HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Workload Management and Scheduling

Prof. Hasso Plattner

What is Workload Management and Scheduling?

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

- We understand
  - **Workload management** as a strategy to ensure that the service level objectives of different database queries (workloads) are met
  - **Scheduling** as a form of workload management that assigns queries (or parts of queries) to database resources, in order to meet a given set of service level objectives
- General remarks
  - Another form of workload management is **admission control**, which restricts the number of parallel queries in the database
  - In the context of enterprise computing, we have to deal with **dynamic scheduling**, as the set of queries is not known in advance
  - Scheduling decisions are expensive; complex scheduling in the presence of many queries is often prohibitively expensive (strategies like “fast in, fast out” or heuristics are appropriate)

Workload Management for Mixed Workloads

- Mixed workloads
  - Transactional Queries: short running, strict time constraints
  - Analytical Queries: long running, soft time constraints
- Running mixed workloads on a single database instance leads to potentially conflicting optimization goals
  - Guarantee transactional query throughput
  - Maximize overall query throughput
  - Minimize response-times for ad-hoc analytical queries

Some form of workload management is required to ensure service level objectives for transactional and analytical queries

Workload Management for Mixed Workloads (2)

- Pure admission control outside the database falls short, as long running queries cannot be controlled once entered the database
- Design hypothesis for scheduler: Separate transactional and analytical queries and dynamically assign resources depending on database load
  - Transactional queries
    - Complex query runtime prediction imposes too much overhead
    - Assign as many resources to OLTP queries required to guarantee throughput goals
  - Analytical queries
    - Schedule on the remaining resources as efficiently as possible
    - Adjust degree of concurrency depending on transactional load