technische universität
dortmund

Job Scheduling to Manage Power in Large Parallel Systems

Uwe Schwiegelshohn
TU Dortmund University

Benefit of an Energy Aware Algorithm Implementation

- You have developed and implemented great algorithms that run on large systems and promise to reduce the total amount of required energy by your application while achieving a reasonable response time.

- What is the benefit you can expect from your algorithm?
  - Peace of ecological conscience?
  - Reduced cost for running your application?
  - Any other goal?

Content

- Scheduling Objectives
- Considering Energy
- Amazon Web Services and Spot Instances
- Mid Range Planning
- Local Scheduling
- Conclusion

Common Scheduling Objectives

- C_{max}
  - Minimization of operating costs
  - Operating costs are fixed. Earlier completion leads to a decrease in the ratio *cost per operation time*.
- L_{max}, $\sum T_j$, $\sum U_j$
  - Observation of quality of service promises, minimization of penalties
- $\sum w_j C_j$, $\sum w_j T_j$, $\sum w_j U_j$
  - Consideration of multiple user objectives
  - Linear combination with individual weights
- We can use common objective functions to derive new objectives:
  - Example: Maximization of utilization $\sum p_j \cdot (1-U_j)$ with access control
- But energy does not really fit!

Consider Energy from an Application Point of View

- Dominant component of operating costs in large systems
  - Mainly usage dependent with complex relations
    - Storage
    - Computing
    - Communication
    - Cooling

- Application point of view (in the future)
  - Bicriteria optimization for a single problem: speed C_{max} and energy
    - Selection of the suitable Pareto optimum
    - Shifting the Pareto front by means of energy efficient algorithms

Consider Energy from a Producer Point of View

- Technical restrictions
  - No sudden increase of power delivery
  - No efficient storage of large amounts of energy

➢ Power demand estimation is necessary.
  - Many consumers of small amounts of power
    - Reliable stochastic prediction
  - Few consumers of large amounts of power
    - Purchase guarantee for a quantity of power in exchange for a better price and delivery guarantee

Owner of Large Systems

- Consumer of large amounts of power
- Focus in previous times
  - Keep as many nodes busy as possible to optimize the investment.
- Focus now
  - Reliably predict power consumption while supporting on demand computing

- Uncertainty
  - Computing requests in the future
  - Power consumption of each application
    - **Large volatility of power consumption makes the problem worse!**

Consequences of a Wrong Estimation

- Underestimation
  - Rejecting new users and/or terminating user applications
    - Users are not happy!
  - Buying additional power on the spot market
    - The price for a significant amount of power may be very high!

- Overestimation
  - Selling excess power on the spot market
    - There may not be enough customers for a large amount of power on a short notice!
  - Starting additional applications

Spot Instances: The Amazon Webservice Approach

- Underestimation
  - Spot instances can be terminated immediately
    - Users have agreed to this condition and receive the last fractional hour for free.
  - No need to buy additional power on the spot market

- Overestimation
  - New resources are made available for spot instances.
  - No need to sell power on the spot market if there is enough demand for spot instances

- Spot instances can keep spare machines busy while they are not needed.

Are Spot Instances the Perfect Solution?

- Other large cloud providers did not adopt the approach.

- Are there enough applications that do not mind sudden termination?
  - Applications that are able to do its own checkpointing
  - Applications that consists of many independent applications with short processing times (significantly less than an hour)

- Are spot instances significantly cheaper than on-demand instances?
  - Some people have developed schemes to provoke early termination and exploit repeated free computation time.
  - The bidding process occasionally produces prices that exceed on-demand prices.

Preemption Instead of Termination

- Large running batch jobs without deadlines
  - There is still a performance guarantee for these jobs.
- Underestimation
  - Running batch jobs are preempted.
- Overestimation
  - Preempted batch jobs are resumed or new batch jobs are started.
- Batch jobs may be internal or external.
  - Google uses jobs that re-index data bases.
- Compared to the processing time of a batch job the context switch penalty must be small enough to be ignored.
- There can only be a small number of context switches.

Hierarchical Algorithmic Approach

- The problem includes a large number of uncertainties that cannot be reliably predicted in advance.

- We propose a hierarchical approach consisting of a global schedule covering the power estimation range and a local schedule addressing the actual power consumption of near future.
  - Global schedule: we assume average power consumption for all applications using a specific type of machine and estimate the arrival of new requests based on historical data.
  - Local schedule: we consider additional information regarding the power consumption of individual applications and apply machine consolidation.

Algorithmic Model for Mid Range Planning

- We ignore all machines occupied by on-demand jobs.
- Each job j is allocated to one machine type i.
- We introduce time instances t_k with $t_0=0$ and $t_k > t_{k-1}$ whenever
  - the power estimation changes,
  - the (expected) number of a machine type for on-demand jobs changes,
  - a new batch job is (expected to be) submitted,
  - a deadline of a batch job is reached.
- We consider interval $[t_{k-1},t_k)$ with $\Delta_k=t_k-t_{k-1}$.
  - There is a constant number of machines $m_{k,i}$ for each type i and a constant target amount of power P_k within interval k.
  - A deviation $\mu_{k,i}$ of machines of type i results in a deviation $\delta_{k,i} \cdot \Delta_k = h_{k,i} \cdot \mu_{k,i}$ in energy.

Linear Program

$\min\{\max_k\{\sum_i \delta_{k,i}\}\}$	such that
$\sum_k p_{j,k} = p_j$	for all jobs j
$p_{j,k} \le \Delta_k$	for all jobs j and intervals k
$(P_k - \sum_i \delta_{k,i}) \cdot \Delta_k \le \sum_j h_{k,i} \cdot p_{j,k}$	for all intervals k
$\sum_j h_{k,i} \cdot p_{j,k} \le (P_k + \sum_i \delta_{k,i}) \cdot \Delta_k$	for all intervals k
$p_{j,k} \ge 0$	for all jobs j and intervals k
$\delta_{k,i} \ge 0$	for all intervals k and machine types i
$p_{j,k} = 0$	if interval k is not part of interval $[r\_j,d\_j)$ for job j

Algorithmic Solution for Mid Range Planning

- The solution of the linear program minimizes the maximum deviation of power over all intervals.
- There is at most one machine of each type that is only fractionally used in each interval.
- We can generate a preemptive schedule using, for instance, McNaughton's method for each interval.
- We can apply a variety of objective functions
  - We can force a smaller deviation in the first intervals by using weights.
- We can use different factors for negative and positive deviation from the target power in an interval.

Consequence of Mid Range Planning Results

- No deviation of power over all intervals
  - No change of the strategy is required.
- Repeated or significant deviation of power over all intervals
  - Determination of the cause and modification of the model
- Overestimation of total energy demand
  - Incentives to users to submit jobs with short deadlines
- Underestimation of total energy demand
  - Increasing the deadlines for some batch jobs
- Significant imbalance of the energy demand
  - Adjusting release times and deadlines

- Advantage of mid range planning: More time to develop strategies

Local Scheduling

- The local scheduling algorithm considers the current interval (or the first few intervals).
- Even if we maintain the average target power consumption for the interval there may be fluctuation within the interval.
  - Determining a schedule that observes the target power at any moment is NP-hard (Partition problem).
  - If we consider the energy consumption within a very small interval we can achieve our goal by possibly using a very large number of preemptions.
- We may be able to reduce power consumption by considering the allocation of processors to racks.
  - Such consolidation may require a substantial amount of job migration.
  - An upgrade of a machine allocation may result in saving power.

Hardware Support for Local Scheduling

- Local generator to handle brief power peaks
  - Local generators are available in some installations to provide enough time for shutting down the systems in case of power failure.
- Batteries to buffer excess energy and release it later during power peaks
  - We can use similar batteries that are part of UIP (Uninterruptible Power Supply) in many installations.
  - UIP batteries themselves cannot be used for safety reasons.

Additional Goals of Local Scheduling

- Minimize the amount of preemption and migration
  - Preemption and migration lead to additional communication and therefore additional power consumption.
  - Although the preemption and migration penalty is small a large number of preemption and migrations may have a significant performance impact.
- Avoid significant sudden changes in power consumption
  - Sudden power changes put a huge strain on some mechanical cooling systems.
  - Sudden power surges may have negative electrical effects.

Conclusion

- Is there a quantitative benefit for an energy efficient algorithm that does not also reduce processing time?
  - Yes, if you are the system owner and pay for the energy bill directly.
  - No, if you run the application on a system that you do not own.
- Can system owners improve energy efficiency of their systems?
  - Partly by optimizing general properties, like housing and cooling.
  - No with respect to the application without any additional information
- What can be done in the future?
  - Define an allocation based on the power consumption instead of or in addition to the usage time.
  - Develop techniques to measure power consumption of an application.
  - Develop techniques to reliably predict power consumption of an application