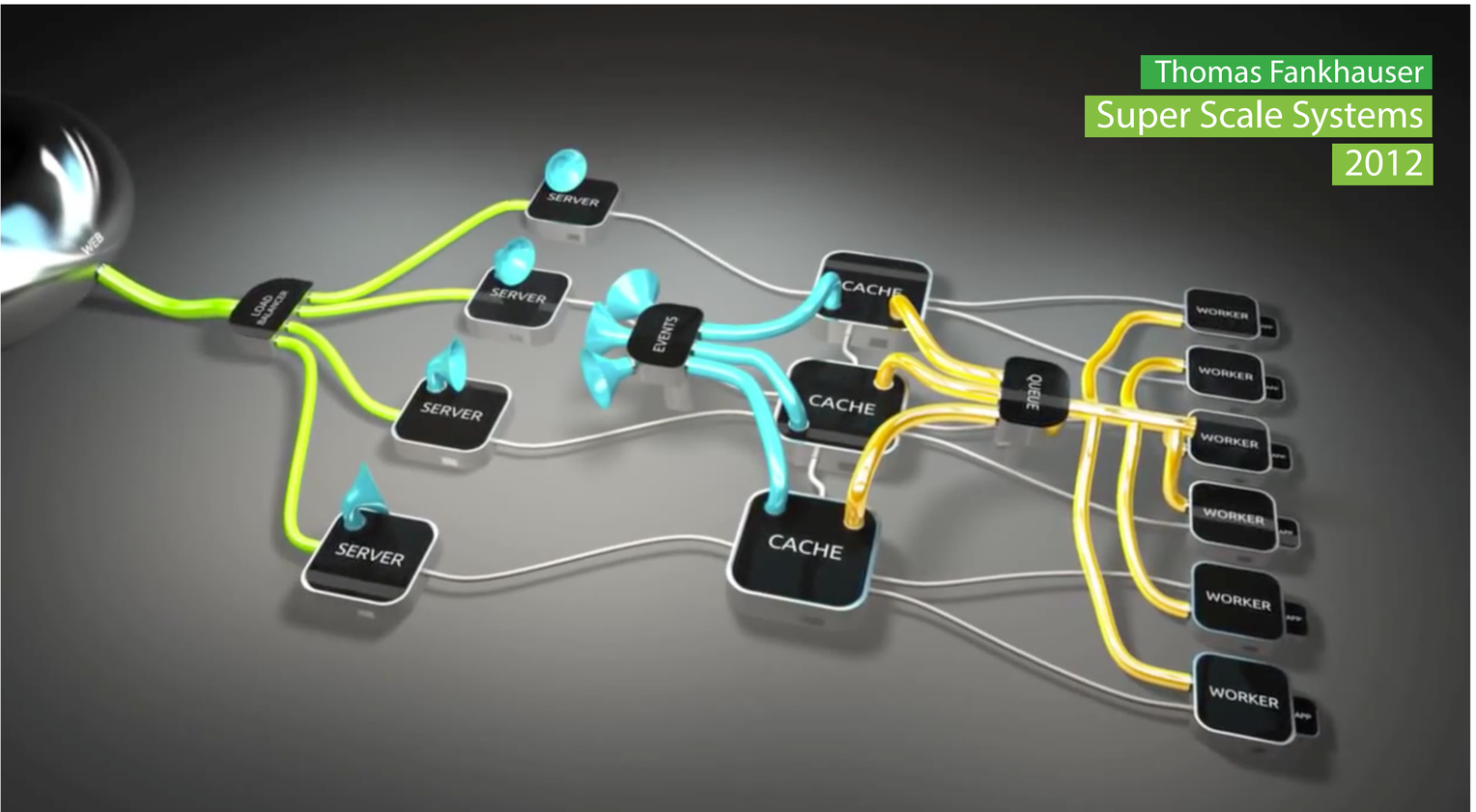
Web Scaling Frameworks

PhD Transfer Event

Thomas Fankhauser
Qi Wang
Christos Grecos
Xinheng Wang
Ansgar Gerlicher

Original Idea

Master´s Thesis



Thomas Fankhauser
Super Scale Systems
2012

Thomas Fankhauser
Master Thesis

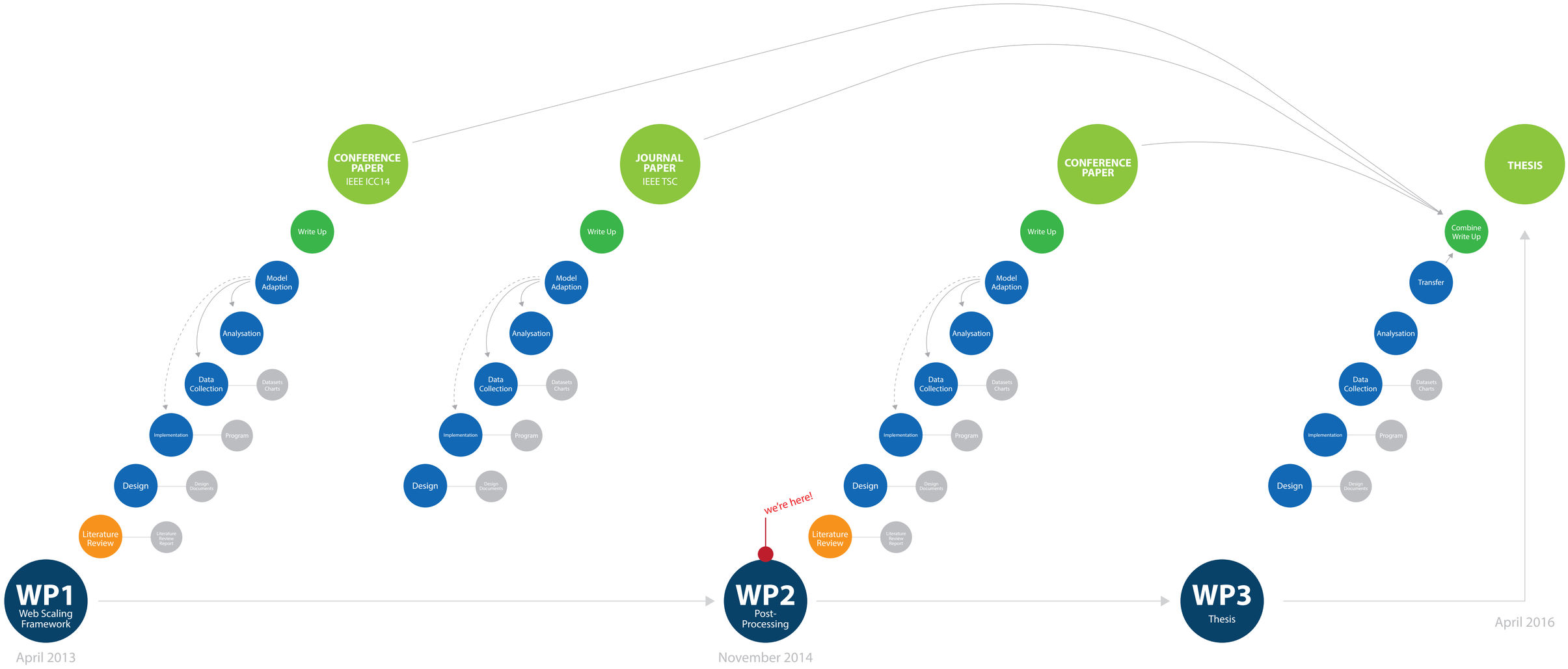
SUPER SCALE SYSTEMS

HOW TO SCALE THE SOCIAL WEB

Media University
Stuttgart

Project Roadmap

Work Packages & Publications



Work Package 1

Web Scaling Frameworks: IEEE ICC Conference Paper

CONFERENCE PAPER
IEEE ICC14

Write Up

Model Adaption

Analysation

Data Collection — Datasets Charts

Implementation — Program

Design — Design Documents

Literature Review — Literature Review Report

Presented at IEEE Flagship Conference:
IEEE International
Conference on Communications 2014
in Sydney, Australia

Web Scaling Frameworks: A novel class of frameworks for scalable web services in cloud environments

Thomas Fankhauser*†, *Student Member*, IEEE, Qi Wang*, *Member*, IEEE, Ansgar Gerlicher†, *Member*, IEEE, Christos Grecos*, *Senior Member*, IEEE and Xinheng Wang*, *Member*, IEEE

***Abstract*—The social web and huge growth of mobile smart devices dramatically increases the performance requirements for web services. State-of-the-art Web Application Frameworks (WAFs) do not offer complete scaling concepts with automatic resource-provisioning, elastic caching or guaranteed maximum response times. These functionalities, however, are supported by cloud computing and needed to scale an application to its demands. Components like proxies, load-balancers, distributed caches, queuing and messaging systems have been around for a long time and in each field relevant research exists. Nevertheless, to create a scalable web service it is seldom enough to deploy only one component. In this work we propose to combine those complementary components to a predictable, composed system. The proposed solution introduces a novel class of web frameworks called Web Scaling Frameworks (WSFs) that take over the scaling. The proposed mathematical model allows a universally applicable prediction of performance in the single-machine- and multi-machine scope. A prototypical implementation is created to empirically validate the mathematical model and demonstrates both the feasibility and increase of performance of a WSF. The results show that the application of a WSF can triple the requests handling capability of a single machine and additionally reduce the number of total machines by 44%.**

I. Introduction

The enormous growth of smart mobile devices in combination with social web services increases the number of requests that need to be processed by modern web platforms in a timely fashion. Whereas cloud computing provides the ability to provision the hardware needed, state-of-the-art Web Application Frameworks (WAFs) do not offer integrated scaling concepts to deal with automatic resource-provisioning and elastic caching or ensure a guaranteed maximum response time.

They are rather designed to abstract common functionalities needed for web application development including data-management, url-mapping, session-handling and response-generation. Today, users progressively access the social web from anywhere using their mobile smart devices, which leads to increased traffic. A single computing resource might not be able to satisfy such an amount of requests - only the junction of multiple computing resources, where each resource gets a small share of the total requests, allows to handle such huge amounts of requests in aggregation. Handling the exponentially increasing global requests adds the requirement of being able to run multiple instances of an application for highly scalable web services. The major challenges that are introduced by this requirement are the management of the shared resources, the balancing of the requests among all instances and the decision when to spawn or terminate instances. These challenges are collectively referred to as horizontal scaling [13], [14], [16].

Our experiments have showed that WAFs have different strengths and weaknesses. A highly abstracted WAF like Ruby on Rails, for example, was slower than the very thin WAF node.js but more powerful regarding data management and interface rendering. If a web service needs to provide both a fast and slim JSON API and a full blown HTML website it is the best solution to combine both WAFs. As both the horizontal scaling and web service composition are very complex matters, it makes sense not to introduce them to WAFs but offload them to another layer - the Web Scaling Framework (WSF) proposed in this paper. Fig. 1 illustrates a WSF that incorporates multiple WAF applications.

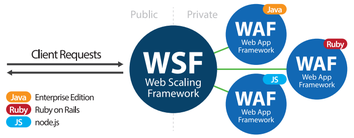


Fig. 1. **The relationship between the WSF and WAFs**

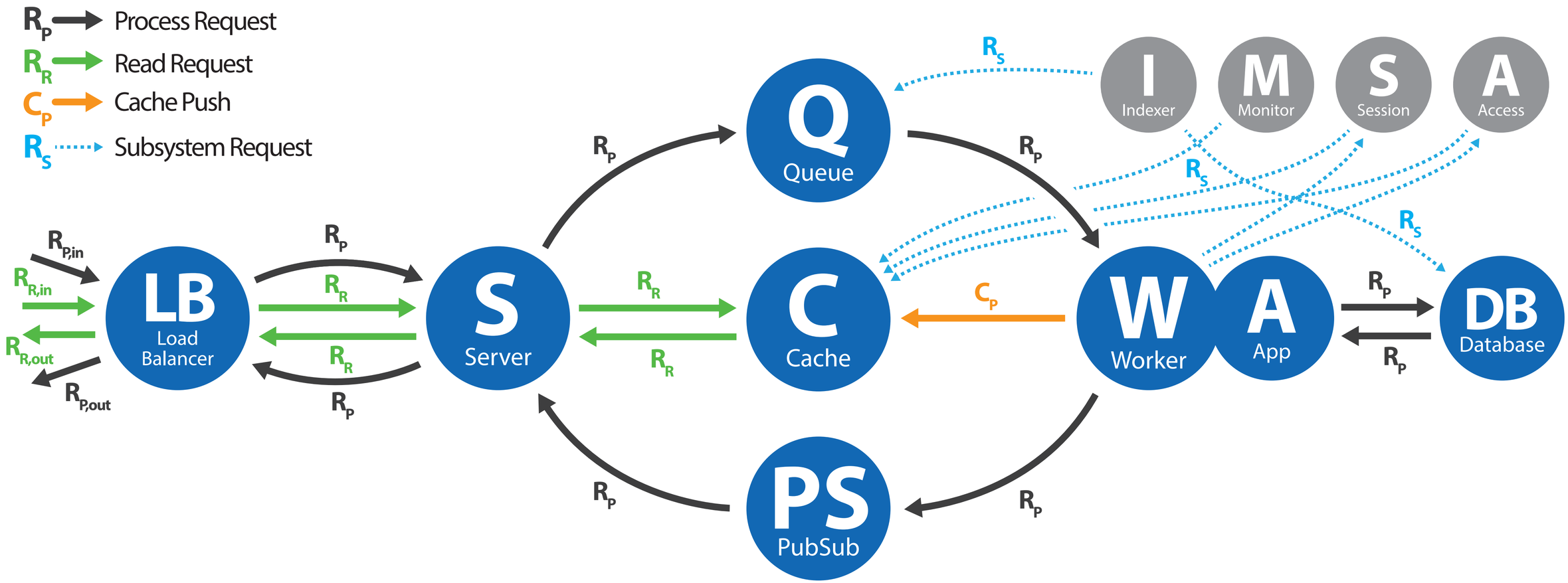
To comply to a proposed class of WSFs, a WSF should:

- take over the responsibilities of scaling and incorporate existing WAFs
- separate the business logic in the web service from the hosting logic
- connect to and combine existing WAFs to a compound web service using standard HTTP requests
- introduce low overhead when added, whilst adding the instant ability to scale
- constantly adapt their infrastructure to fit the required performance at all times

*School of Computing, University of the West of Scotland, Email: {Thomas.Fankhauser, Qi.Wang, Christos.Grecos, Xinheng.Wang}@uws.ac.uk
†Mobile Application Development, Stuttgart Media University, Email: {fankhauser, gerlicher}@hdm-stuttgart.de

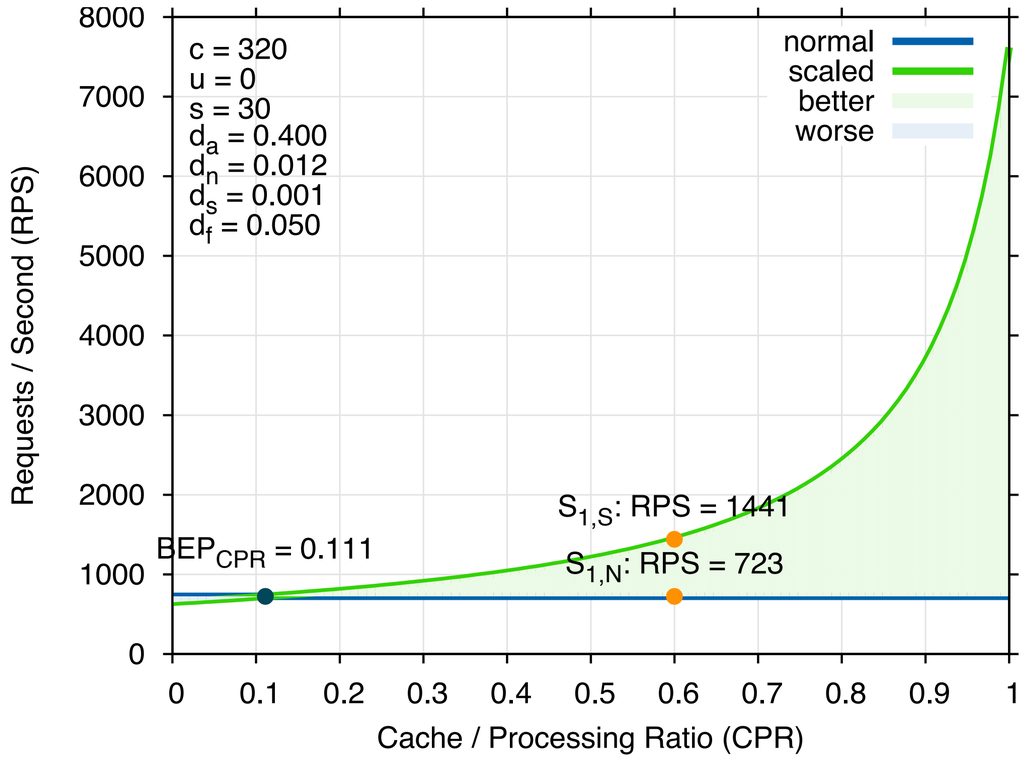
Work Package 1

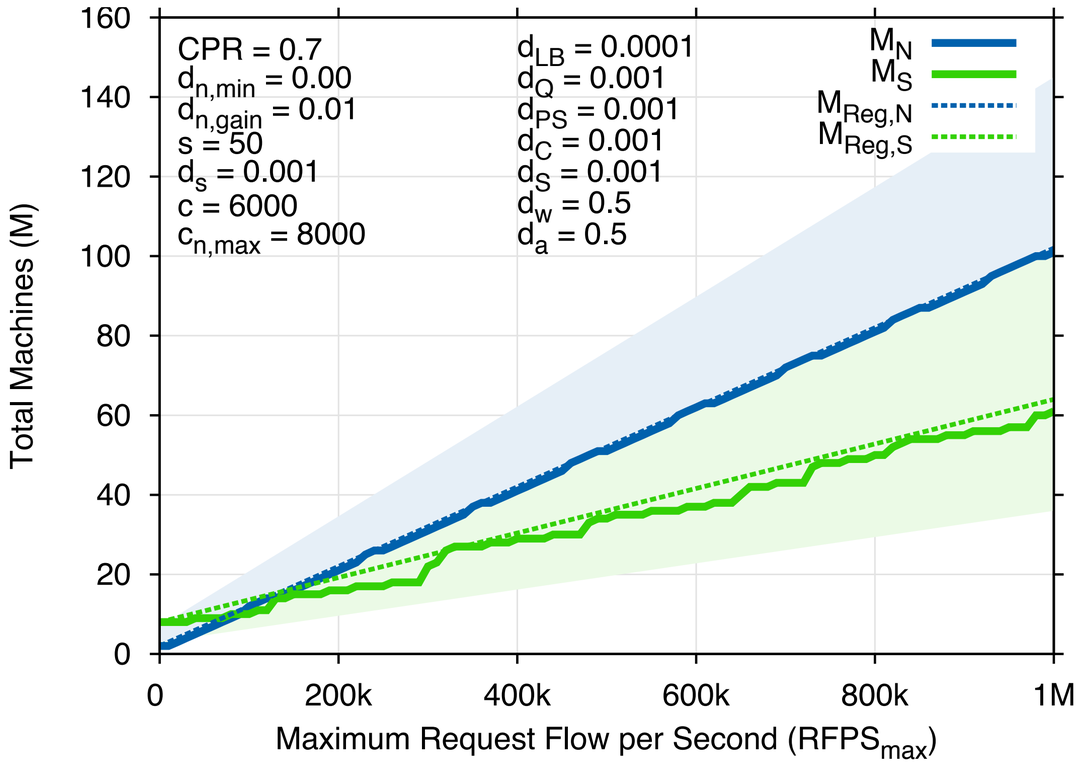
Web Scaling Frameworks: IEEE ICC Conference Paper



Work Package 1

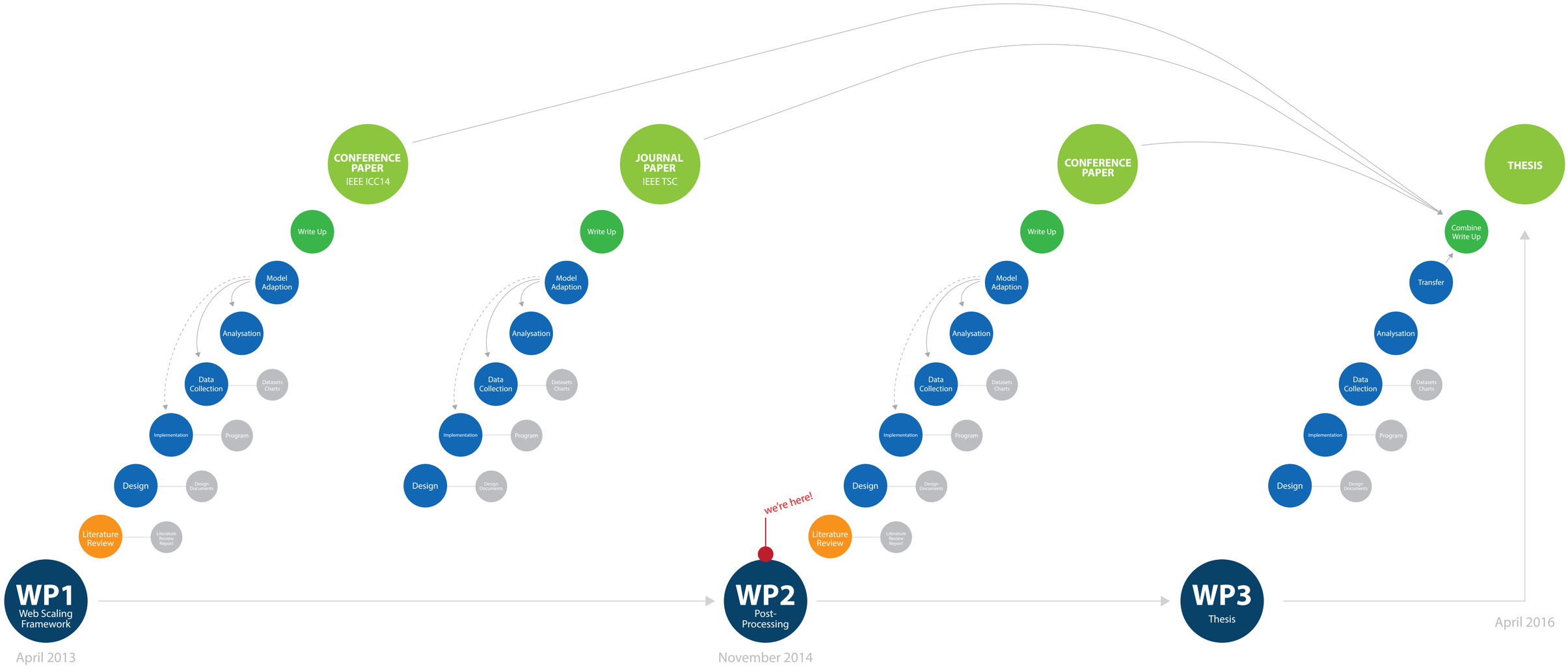
Web Scaling Frameworks: IEEE ICC Conference Paper





Project Roadmap

Work Packages & Publications



Work Package 1

Web Scaling Frameworks: IEEE TSC Journal Paper

JOURNAL PAPER
IEEE TSC

Introducing Elastic Scalability to Web Services in the Cloud with Web Scaling Frameworks

Thomas Fankhauser, *Student Member, IEEE*, Qi Wang, *Member, IEEE*, Ansgar Gerlicher, *Member, IEEE*, Christos Grecos, *Senior Member, IEEE*, and Xinheng Wang, *Member, IEEE*

Abstract—In the social web, web services have to accommodate a significant number of requests due to the high interactivity of current applications. Applications have to be built in a scalable fashion so the number of machines can be adapted to highly dynamic traffic situations. In the deployment context, web service providers often need to focus initially on the business logic which prevents detailed scalability considerations. If however a critical mass of customers is reached, providers need to be able to scale-up immediately their web services to stay in business. State-of-the-art Web Application Frameworks (WAFs) focus on the creation of application logic including data validation, view composition and session handling. However, they don't offer integrated cloud scaling concepts that handle the automatic provisioning of resources. Web service providers have to create custom-built systems that consider scalability issues manually. As the creation of such scaling-systems is a very complex, we proposed in our recent work the concept of Web Scaling Frameworks (WSFs) in order to offload scaling to another layer of abstraction. WSFs are composed of traditional WAFs with multiple other components to provide scalability right from the launch of the deployment cycle. In this work, a detailed design for WSFs including necessary modules, interfaces and components is presented. A mathematical model used for performance rating is evaluated and enhanced on a computing cluster of 42 machines. Traffic traces from over 25 million real-world applications are analysed and evaluated on the cluster to compare the WSF performance with a traditional scaling approach using WAFs and caches. The results show that the application of WSFs can reduce the number of total machines needed for three representative real-world applications — a social network, a trip planner and the FIFA World Cup 98 website — by 32%, 63% and 92% respectively.

Index Terms—scalability, web service, cloud computing, web scaling frameworks, performance evaluation, software architecture

1 Introduction

The demands for modern web services increase due to the soaring social nature of the web and the upsurge of the total number of mobile devices. Web services have to deal with a high level of interactivity in applications, which in turn introduces enormous amounts of requests. Static websites are replaced by dynamic and highly interactive applications. For instance, TV shows deploy apps that allow users to influence the course of the show, advertisements are brought to customers only if they remain in the vicinity of advertised target locations, smart sensors deliver data for all kinds of metrics which users see on their mobile devices, and cars communicate traffic situations, report traffic jams and find intelligent routes based on live data.

The above scenarios introduce new challenges to web service providers and developers. As single-server systems are not able to handle the increased load, applications need to be built in a scalable fashion. Requests have to be balanced over all available machines, resources need to be shared without conflicting versions, distributed transactions have to be processed in a fault tolerant manner, and the number of machines has to be adapted to highly dynamic traffic situations. Typically, web service providers need to reach a critical mass of users to be successful. If the critical mass is reached, the web services need to be able to scale-up immediately to stay in business. Before this threshold is reached, providers need to focus on the business logic, which often prevents detailed scalability considerations.

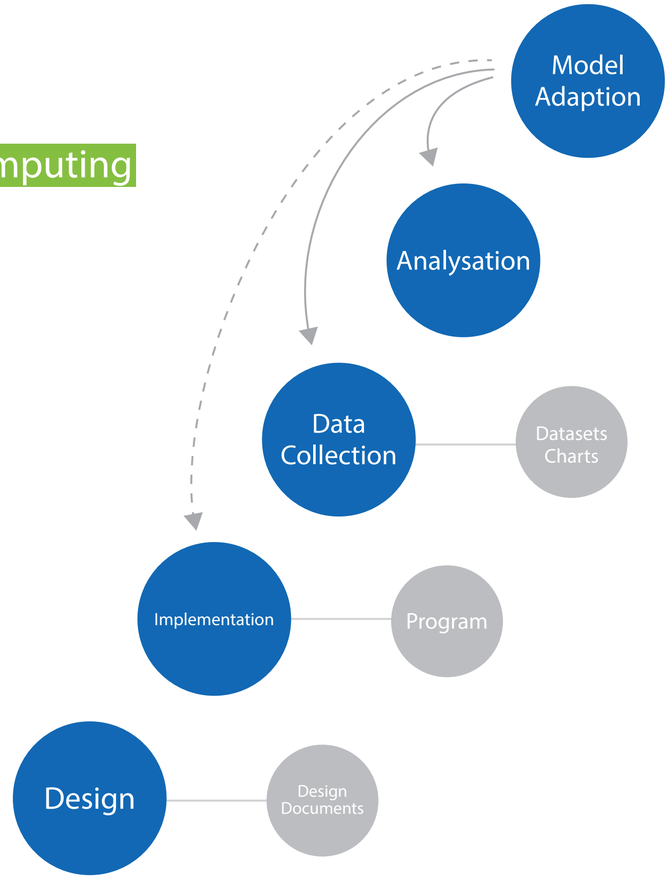
State-of-the-art Web Application Frameworks (WAFs) are designed to abstract common functionalities needed for the efficient implementation of web services. They focus on the creation of application logic, data structures, data validation, view layer presentation and session handling. They don't offer integrated scaling concepts that handle the provisioning of resources, manage elastic caching or ensure guaranteed response times. Today, web service providers have to create custom-built systems that consider these scalability issues manually.

The creation of such scaling systems is a very complex matter. Hence, we proposed the introduction of Web Scaling Frameworks (WSFs) in our recent work [1]. WSFs offload scaling to another layer of abstraction. They take over the responsibilities of scaling by embedding existing WAFs in a larger system. The application logic stays on the side of the WAFs while the scaling logic is provided by the WSF. Fig. 1 illustrates the interplay between a WSF and multiple WAFs. To utilise an existing interface, the frameworks communicate with each other using HTTP. WSFs provide instant scalability to common WAF applications whilst only introducing a low overhead. To meet the performance requirements at all times, the infrastructure is adapted automatically. Resources are provisioned on a pay-per-use basis to benefit from the concept of cloud computing. WSFs are able to transparently use Software-as-a-Service (SaaS) or machine-cluster components.

- *T. Fankhauser, Q. Wang and X. Wang are with the School of Computing, University of the West of Scotland*
  E-mail: {Thomas.Fankhauser, Qi.Wang, Xinheng.Wang}@uws.ac.uk
- *C. Grecos is an Independent Imaging Consultant.*
  E-mail: grecoschristos@gmail.com
- *A. Gerlicher is with the Media University Stuttgart.*
  E-mail: gerlicher@hdm-stuttgart.de

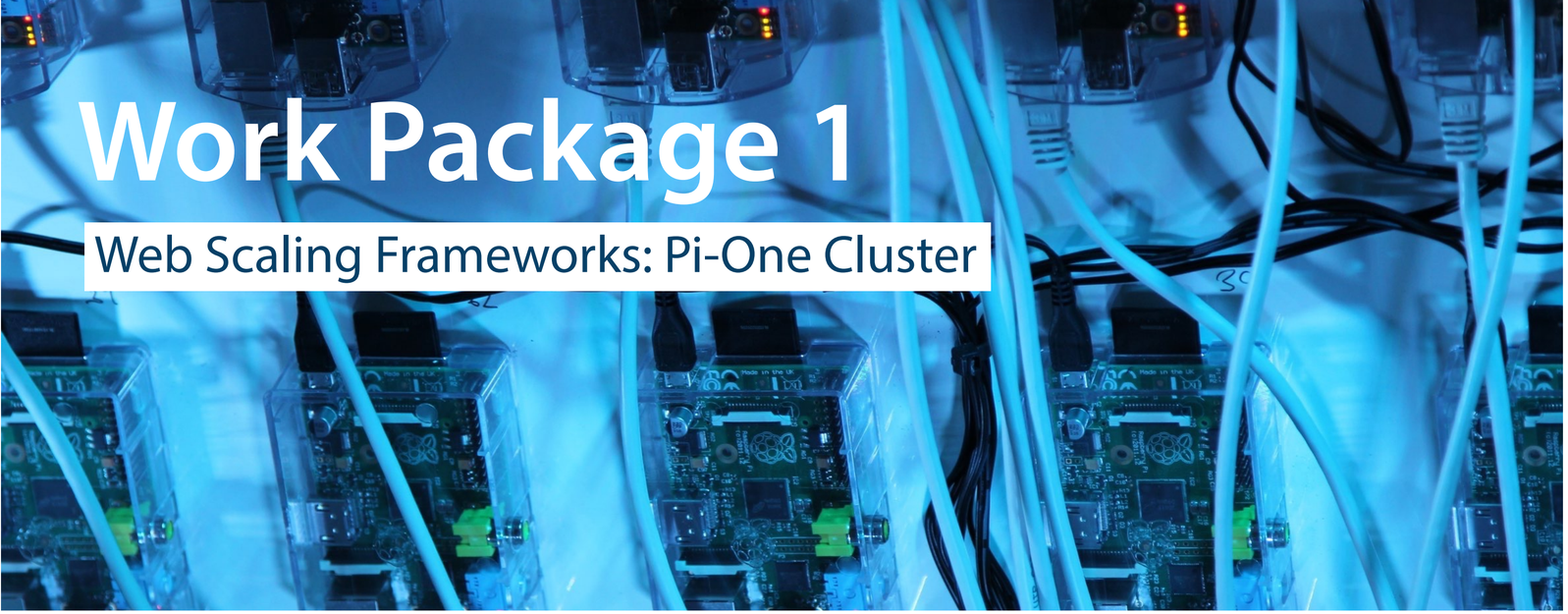
Write Up

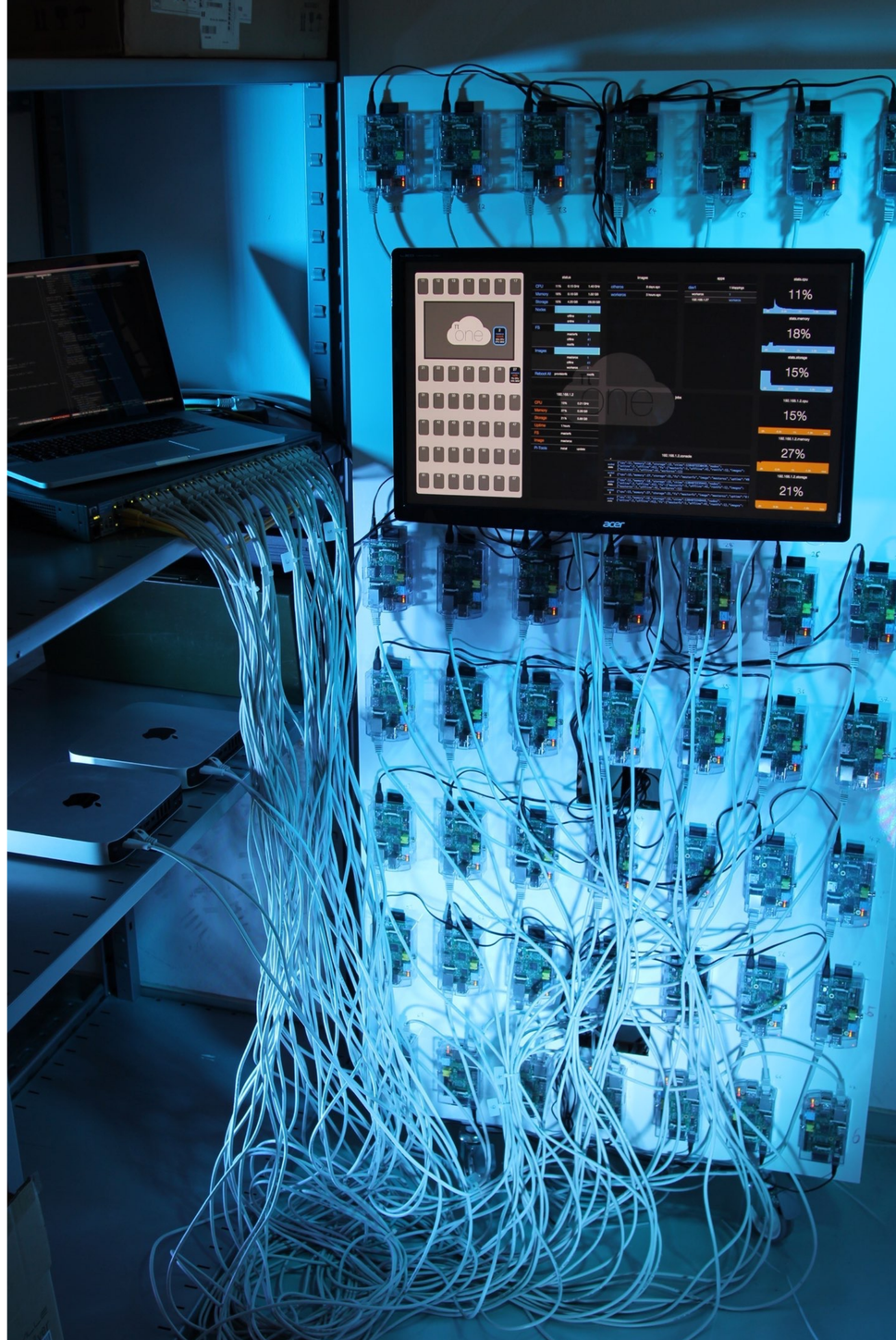
Submitted to:
IEEE Transactions on Services Computing



Work Package 1

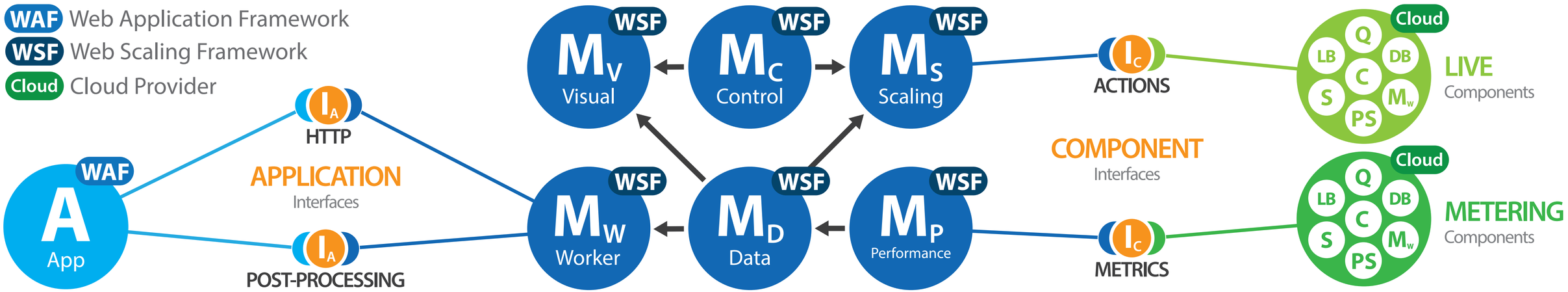
Web Scaling Frameworks: Pi-One Cluster





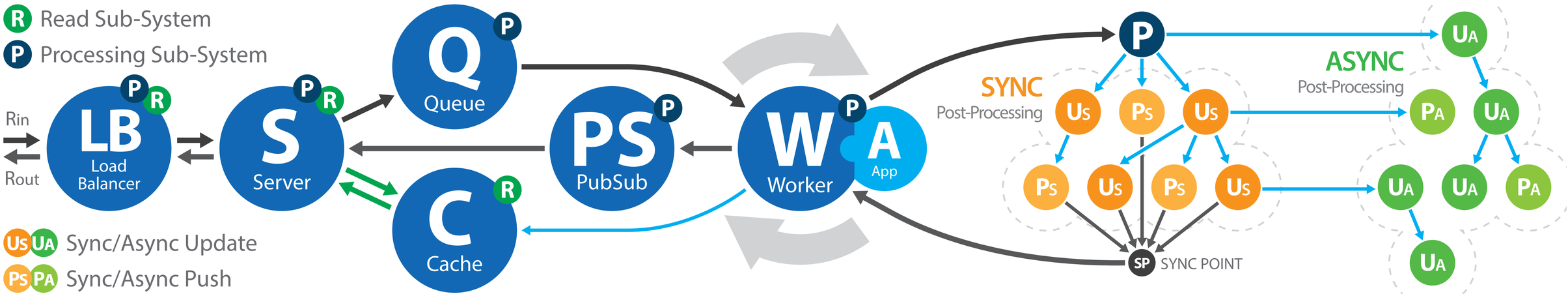
Work Package 1

Web Scaling Frameworks: IEEE TSC Journal Paper



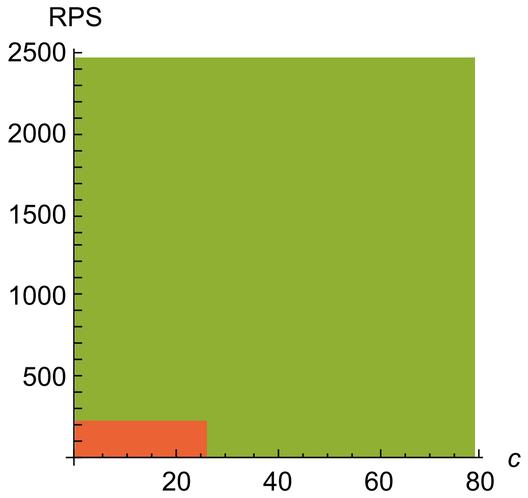
Work Package 1

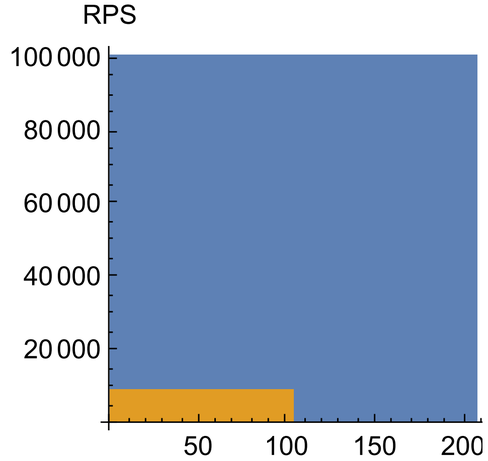
Web Scaling Frameworks: IEEE TSC Journal Paper



Work Package 1

Web Scaling Frameworks: IEEE TSC Journal Paper

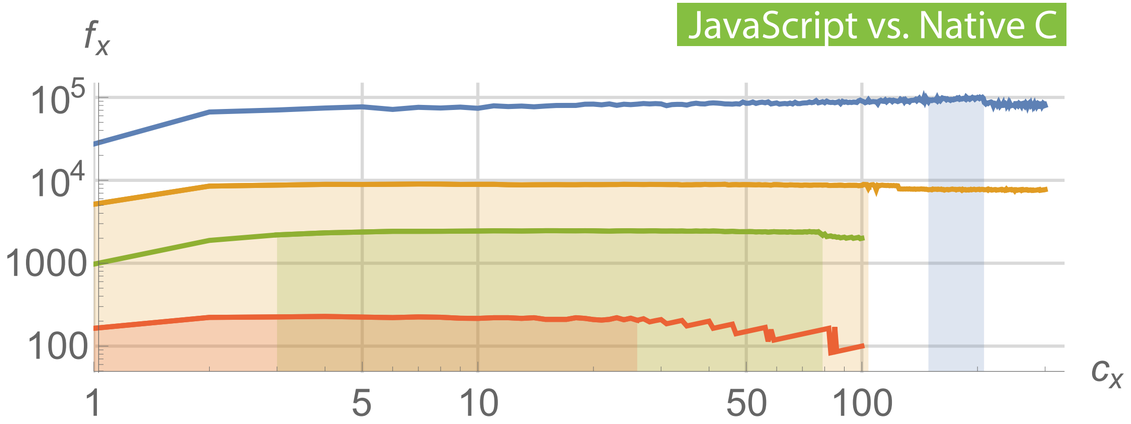


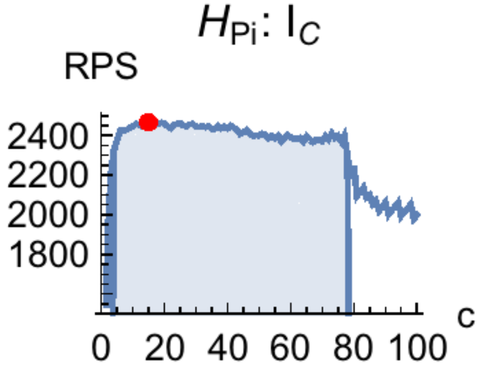


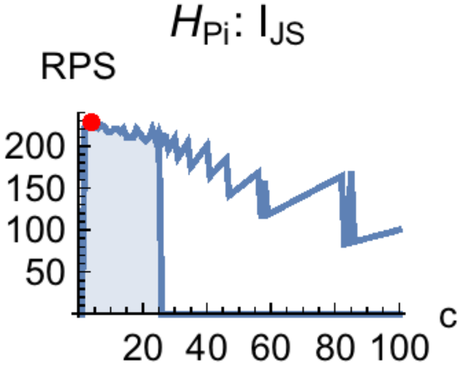
Hardware vs. Software

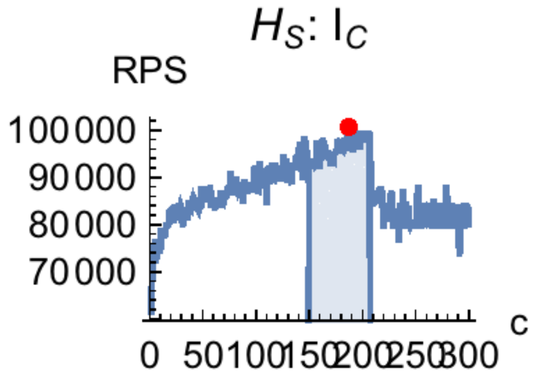
Pi vs. Server

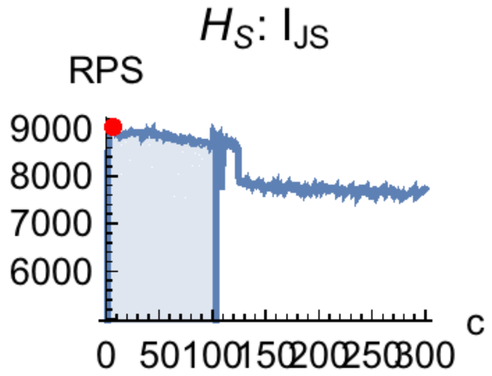
JavaScript vs. Native C





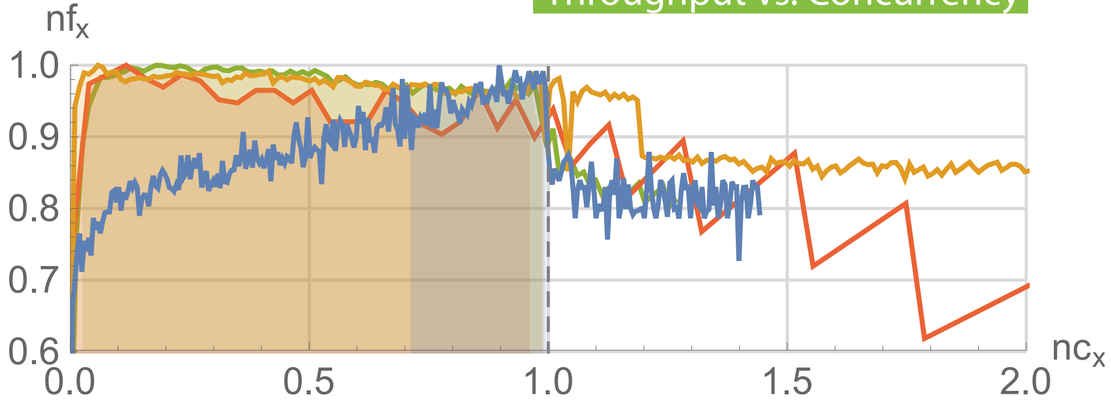






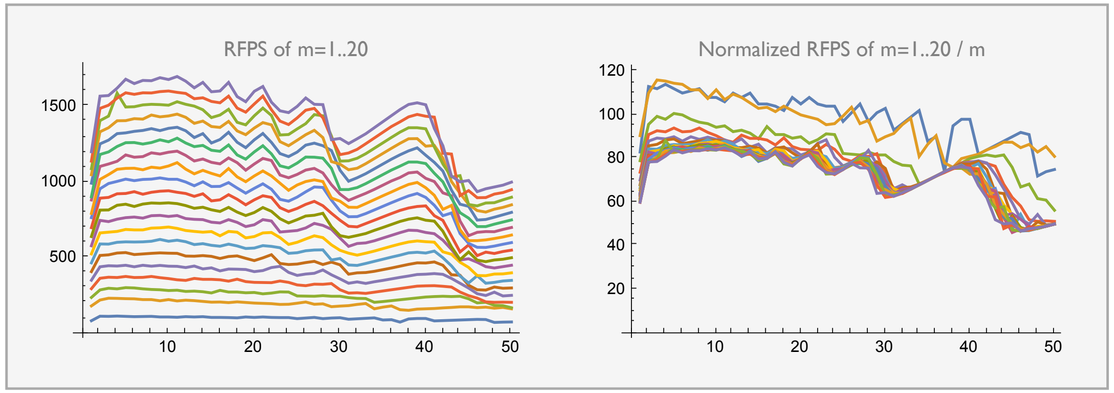
Evaluation

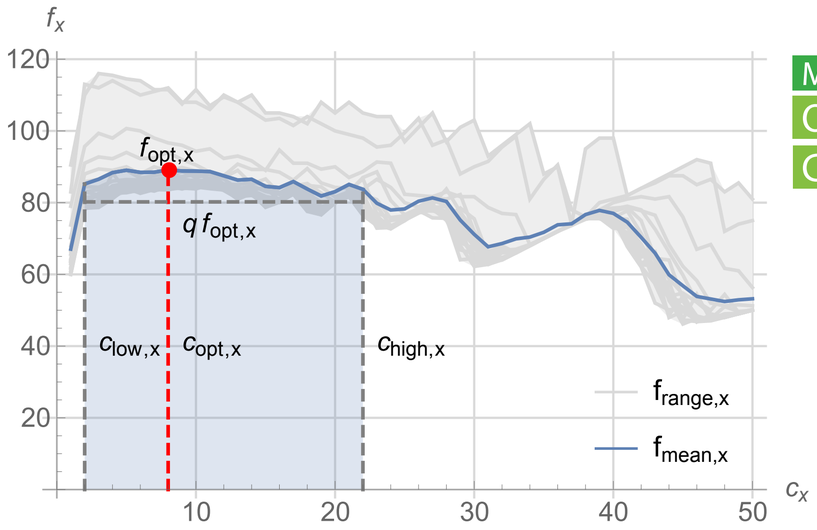
Throughput vs. Concurrency



Work Package 1

Web Scaling Frameworks: IEEE TSC Journal Paper

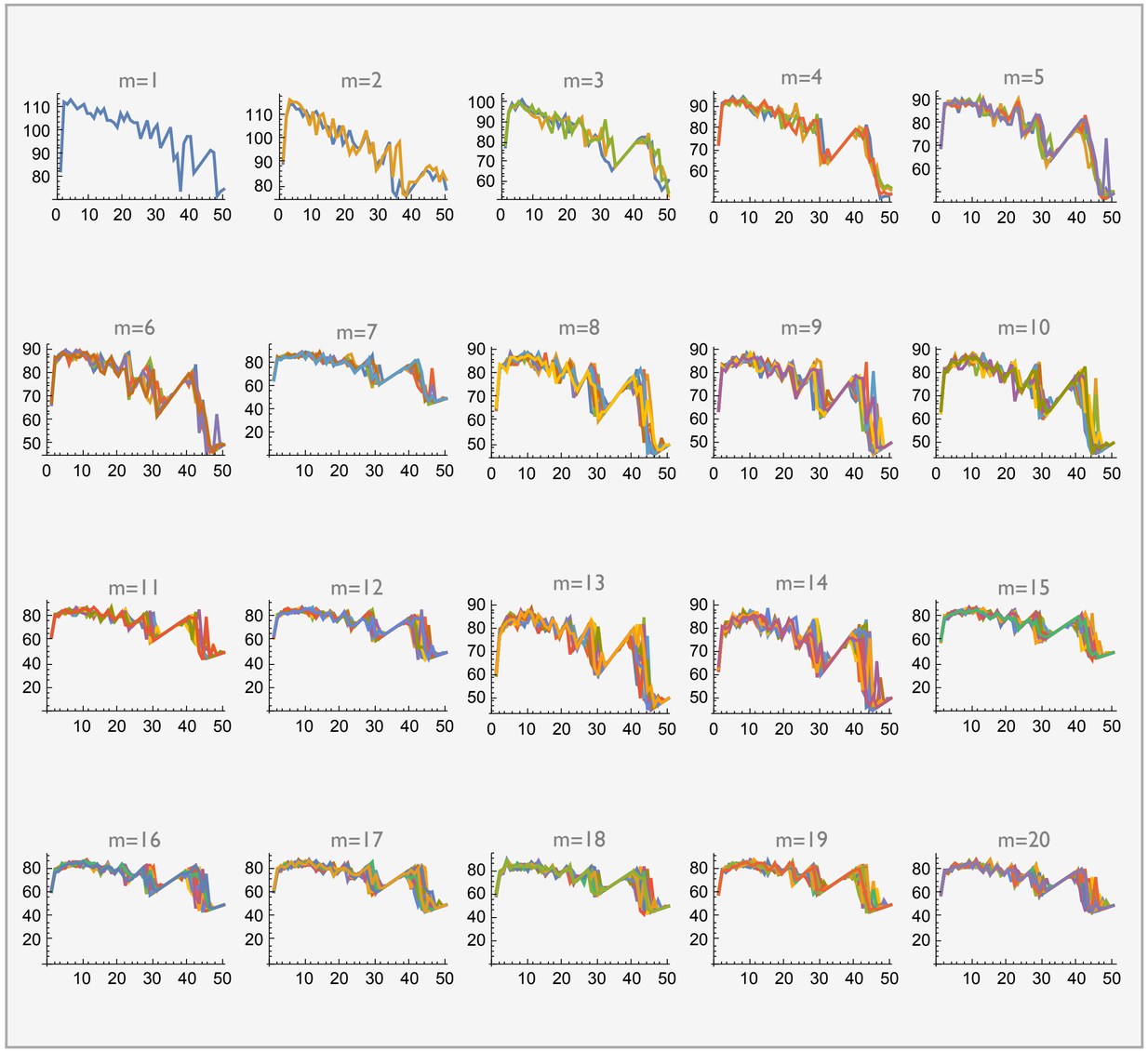




Measured

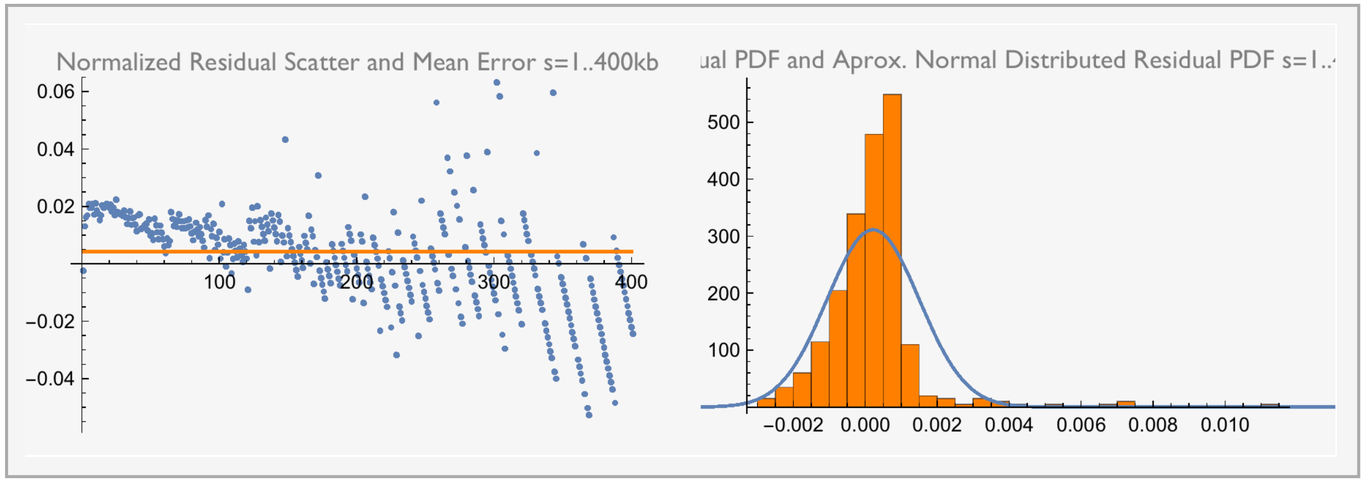
Optimal

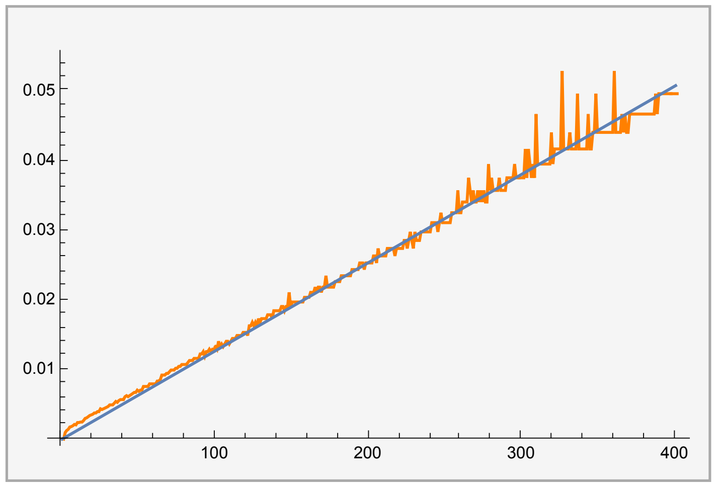
Concurrency Range



Work Package 1

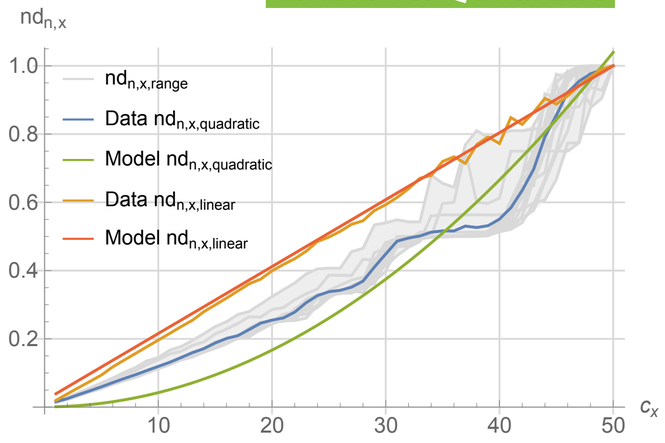
Web Scaling Frameworks: IEEE TSC Journal Paper

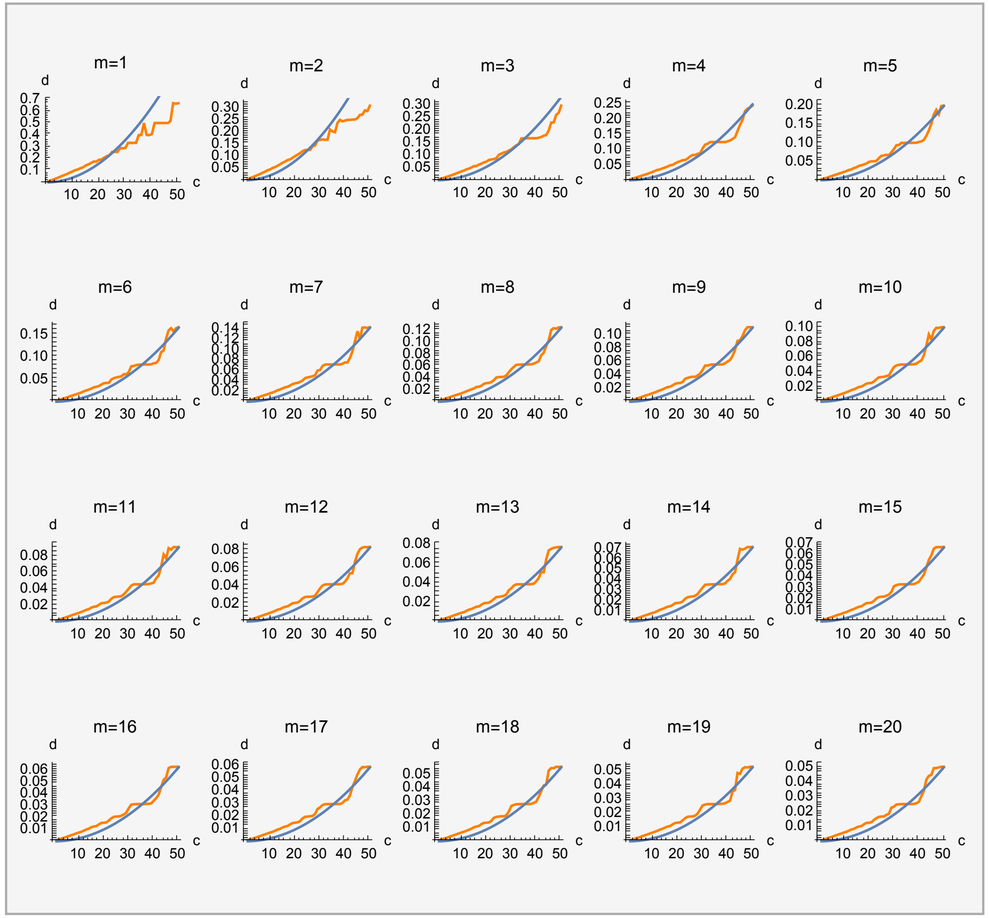


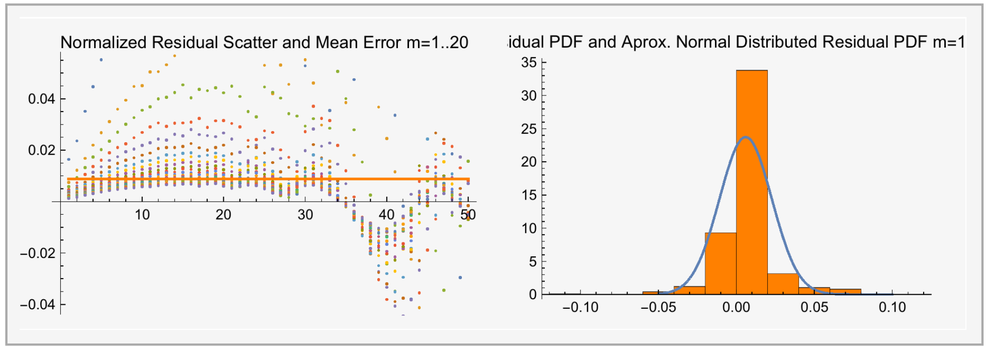


Network Delay

Linear vs. Quadratic





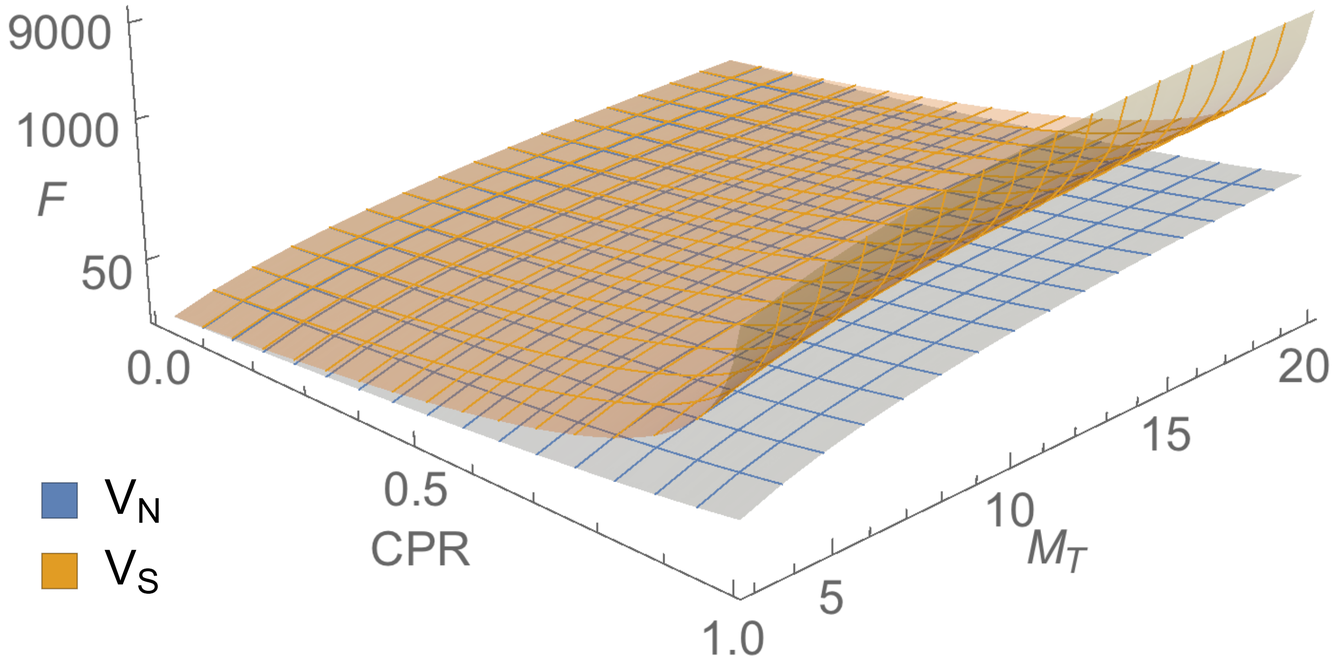


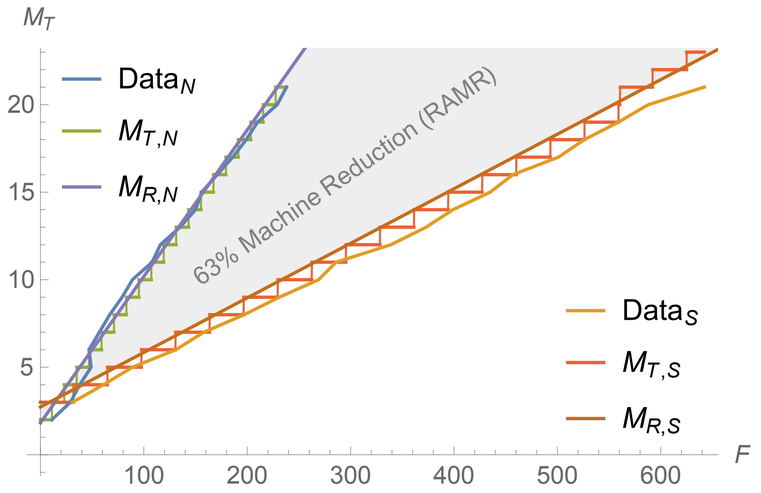
Work Package 1

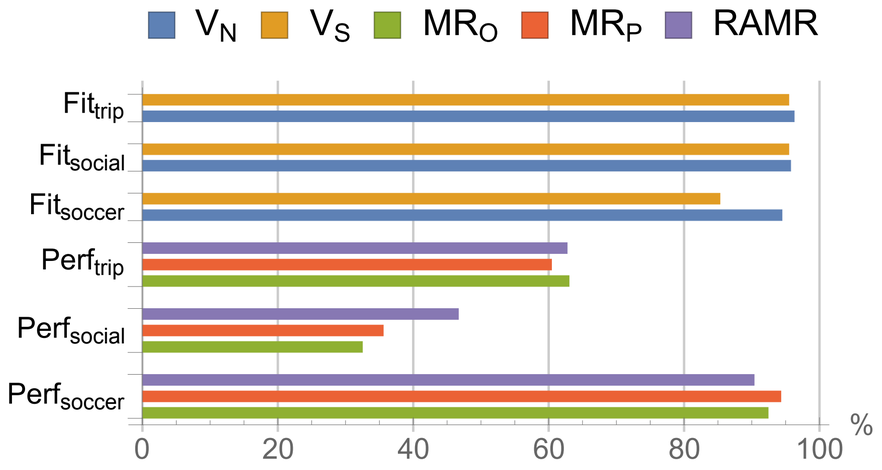
Web Scaling Frameworks: IEEE TSC Journal Paper

Evaluation

Normal Version vs. Scaled Version

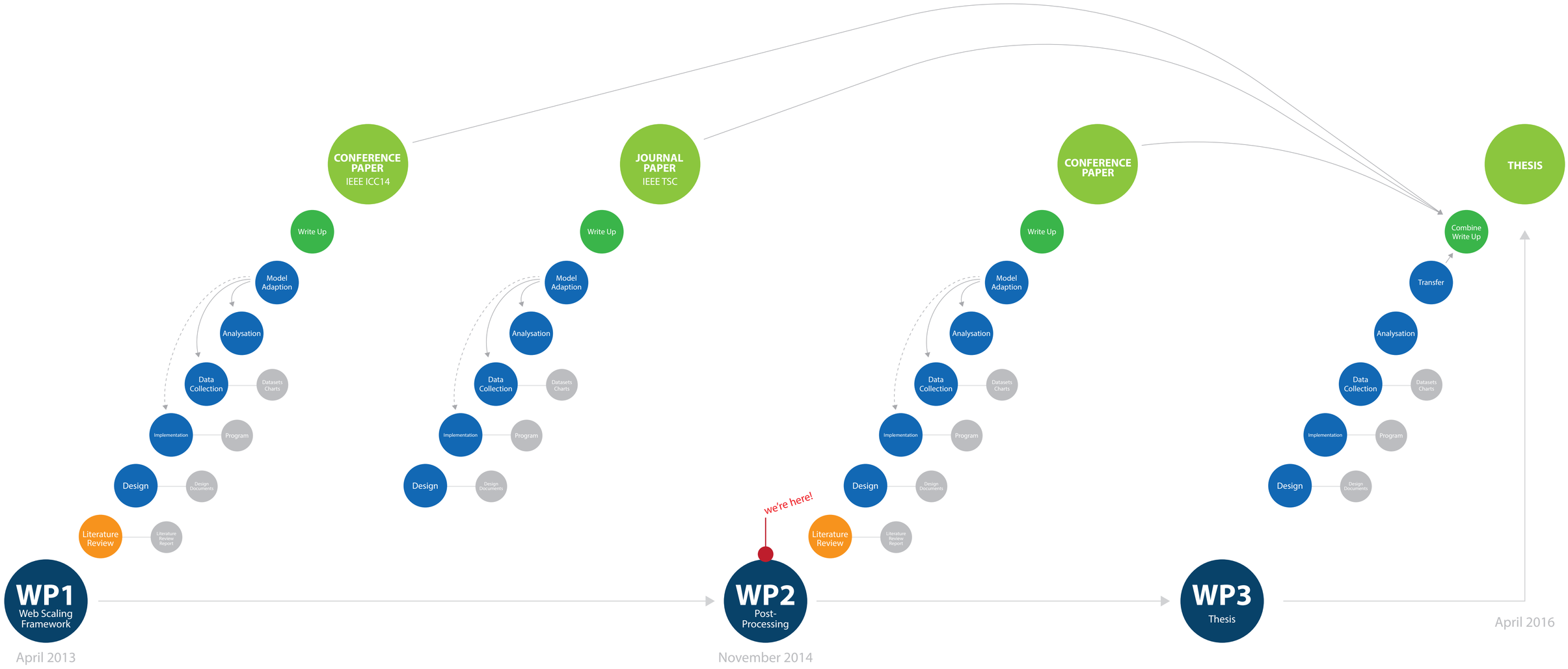






Project Roadmap

Work Packages & Publications

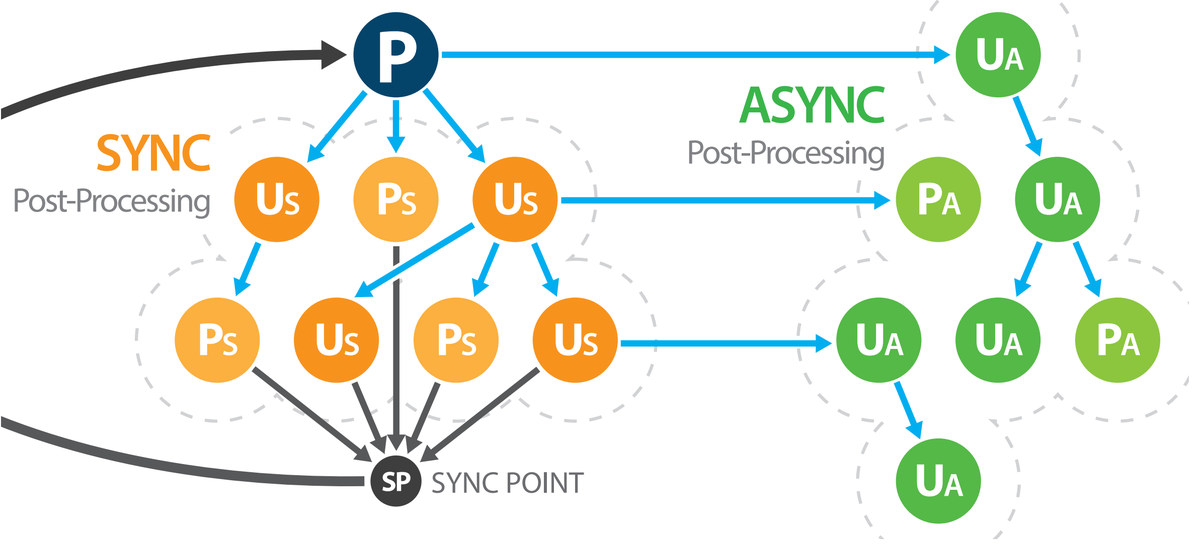


Work Package 2

Reactive/Proactive Post-Processing: Conference Paper

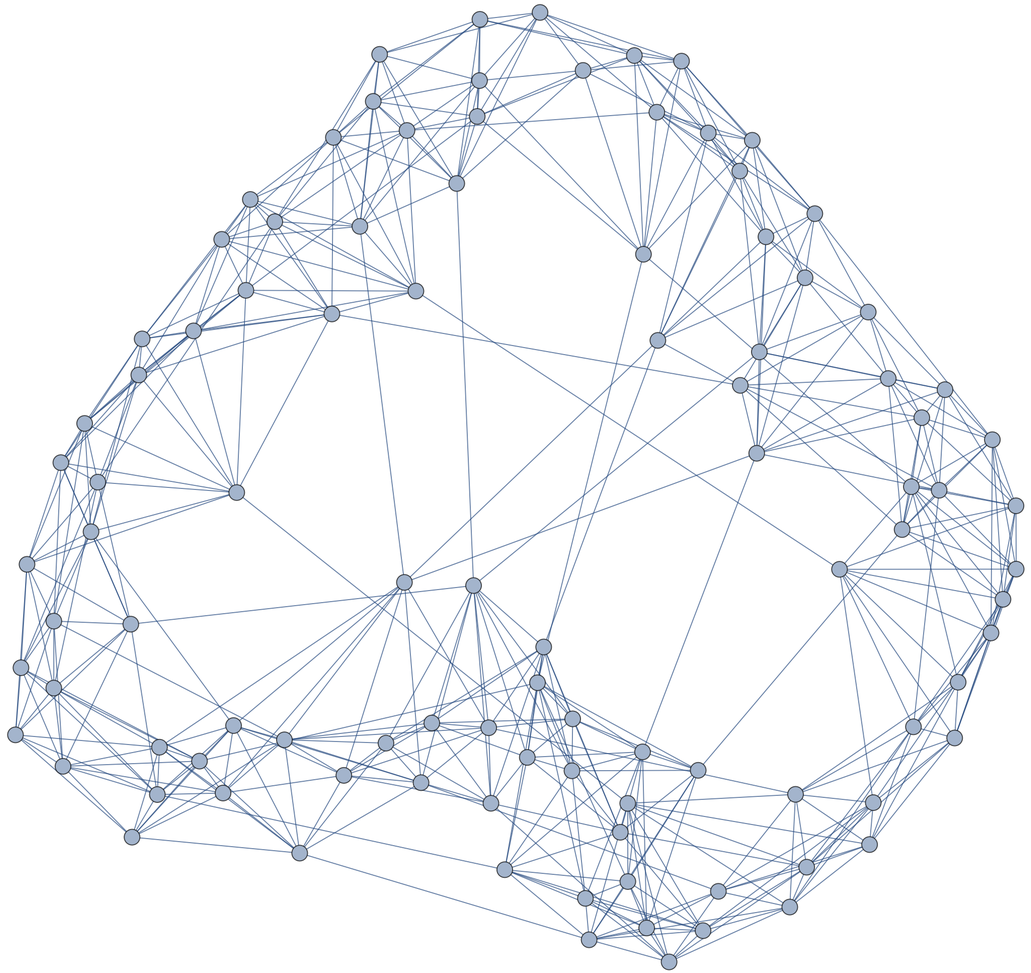
Next focus:

Find data structures and algorithms that allow a fast, distributed and parallel processing of dependencies



Work Package 2

Reactive/Proactive Post-Processing: Conference Paper



Dependency Analysis

Data Structures

Declaration

Generation

Visualisation

Link Analysis

Work Package 2

Reactive/Proactive Post-Processing: Conference Paper

Pre/Post-Processing

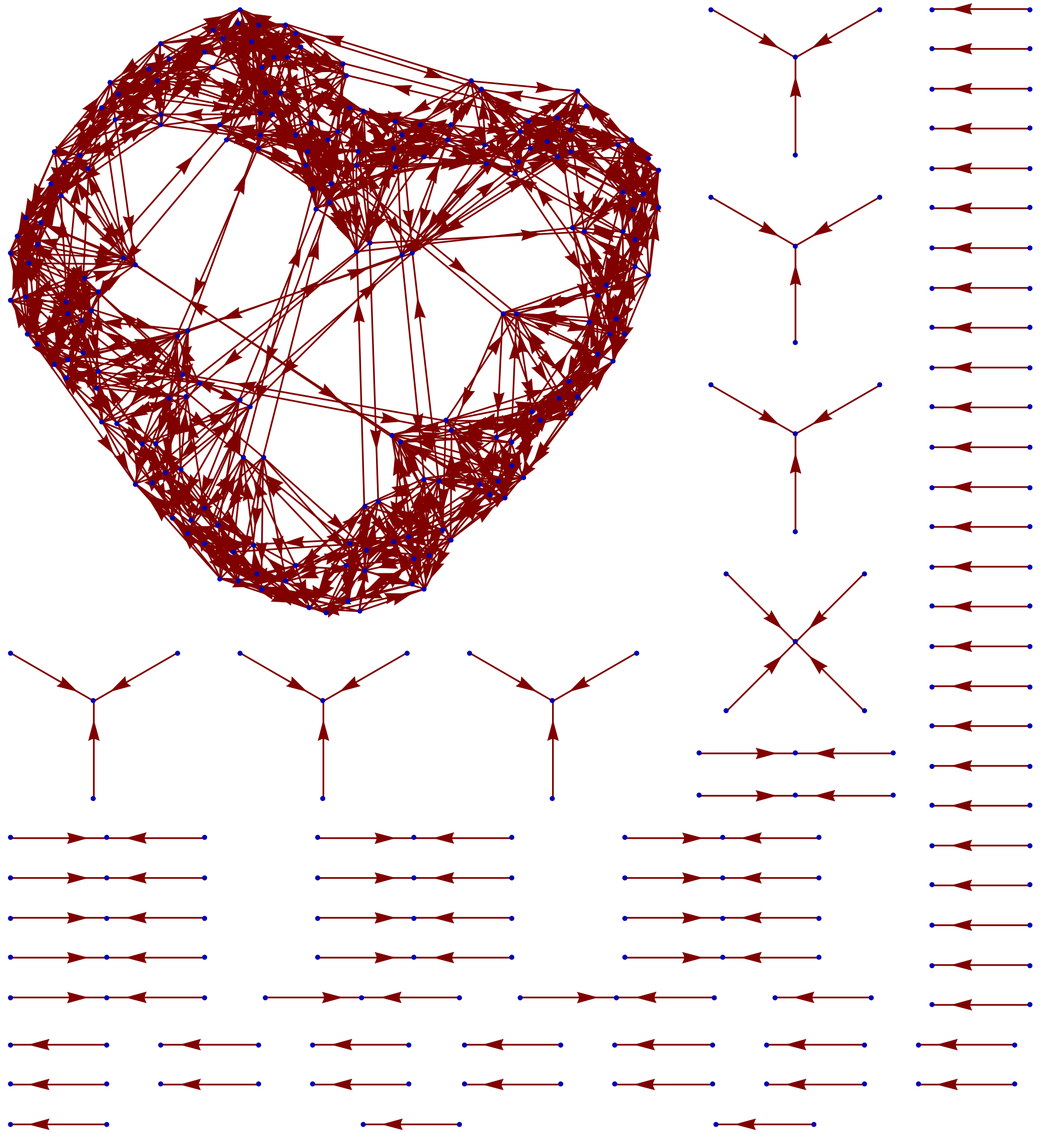
Synchronous vs. Asynchronous

Optimisation

Parallelisation

Fragmentation

Eventual Cache

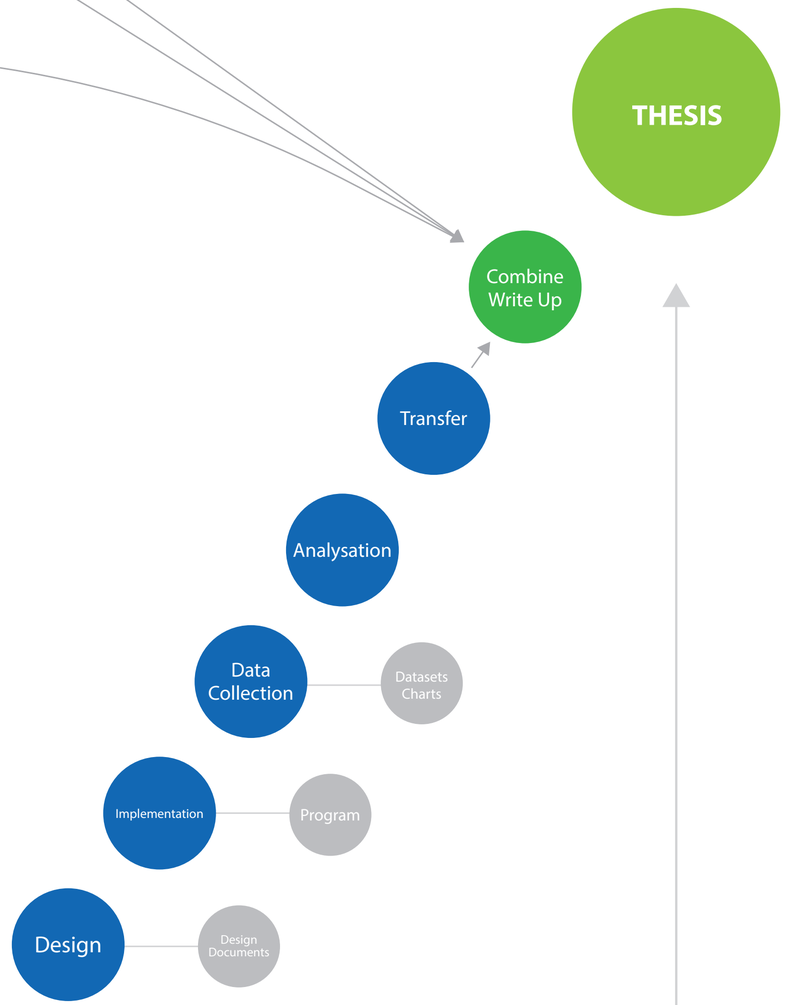


Work Package 3

Web Scaling Frameworks: Thesis

Thesis focus:

Create and evaluate a full-stack Web Scaling Framework with dependency declaration and optimised post-processing algorithms



WP3
Thesis

April 2016

UWS UNIVERSITY OF THE WEST of SCOTLAND

HOCHSCHULE DER MEDIEN

Web Scaling Frameworks

PhD Transfer Event

Thomas Fankhauser
Qi Wang
Christos Grecos
Xinheng Wang
Ansgar Gerlicher