Programs running on different cores in a multicore server are often forced to share resources like off-chip memory, caches, I/O devices, etc. This resource sharing often leads to degraded performance, a slowdown, for the programs that share the resources. A job scheduler can improve performance by co-scheduling programs that use different resources on the same server. The most common approach to solve this co-scheduling problem has been to make job-schedulers resource aware, finding ways to characterize and quantify a program’s resource usage. We have earlier suggested a simple, program and resource agnostic, scheme as a stepping stone to solving this problem: Avoid Terrible Twins, i.e., avoid co-schedules that contain several instances from the same program. This scheme showed promising results when applied to dual-core servers. In this paper, we extend the analysis and evaluation to also cover quad-core servers. We present a probabilistic model and empirical data that show that execution slowdowns get worse as the number of instances of the same program increases. Our scheduling simulations show that if all co-schedules containing multiple instances of the same program are removed, the
average slowdown is decreased from 54% to 46% and that the worst case slowdown is decreased from 173% to 108%.

Stichworte: Co-scheduling; Same Process; Scheduling; Allocation; Multicore; Slowdown; Cluster; Cloud

Dewey Dezimalklassifikation (Liste): 000 Informatik, Wissen, Systeme

Herausgeber: Clauss, Carsten; Lankes, Stefan; Trinitis, Carsten; Weidendorfer, Josef

Kongress- / Buchtitel: Proceedings of the Joined Workshops COSH 2017 and VisorHPC 2017

Jahr: 2017

Monat: Jan

Seiten: 13--20

Print-ISBN: 978-3-00-055564-0

Revied: ja

Volltext / DOI: http://doi.org/10.14459/2017md1344414

Occurences:
- Hochschulbibliographie > 2017 > Fakultäten > Informatik > Informatik 10 - Lehrstuhl für Rechnertechnik und Rechnerorganisation (Prof. Schulz)
- Einrichtungen > Fakultäten > Fakultät für Informatik > Lehrstühle der Informatik > Informatik 10 - Lehrstuhl für Rechnertechnik und Rechnerorganisation (Prof. Schulz) > COSH2017

Entries: