Lessons learned from procuring a fairly large HPC system

Thor Wikfeldt (ENCCS, Uppsala University) and Gert Svensson (PDC, KTH)

These slides: https://hackmd.io/@KTW/HPC-procurement-lessons

Background

- The PDC Center for HPC, KTH is one of two major HPC centers in Sweden.
- In 2019 the Swedish Research Council, via SNIC, approved to fund a replacement system for Beskow

Background

- Should replace Beskow and Tegner at PDC, Aurora at Lunarc and Hebbe at C3SE
- General-purpose cluster for all researchers and PDC industrial partners in Sweden
- For CPU-workloads as well as GPU-workloads

Procurements 101

Different procurement alternatives:

- Full-fledged procurement
  - Only mandatory requirements - select the lowest price
  - Some kind of evaluation(bid) - fixed price
  - Some kind of evaluation(bid, price)
  - Involves setting scores or “costs” on the requirements

Requirements and evaluation

- Mandatory requirements (something **must** be fulfilled)
- Optional requirements (something **should** be fulfilled)
    - Only meaningful in combination with evaluation

Public procurement alternatives

One stage Open procurement

- Publish all requirements, conditions and evaluation
  - Reply to possible questions (can include small changes)
- Get bids
- Evaluate bids according to the requirements
  - Possible to send questions to the vendors
- Select the bid with the highest score

Negotiated Procedure with Prior Publication

- Invite vendors with general description (publication)
- Select possible vendors, and send to selected vendors all requirements, conditions and evaluation
    - Reply to potential questions from the vendors (can include small changes)
    - Receive bids
        - Evaluate bids according to the requirements
        - Possible to negotiate with the vendors to improve the bids.
        - Requirements can’t be changed
        - Evaluation and negotiating can be repeated many times
    - Ask for best and final offers
    - Select the bid with the highest score

Competitive Dialogue

- Requirements can’t be fulfilled with standard solutions.
- Some innovation is involved.
- The end-result is known but hard or too complex to write requirements.

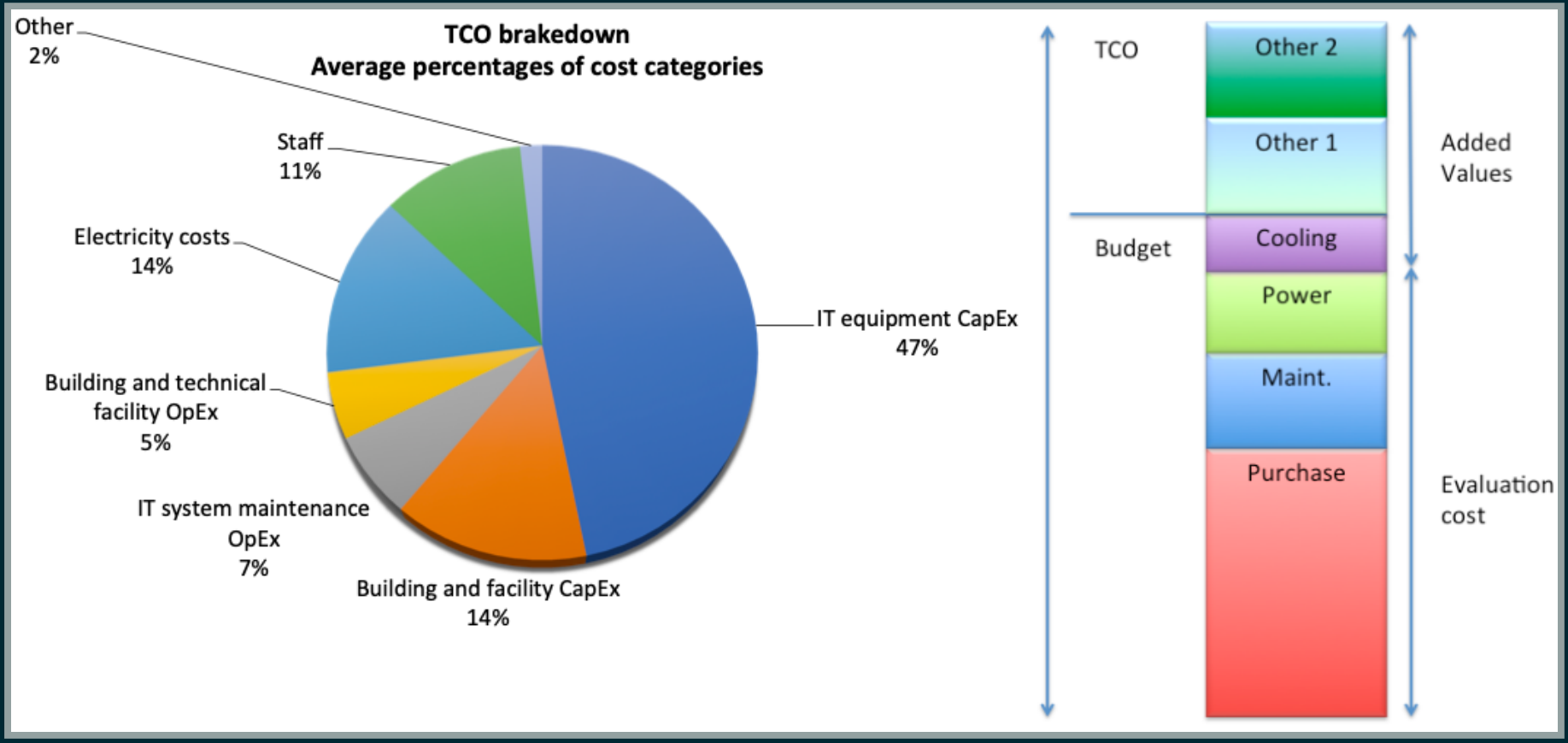
Lessons learned

- PDC chose **negotiated procedure**.
- Rather new form of procurement.
    - less experience and court verdicts.
- Vendors know they have a second chance.
    - This leads to lower quality of the bids.
    - You don’t get the final price from the start.
- Negotiating takes some time.
- Advantage: you can point out weak areas of a bid.

Total cost of ownership (TCO) approach

- The cost of running an HPC system consists of many components:
  - Purchase price (HW + SW)
  - Installation
  - Maintenance and service
  - Power and cooling cost
  - Costs to operate the system
  - Costs for the building
- Depending on how the funding is set up, different parts of the TCO can be evaluated.
- For PDC we included all costs above EXCEPT operational costs and building.
- Give us the system with the highest score for a fixed TCO.

Total cost of ownership (TCO) approach



Lesson learned

- We estimated power usage by benchmarks
- One vendor used power capping

Different sections of a procurement

- Qualifications
  - Economical and technical
    - For example minimum turnover and minimum position on the TOP500 list
- Technical requirements
  - Must and should requirements
- Commercial conditions
  - Delivery
  - Acceptance test
  - Integration in center environment
  - Final acceptance test
  - Maintenance
- Draft contract

Lesson learned

- TOP500 requirement useful to eliminate small companies.
- Integration in PDC environment
  - Two weeks for vendor and PDC staff together to integrate into PDC software environment (we will see how it works out).
- Final acceptance test
  - One month with real users on the system
  - Measure availability
  - Not fulfilled -> Redo the test

Lesson learned

- The difficulty to ask for what you really want
- We want things that are difficult to measure and specify:
- Delivery in time
- Fast maintenance
- Help if problems occur
- Software that is easy to use.

Lesson learned

- Contract
  - Must fulfil all requirements during system’s maintenance period
  - Must fulfil the availability during system’s maintenance period
  - All software should be updated to the latest version during the maintenance period of the system
    - All future versions of Fortran (for example)

Benchmarking

Purpose of benchmarks

- Measure performance of proposed systems for *future workloads*
- Test all performance aspects in a balanced way
    - size of machine, FP performance, memory bandwidth, interconnect, I/O, software environment, ...
- Eliminate low quality bids early
- Reduce risk
- Enable a scoring system to discriminate between bids
- Evaluate support capabilities post-delivery

Benchmarking

Lessons learned

- Vendors (even large ones) don’t have massive test resources for benchmarking
    - Some use only own systems, others rely on third-party vendors for e.g. benchmarks on accelerators
    - They will extrapolate performance to production machine, but quality and risk-taking can vary a lot between vendors
- It’s difficult to find a good mix of benchmarks that corresponds to intended future use - not too many but not too few
- You need to avoid inadvertently eliminating certain hardware
- Ease of porting or tuning applications should be taken into account

Application benchmarks

- Should cover relevant scientific areas
    - Input from stakeholders, e.g. reference group with representatives from major user communities
- Cover different algorithms, programming languages and libraries. Include compute, memory, network and IO-bound codes
- Representative subset of applications
    - Past usage as a guide: SLURM logs, granted allocations
    - But take into account future HPC directions
    - Codes which can run or be ported to relevant architectures
    - Cover also representative *job types*

Application benchmarks

PDC procurement, phase 1

- Gromacs (MD): one strong-scaling and one throughput case
- Nek5000 (CFD): strong-scaling case
- VASP (DFT): throughput case
- PowerFLOW (CFD): throughput case

Phase 2

- Gromacs (MD): one strong-scaling and one throughput case
- PyFR (CFD): single- and double-precision strong scaling cases
- CP2K-Quickstep (DFT): throughput case

Application benchmarks

Lessons learned

- Avoid politics in benchmark selection when engaging with stakeholders
- Consider user survey and/or user interviews
- Monitor and collect data on you current HPC system
- Strike a balance between past and emerging workloads
- Make sure to have benchmarks that test all hardware and software features you desire
- To test vendors' ability and willingness to collaborate, allow reasonable code modifications

Application benchmarks

Lessons learned

- Expert advice is critical. Involve people with expertise in chosen HPC applications
  - They can design good benchmark cases and document them
  - They can understand reasons for surprising results from vendors and scrutinize benchmark reports
- Keep it simple and clear
  - Complicated benchmark setup or documentation can be misunderstood by vendors
- Include strict requirements on (or score) the quality of detailed benchmark reports. Poor documentation makes reviewing difficult
- AI benchmarks (https://mlperf.org/) are currently immature, but will be important to include in future procurements

Synthetic benchmarks

- Synthetic (standard) benchmarks test a variety of different performance metrics
- Can be included in scoring or as should/must requirements to ensure that system meets required standards

Challenges

- Using only one or a few synthetic benchmarks will bias the machine (e.g. SPEC performance optimized by high core-count processors, limiting to HPL will lead to same-sized proposed machines)
- Not representative of real workloads. System designers known to optimize designs based on synthetic benchmarks

Synthetic benchmarks

PDC procurement

- HPL: Measures attainable FP performance by solving a large, dense system of linear equations
- STREAM: Measure attainable sustained read and write bandwidth of the primary memory of a single node
- FFT: Measures performance of computing large fast Fourier transformations for double-precision data
- GPCNET: Measures performance of the fast interconnection network using MPI under the presence of congestion
- IOR: Measures the sustained read and write performance achieved by concurrent access from multiple clients to the parallel file-system.
- mdtest: Measures achievable performance of meta-data only operations from multiple clients on the parallel file-system.

Synthetic benchmarks

Lessons learned

- Expert advice is critical
- Vendors mostly know how to do synthetic benchmarks
- However, need to make instructions clear:
  - some cases have degrees of freedom: how many nodes and cores per node should be used?
  - code modifications allowed?
- Use synthetic benchmarks as should- and must-requirements rather than including them in scoring system

Evaluation metric

- The metric that you tell vendors to optimize will determine what characteristics vendors will focus on
- Possible metrics: run duration, scalability, performance/power, performance/cost, ...

PDC procurement

- *Throughput type*: Number of jobs completed in one day on full machine. Tradeoff between number of nodes offered and the performance of each node
- *Strong scaling type*: Number of jobs of given application that can be completed in one day on full machine with *only one job running at a time*. Tests scalability and network

Evaluation metric

Lessons learned

- Many HPC metrics have been developed
  - Discussed at every Supercomputing conference
  - Worth to spend time on researching this
  - See references below
- Constraints on reported performance, variability of runtimes, etc. should be taken into account
- Consider higher-level system issues, e.g. how effectively a system can schedule and manage varied workloads, how rapidly a system can launch jobs, how quickly it can recover from system outages.

Scoring

PDC procurement

- Assign Max Points to each application benchmark case based on past usage.
- Phase 2 benchmark scores scaled by number of days in production.

Algorithm:

1. Winner in each case receives max points, others receive points *relative* to the winner
2. Sum up to Total benchmark score
3. Eliminate bid with lowest score
4. Repeat from 1 until two bids remain

Scoring

Lessons learned

- Relative scores have pros and cons
  - No need to design absolute performance scales for each benchmark, which could become imbalanced
  - Simple version can be rigged: vendors A and B have competitive bids. Devious vendor C submits bid tuned to far outperform both A and B on one benchmark in order to favor A or B
  - Vendors can not compute their own final score
- In aggregating benchmark scores, the method of averaging (arithmetic, harmonic or geometric mean) can matter.
- Other metrics make scoring simpler, e.g. job-mix models

References

- William T.C. Kramer, John M. Shalf, Erich Strohmaier, The NERSC Sustained System Performance (SSP) Metric
- Richard Blake, Francois Robin, Marco Sbrighi. “PRACE HPC Systems Procurement Best Practice”
- SC17 BoF session Total Cost of Ownership and HPC System Procurement

Discussion