

Ekaterina Antakova

Senior Software Engineer, Intel

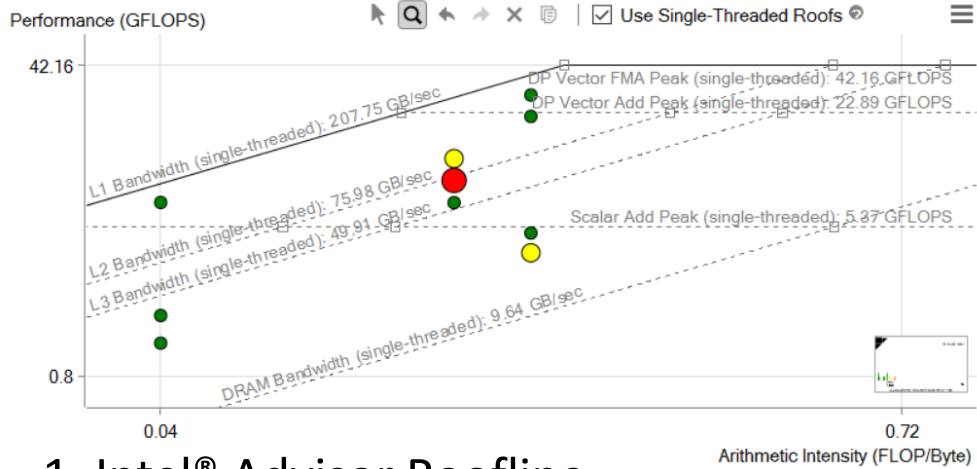
kate@antakova.ru

<https://www.linkedin.com/in/ekaterina-antakova/>

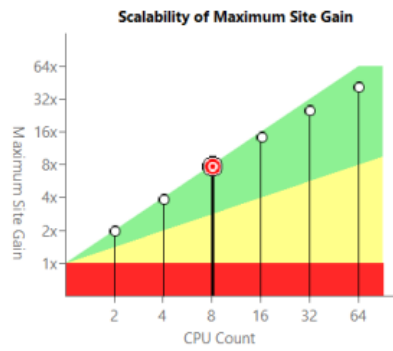
Key Skills

- Advanced C++ (Boost, wxWidgets, RapidJSON)
- Cross-platform GUI development
- Intermediate Python (pytest, pandas, scikit-learn, Jupyter, ipywidgets)
- Git, SVN, Jira, XML, XSLT
- Experience with HTML, CSS, JavaScript, Vue.js
- Performance analysis and optimization
- Correctness analysis
- Stakeholder, requirements and release management, agile methodologies

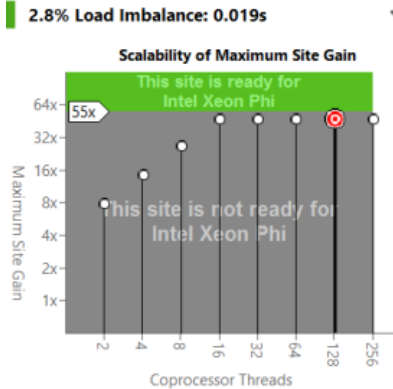
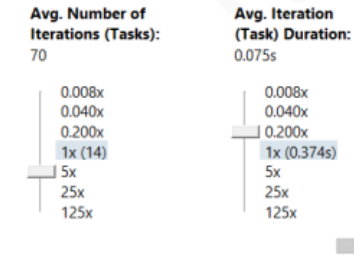
Projects



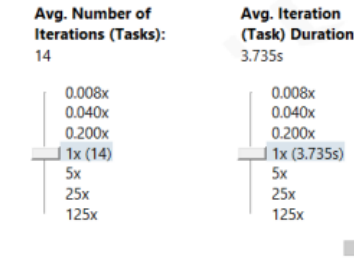
1. Intel® Advisor Roofline



Loop Iterations (Tasks) Modeling



Loop Iterations (Tasks) Modeling



3. Modeling of parallel performance on CPU and accelerator

Function Call Sites and Loops	Performance Issues	CPU Time	Type	Why No Vectorization?	Vectorized Loops
		Total Time	Self Time		Vector... Efficiency Gain E... VL (Ve...
[loop in col_f_e_and_d_m_SompS]	1 Potential under...	22.290s	22.290s	Vectorized (Body; P...	SSE2 -84% 1.67x 2
[loop in col_f_e_and_d_s_SompS]	1 Potential under...	10.871s	10.871s	Vectorized (Body; P...	SSE2 -91% 1.81x 2
[loop in col_f_angle_avg_m_Somp]	2 Unoptimized fl...	6.540s	6.540s	Vectorized (Body; P...	SSE2 -92% 1.83x 2
[loop in col_f_angle_avg_m_Somp]	1 Unoptimized fl...	4.309s	4.309s	Vectorized (Body; ...	SSE -83% 1.66x 2
[loop in ellip_agm_v at elliptics.F9]	1 Unoptimized fl...	4.100s	4.100s	Vectorized (Body)	SSE2 -100% 2.43x 2
PetscCheckPointer		1.640s	1.600s	Function	
Bicub<Kokkos:Device<Kokkos:Op		1.450s	1.450s	Inlined Function	

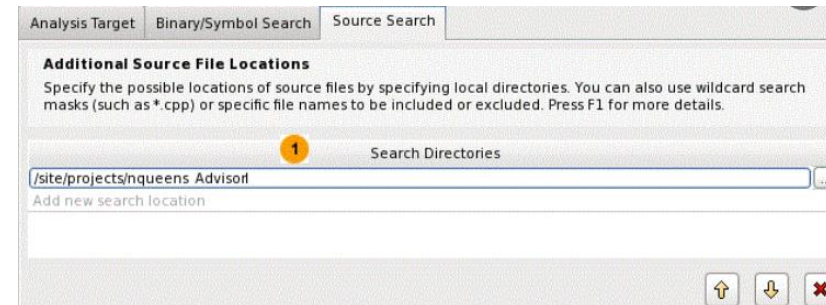
All Compiler Diagnostics

Loop vectorization possible but seems inefficient

Cause: The compiler vectorizer determined the loop will not benefit from vectorization. Common reasons include:

- Non-unit stride memory access
- Indirect memory access
- Low iteration count

2. Intel® Advisor top loops combined with Compiler vectorization report



4. Custom GUI controls and dialogs








Publications, Presentations



Have you ever threaded an application but seen little performance gain? Have you hit a "scalability ceiling" where performance gains level off as you add more cores? Implementing a parallel algorithm can be a lot of effort. Wouldn't it be great to explore a couple of different implementation schemes and see which is best, before investing in the heavy lifting of full

[How to Design for Scalable Performance.](#)

Intel(R) Parallel Universe Magazine, 2015

- ▶  1:33:43 Мастер-класс "Основы программирования на Python"
Ekaterina Antakova
- 2  1:05:17 Мастер-класс по контейнерным типам данных ...
Ekaterina Antakova
- 3  1:04:43 Мастер-класс: отладка и написание надёжного кода -...
Ekaterina Antakova
- 4  1:33:15 Никита Ладоскин - Модули в Python, OS, регулярные...
Ekaterina Antakova
- 5  41:53 Екатерина Антакова. Подготовка данных для...

[Python for Data Analysis Workshops](#)

for Women in Big Data Community, 2018

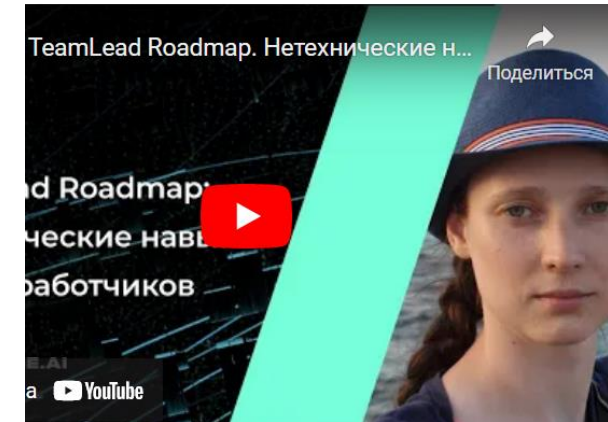


18 The Roofline Model

Intel's **Cedric Andreolli**, **Jim Cownie** and **Kate Antakova** highlight the importance of Roofline optimisation for HPC application development

[Understanding and Improving the Performance of Bandwidth Bound Code.](#)

Scientific Computing World – HPC 2018-2019



[Semi-Technical Skills for Developers. Overview of TeamLead Roadmap.](#)

Xperience.AI seminar, 2021