

Research Reports in Computer Science

Band 7

**Andreas Wespi**

**History-Based Batch Job Scheduling on a  
Network of Interactively Used Workstations**

Shaker Verlag  
Aachen 2000

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

*Wespi, Andreas:*

History-Based Batch Job Scheduling on a Network of  
Interactively Used Workstations / Andreas Wespi.

Aachen : Shaker, 2000

(Research Reports in Computer Science ; Bd. 7)

Zugl.: Basel, Univ., Diss., 2000

ISBN 3-8265-8295-0

Copyright Shaker Verlag 2000

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publishers.

Printed in Germany.

ISBN 3-8265-8295-0

ISSN 1436-6967

Shaker Verlag GmbH • P.O. BOX 1290 • D-52013 Aachen

Phone: 0049/2407/9596-0 • Telefax: 0049/2407/9596-9

Internet: [www.shaker.de](http://www.shaker.de) • eMail: [info@shaker.de](mailto:info@shaker.de)

# History-Based Batch Job Scheduling on a Network of Interactively Used Workstations

Abstract

Andreas Wespi  
Department of Computer Science  
University of Basel

Networks of workstations are the computing environment at many sites nowadays. Workstations are used for daily interactive work but also, owing to their excellent price/performance ratio, as dedicated computing servers. As it is known that many workstations sit idle in a network of interactively used workstations, approaches have been developed to make use of the idle resources by employing them for computation-intensive tasks.

Distributed Batch Job Scheduling Systems (DBJSSs) have been built that allow batch jobs to be run on interactively used workstations in a way that the impact on interactive users is kept to a minimum. Whereas batch job scheduling in a dedicated cluster of computing servers is well understood, scheduling in a network of interactively used workstations, due to the unpredictability of the available resources, is quite different and has only marginally been investigated in the past. As a consequence, either simple scheduling strategies are implemented in current DBJSSs or the system administrators are given the means to implement their own scheduling strategy without any further assistance.

I have analyzed the workstation utilization data of the distributed computing environment of the IBM Zurich Research Laboratory for more than one year. Based on a new approach, I have evaluated the impact of different definitions of the workstation idle state on the total amount of resources available for batch job processing. Furthermore, I could show that many workstations have a regular usage profile. This observation has led to the development of novel, history-based batch job sched-

uling algorithms for networks of interactively used workstations. To quantify the performance benefits of the newly developed algorithms, the SIByL simulation environment has been built to analyze different batch job scheduling approaches under the same environment conditions.

Simulation results show that history-based batch job scheduling algorithms outperform the traditional scheduling approaches implemented in many DBJSSs. The traditional approaches are based only on the state of the workstations at the time when the scheduling decision takes place, whereas history-based algorithms also consider the workstation usage in the past. For moderately loaded systems and environments where job migration is of limited use, the history-based algorithms result in mean job turnaround times that are about 30% shorter than the times obtained with the traditional scheduling approaches.

History-based batch job scheduling results in an increased amount of data to be processed to find a good job placement. I show how current DBJSSs can be easily enhanced so that they support history-based batch job scheduling without generating a lot of additional network load or additional computing load on the client workstations. Increased computing load is placed only on the server side. Furthermore, the SIByL simulation tool can be integrated in the DBJSS server such that, based on prior workstation usage data and job execution statistics, the best scheduling algorithm is selected adaptively out of a set of predefined algorithms.