Math Library Challenges in Auto-Tuning

Greg Henry
Intel® MKL Architect

My Perspective on Automation

- We're a consumer of numerical auto-tuning and automation
- Automation is critical to math library success
- When optimizing, the closer you are to peak performance, the more everything has to be perfect or you'll lose performance.
- We hand-tune + auto-search accordingly

Dispatching is an important feature

- Run optimal code on any HW automatically.
- We use internal dispatching to accomplish this.
- We want to be competitive everywhere we run (not just on Intel HW).

BLAS as an example of automation

- All our BLAS routines are based upon an internal API.
  - Lower level library routines than the BLAS themselves
- Our developers can use this API to get even more performance control
- Internal API routines are both auto-tuned and hand-tuned
- Examples to mention:
  - Dynamic run-time examples
  - Distributed memory searches

Our Auto-searching tools

- Obvious targets: find the optimal block-size or loop ordering
- Non-obvious things: finding the best assembly instruction streams
- Search spaces can be large
  - We've techniques for trying more cases per second than other approaches

Automation hurdles aren't just about performance

- Finding the best library isn't our biggest hurdle
- Identifying and dealing with regressions while making code changes can involve far more data parsing.

Conclusion

- Auto-tuning is critical.
- Please send me ideas/feedback at greg.henry@intel.com