"There's a big difference between knowing the name of something and knowing something." --- Richard P. Feynman requires large amount of Many-tasks Computing **High Throughput** computing power to reduce periods of time or for Computing (HTC) **Chapter 4. A Proactive Approach for** Chapter 2. From work stealing to reactive load balancing Chapter 3. Performance modeling and analysis short periods of time Chapter 6. Experiments Overview of the **Dynamic Load Balancing** evaluation an approach in CS to parallel also requires large 1.1. Overview amounts of computing gap between HPC & HTC. MTC but for longer time. emphasizes using many computing resources over short periods of time Environmental ` task-based parallel dynamic load and parallel computer clusters why we need a to complete many computation tasks. related models so far Setup applications balancing applications performance model more info more detailing the approach omputer cluster vs grid balancing strategies for applying in practice omputing vs data cluster which aspects people bound of load balancing performance of load Evaluation of Proactive Evaluation of Online Evaluation of Colimit/gap performance traditional algorithms behaviors Load Prediction refers to large amount of data and Load Balancing slowdown .g., pre-partitioning alg why this can work in current task-based programming different research fields, e.g., parallel computer archs? \_ model support \_ numberical simulations parallelism task migration prediction accuracy a type of computation in process different methods for that large computation & overhead task offloading to Appendix for Related Perf. Model experiments with experiments with reactive load problems are often divided work stealing balancing the load overhead synthetic mircorealistic application, to be solved in parallelism. balancing 4.1. Feedback task offloading benchmark, MxM Parallel computing has been taxonomy known as a dominant paradigm in computer architecture, 3.1. A performance model of reactive load balancing mainly refering to multi- parallel computing shared memory general idea based on core processors. issues: too late and issues: wrong decided reactive load balancing/ vary node vary thread vary ML time failed requested and insufficient migrated scales parallel computer a new performance parallel computers memory architectures 2.2. Task-based parallel programming models task-based task parallel applications algorithm for FB task hard to define machines with more experiment and profiled offloading because of appearing information on several levels parallel programmi thesis contribution Overview of two thesis contribution 2: models 4.2. ML-based task offloading **Chapter 7: Conlcusion & Future Work** general idea for M observations in previous performance model based task offloading methods for performance slowdown to data-level example to illustrate 4.2.1. Reprequisite and Design bit-level data parallel distributed-mem programming model programming model programming model methods task-parallelsimdifferent from FBhow to define input/output how to imagine these instruction-level task offloading for online load prediction? how much imb by how much load and tasks operations at runtime?/ LB formulation in terms \_ perf\_slowdown? of task-based apps feedback task offloading extension: co-4.2.2. Online Load Prediction 3.1.1. Deterministic estimation cheduling tasks across controversal questions ML-based static LB \_multiple apps\_\_ bottlenecks with ask offloading bottlenecks with ML-models deterministic estimation input/output reactive load work stealing others factors at app task-based parallel side purpose dynamic LB & mention online load prediction outlook: results example with example with outlook: future constraints main operations Samoa2 factors at work to cost the model system side what does matter? 2.3. Task-based parallel applications and runtimes second contribution first contribution ove average bound naximum bound over a proactive a performance model 4.2.3. Proactive Task Offloading Algorithm approach task-based parallel task-based parallel master-worker runtimes task migration overhead balancing operation overhead detail how the K in average K in min-max exerpiments exerpiments \algorithm works ( how to define a task? 1.2. Problem formulation 3.1.2. Discrete time modeling task-based apps in & Motivation Functioning components An example or illustration common use case feedback task ML-based task example with task migration \[
\] offloading the idea of tasking in strategies a discrete time model principle co-scheduling task assumption across multiple apps mapping onto recursion use principle of the mapping reactive use case model state-of-the-art taskoperations to the model 4.3. Co-scheduling tasks across detail how to co-schedule based parallel runtimes multiple applications tasks across multi-apps parameters operations 2.4. Related work load balancing challange RQ1: How can we model the behavior of work stealing and example 2.4.1. Dynamic load balancing passive reactive load balancing to understand their limits in distributed memory systems? 3.2. Model simulation and evaluation **RQ2:** How can we proactively balance the load of task dynamic load balancing on parallel applications at runtime? ∖ task-based parallel apps ∫ extending the extending the performance model illustration for use SQ1: How can we predict the load of tasks to model simulation and proactive approach case 1 case 2 evaluation support proactive load balancing at runtime? **SQ1:** How can we proactively co-schedule reactive task offloading ) ( reactive task replication modularize the tasks to balance the load? somponents of simulation formulation & related parameters at app&sy context intro overview about DLB input parameters 3.2.1. Simulator Implementation Chapter 5. Proof of Concepts Overview of the and Contribution detailing objects and classes in \class&communication diagram (indirect params) proactive task offloading ) a proactive approach 3.2.2. Simulator Evaluation and Insight Chameleon the experiments of the experiments of 2.4.2. Work stealing framework real applications evaluation work stealing Balancing as A Plugin Tool with/without dynamicload Application as A Black proactive scheme for comigratable task definition scheduling tasks across multi) 3.3. Toward a proactive idea in load 2.4.3. Reactive load balancing balancing and co-scheduling tasks formulate the problem with a simulator influence parameters task-based reactive load balancing toward a proactive idea in runtime LB and co-scheduling tasks 1.5. Publications 1.6. Thesis outline communication ) reactive/proactive three representative pillars reactive task replication reactive task offloading motivate → proactive approach affecting task migration