Instructions:This form is to be completed by the manager and candidate for Technical Leader position.

**Tip**: Verify word count by highlighting the text. Word count will appear in the tool bar at the lower left of your screen in Microsoft Word.

|  |  |
| --- | --- |
| Candidate Name and WWID | Jatin Bhateja, 11846689 |
| Business Group/ Sub-Group | DCAI/DCAI Software/DSE/DLJ |
| Current Job Title | Cloud Software Development Engineer |
| Manager Name and WWID | Sandhya Viswanathan, 11267150 |

Additional Guidance on Sections [0][1][2][3] (1700-word limit combined). Highlight the following:

* Internal/external code – provide links when possible.
* Architectural or strategic documents the candidate (co-)authored.
* Industry recognition.

|  |  |
| --- | --- |
| 0 | Business justification and forward-looking expectations for Individual   * Why is this role business critical at this time? How does this role give Intel a competitive advantage? * The Technical Lead title is a recognition of the candidate’s developing technical leadership skills and their future potential. Focusing on the next 24 months, what are their development expectations and how is their role and expected technical leadership and business contributions expected to change? |
| * Java is one of top languages for cloud and data center workloads. Java JIT compiler is a critical component of the state-of-the-art open-source OpenJDK Java runtime. In upcoming Intel Architecture (IA) server platforms (DMR, DMR-HD, NVL Entry), we are introducing significant improvements to ISA with Intel APX and Intel AVX 10.x. JIT compiler optimizations/enabling to use these latest Intel ISA would help to run existing/new workloads best on Intel platforms and give us competitive advantage. ARM and RISC-V companies are investing in OpenJDK and have become very active in recent years optimizing Java runtime for their platforms. Jatin Bhateja with 20+ years of Compiler expertise and coveted “Reviewer” status in OpenJDK community is the perfect candidate to be the Java JIT Compiler Technical Lead and drive IA optimizations in the Java community guiding other engineers in the team. Jatin is also crucial to lead and drive to completion our new language feature initiatives in the OpenJDK around Float16 and Java Vector API working closely with Oracle and ARM. In addition, Jatin has also started working with Oracle JVM architect in exploring/analyzing CPU feature requirements for upcoming Java features with the goal of Java running best on IA. | |
| 1 | Expertise  Describe the candidate’s skill and expertise in their domain, including specific examples.   * What is the candidate’s area of expertise, industry influence, and sustained body of work? * How does this expertise contribute to Intel’s strategy and success? * How has the candidate demonstrated depth and/or breadth of expertise and how has that impacted Intel and/or the industry? |
| * Jatin Bhateja is working in Java platform optimization team since his joining in December 2018. Jatin has more than 20 years of rich hands-on experience in compilers and EDA front-end tool development. Has a proven track record of contributions to leading open-source compiler[s] and runtime framework and has been instrumental in establishing Intel as a key participant in OpenJDK and Java community, He is currently focusing on adding low precision type support in Java making it future ready for next gen AI workloads and Vector Databases. He is also spearheading Diamond Rapids software enabling efforts in JVM, closing the competitive optimization gaps against ARM servers, and Vector API – Valhalla integration efforts working in close collaboration with Oracle teams.   Following is gist of his year wise significant contributions to OpenJDK since joining Intel:   * **2019**: C2 compile Generic Operands Support * Significant infrastructure changes in instruction selection enabling merging of lots of instruction selection patterns. This resolved one of the major bottlenecks in Vector API 1st Incubation acceptance into mainline OpenJDK. This solution was later adopted by ARM to unify their Neon and SVE backends. * **2020**: AVX512 differentiating optimizations * Designed and implemented an advanced Ternary Logic optimization which folds logic cones into one 3 input LUT to generate ternary logic instructions introduced by AVX512 ISA. * Implemented partial inlining of for small copy/mismatch methods using AVX512 showing 2-3x speedup. * Optimized stubs for copy, fill, and mismatch methods routinely used in almost all Java workloads. * Optimizations of various math operations like ceil, floor, round, floating point min/max, signum, float to integral conversion. * **2021**: Implementation of Java Enhancement Proposal (JEP) 417: Vector API (Third Incubator)   Predicated operations form the backbone of data parallel programming and help in translating conditional logic into straight line data flow. Vector API third incubator added AVX512 optimizations for predicated/masked operations enabling users to write efficient data parallel code in Java.   * Jatin was leading contributor for predicated vector operations support resulting into around 2x performance improvement on AVX512 targets. He extended register allocator to handle opmask registers and added support for masked AVX512 instruction code generation. He also improved the performance of routinely used masking operations, firstTrue, lastTrue, anyTrue, allTrue. * **2022**: JEP 426: Vector API (Fourth Incubator)   Jatin spearheaded the design and development of the JEP in collaboration with OpenJDK community partners (Oracle/ ARM).   * Optimized implementation of various bit level operations, BIT\_COUNT, LEADING\_ZERO\_COUNT, TRAILING\_ZERO\_COUNT, REVERSE, REVERSE\_BYTES. * Added powerful vector compress / expand APIs leveraging AVX512 instructions. These APIs target optimizing columnar data-base predicate pushdown and was used by OpenSourceDB team to optimize ORC reader in prestoDB open-source showing 1.5x gains. * **2023**:   Valhalla – Vector API Integration  OpenJDK project Valhalla led by Oracle introduces value types in Java and is slated to be integrated in mainline OpenJDK in one of the upcoming releases. Java Vector API vectors are immutable quantities and so should be implemented as value types. This is a key requirement for Vector API incubation exit.   * Jatin is leading the Vector API incubation exit and introduction as a preview/product feature in the OpenJDK community. He is the only contributor on this effort from Intel. He came up to speed on Valhalla specification and its complex implementation. He then designed and implemented a prototype of Java Vector API over Valhalla working closely with ARM engineers. As part of this he identified important gaps in Valhalla design/implementation and shared his findings with Oracle JVM architects.   Intel CESG PRC team collaboration   * Delivered optimized parquet unpacking algorithm which was up streamed into parquet-mr mainline and received several testimonials from customer Alibaba. * Collaborated for 1.5 years with PRC team on Vector API backporting efforts to Alibaba JDK Dragonwell 11. * **2024**: * APX support for DMR: Jatin kicked off OpenJDK DMR SW enabling effort with APX EGPR register allocation support, extended GPR state save-restoration using efficient PUSHP2/POPP2 instruction, and new setZUCC ISA support. He also reviewed other team members work on APX encoding and code generation support in the Java runtime. Jatin setup end-to-end validation flow using SDE for OpenJDK. He discovered, root-caused, and reported multiple blocker issues to SDE team. * Float 16 support in Java: Jatin took initiative and collaborated with Oracle/ARM to add Float16 type support as an incubating feature in JDK 24. He then followed it up with optimized scalar float16 operations using Intel FP16 ISA which is integrated into OpenJDK towards JDK 25. Auto-vectorization support is in progress. * **2025** * Jatin is working on identifying and closing the performance gaps with ARM AARCH64 servers (Graviton). * Jatin has started a feasibility study on adopting MicroTSX for large atomic field updates in Valhalla.   Jatin is the only Intel OpenJDK development team member outside US and has made significant contributions to enable intel differentiating features in OpenJDK. | |
| 2 | Leadership  Describe how the candidate develops and builds technical talent:   * REQUIRED: How is the candidate developing technical talent and building the technical leader community in their domain? Provide specific examples, quotes and results. * How does the candidate foster a culture of inclusion, innovation, and empowerment of others’ success? * What are the candidate’s biggest opportunities for growth as a leader? How is the candidate demonstrating self-awareness and a desire to improve? |
| Jatin mentored Fang Xie from CESG PRC team in development of data parallel algorithms using Java Vector API. He guided Fang throughout the implementation and gave suggestions to improve the bit-packing algorithm resulting in the work being accepted and integrated into Apache Parquet. Fang Xie mentioned “Big thanks to Jatin for your hard working on this project and make this happened and provided guidance on how to use/optimize Vector API. This is a very amazing collaboration between our two teams. Very thanks for your team’s support! We will continue pushing bitpack landing in our customers (Alibaba/Tencent/Baidu or other) and actively promote Parquet/Spark/Presto communities for more optimization.”  Jatin mentored Zhonghui Jin from CESG PRC team in backporting Java Vector API from JDK 17 to JDK 11 for Alibaba, a very complex task. Jatin shared his expertise, guided, and helped in root causing blocking complex issues. Joshua Zhu from Alibaba stated “Zhonghui, Jatin, Yunzhang, and team, I would like to thank all Intel members involved in VectorAPI backporting on Alibaba Dragonwell. I appreciate your hard work and close collaboration over the past few months. This week our milestone: the 1st incubator passed verification and has been integrated into the master branch of Dragonwell 11. In the meantime, we also finished the publicity together. VectorAPI utilizes SIMD units of Intel AVX and could help improve the performance of enormous quantities of Java applications in Alibaba.”  Jatin is very approachable and shares his knowledge with other team members without any hesitation. Jatin has guided several team members during their ramp up period e.g. he spent significant time in guiding and resolving queries from OpenSourceDB team specially Abhijit Kulkarni, on JVector and Presto ORC connector predicate push downing optimizations using Vector API. After we lost many engineers due to CPM actions recently, Jatin initiated a training on the Java JIT IR to help the remaining team members come up to speed on compilation technology. Jatin also actively collaborates with the Java Workloads and Benchmarks team members and is a go to person for Java JIT related questions. He has a very good standing and working relations with OpenJDK community members from other companies which will help him guide the team members to drive Intel optimizations and features into OpenJDK. | |

|  |  |  |
| --- | --- | --- |
| 3 | Impact  Describe the candidate’s performance, impact, and results. Include two or more recent examples and describe the impact to Intel and/or the industry.   * How has the candidate formulated strategies to deliver leadership solutions for our internal and external customers? * How is the candidate focused on solving Intel’s most important and difficult technical problems? * How has the candidate demonstrated a track record of relentless execution in bringing ideas to market? | |
| Jatin is the leading contributor from Intel and holds the Reviewer status in the OpenJDK mainline project.  Jatin’s work on the Java Vector API and his efforts to drive its early adoption in Apache Parquet helped Intel drive Vector API based SIMD optimizations into other important Java frameworks like simdjson-java and JavaFastPFOR. Even though the Vector API is still in incubation, this got attention of other key Java framework developers, and they started adopting it. It is now used in frameworks like JVector, Jlama, and Lucene. JVector is a Java embedded vector search engine, used by DataStax Astra DB and Apache Cassandra. Jlama is the Java implementation of a LLM inference engine which utilizes the Java Vector API for faster inference. Lucene is available in products like ElasticSearch. [Elasticsearch 8.9.0](https://www.elastic.co/blog/accelerating-vector-search-simd-instructions) was the first release which enabled this faster implementation by default out of the box showing significant performance improvements in SO Vector benchmark: indexing-throughput improved by 30%, merge-time decreased by 40%, and script-score-query-java-latency improved by 40%. Jatin is now leading the Vector API incubation exit in the OpenJDK community and its introduction as a preview/product feature which will bring data parallel programming to other mainstream Java frameworks and developers.  Jatin is collaborating with the OpenJDK community in bringing Float16 datatype along with optimized Float16 code generation to Java and then follow it up with adding Float16 vectors to the Java Vector API. This will provide further opportunities to optimize Java workloads on Intel platforms as Float16 will help in performing double the vector operations per instruction vs the 32-bit float.  Jatin is leading and guiding the engineers in the team to deliver APX and AVX10.x enabling in OpenJDK with a goal to run existing/new workloads best on Intel platform. | | |
| Additional Guidance on Section [4]:  Provide 2 letters of recommendation. References should be from Grade 10+ level employees. At least one reference should be from a technologist (PE+ or equivalent external role)**. No more than one page (a single paragraph is acceptable).** References should be familiar with the candidate’s work and expertise, cite specific accomplishments, and include examples of the candidate’s impact, leadership, and/or mentorship. | | |
| 4 | **Letters of Recommendation**  Please provide letters of recommendation following the requirements outlined below. References should be familiar with the candidate’s work and expertise, cite specific accomplishments, and include examples of the candidate’s contributions.  Include contact information: name, title, organization, email, phone number, and location.  Copy and paste each recommendation below. Do not embed documents.  Without prior approval for hardship reasons, references may not be provided by members of ELT |
| Please list name of recommenders below   |  |  |  | | --- | --- | --- | | Name | Title | Contact Information | | Vladimir Ivanov | Consulting Member of Technical Staff, JVM Compilers, Java Platform Group, Oracle | vladimir.x.ivanov@oracle.com | | Luca Giacchino | Principal Engineer, Java Platform Engineering, Intel | luca.giacchino@intel.com | | |
| Name Vladimir Ivanov  Title Consulting Member of Technical Staff  Organization  JVM Compilers, Java Platform Group, Oracle  Email vladimir.x.ivanov@oracle.com  I have been closely interacting with Jatin Bhateja (Intel) in OpenJDK project since 2019. Over the years he made numerous contributions to HotSpot JVM (primarily JIT-compilers) and Vector API (Project Panama) which required deep understanding of JVM architecture. As part of Project Panama, Jatin led multiple efforts which required close collaboration with other contributors (primarily, from Oracle and Arm). In particular, he led the project to converge Vector API implementation with Project Valhalla which required close collaboration with other teams. He demonstrated exceptional skills in communicating and negotiating high- and low-level design aspects of the project while successfully trouble-shooting numerous technical issues along the way. Since Jatin joined OpenJDK project his impact in the project grew steadily and now he is one of key members of OpenJDK community successfully representing Intel in there. I fully support his Tech Lead nomination.  Name Luca Giacchino  Title Principal Engineer  Organization  DCAI SW – DSE - DLJ  Email luca.giacchino@intel.com    I have worked with Jatin over the last 6 years on performance optimizations for a wide range of Java applications. Jatin has deep knowledge of Java internals, including compilation, execution, and performance characterization and tuning. Jatin contributed numerous features to the OpenJDK project, such as APX enabling in JIT compilation, AVX-512 implementations of mathematical functions, and a comprehensive Java Vector API. The Vector API is a major milestone for the entire Java ecosystem, allowing applications to leverage vector extensions directly in Java code. These features are critical to all Java optimization projects in our team, which include widely deployed applications/libraries such as Cassandra, Kafka, Presto, Lucene, OpenSearch. For example, his technical guidance allowed the team to deliver a vectorized implementation of data filtering in Presto (a popular open-source analytics database, used by companies such as Meta and Uber). His expertise is also essential in rapidly growing areas such as vector search, with the optimization of the JVector library used in Cassandra. Jatin’s expertise is constantly sought-after inside and outside the team, and he is always prepared to provide valuable technical guidance. This level of knowledge and contributions in a runtime used by hundreds of applications and customers is critical to deliver the value of Intel differentiating features across the industry. | |
|  | |
|  | |
| 5 | Response to Prior Feedback (repeat candidates only)  Month and year of prior nomination. Summary of prior feedback. Describe how the feedback has been or is being addressed, including specific results and data where possible. (350-word limit) |
| NA | |
| 6 | Supporting Technical Evidence (optional)  Use this section to provide any other significant information and evidence in support of the candidate’s nomination, such as recognitions, education, white papers, external publications, patents, involvement in industry societies and communities, etc. (300-word limit) |
| Following are links to Jatin’s open-source contributions from Intel towards various OpenJDK projects [1][2][3], he also collaborated with CESG PRC team to optimize parquet-mr reader using Java Vector API [4]  Prior to joining Intel, Jatin worked in open source for few years and made contributions to LLVM Compiler [5] and developed JVM backend from scratch [6].  [1] <https://github.com/search?q=repo%3Aopenjdk%2Fjdk+author%3Ajatin-bhateja&type=pullrequests&ref=advsearch>  [2] <https://github.com/search?q=repo%3Aopenjdk%2Fpanama-vector+author%3Ajatin-bhateja&type=pullrequests&ref=advsearch>  [3] <https://github.com/search?q=repo%3Aopenjdk%2Fvalhalla+author%3Ajatin-bhateja&type=pullrequests&ref=advsearch&p=3>  [4] <https://github.com/apache/parquet-java/pulls?q=is%3Apr+is%3Aclosed+VectorAPI>  [5] <https://github.com/search?q=repo%3Allvm%2Fllvm-project+author%3Ajbhateja&type=commits&ref=advsearch>  [6] <https://github.com/jbhateja/llvm_jvm/tree/master/lib/Target/JVM> | |