Broad Agency Announcement

Machine learning and Optimization-guided Compilers for Heterogeneous Architectures (MOCHA)

INFORMATION INNOVATION OFFICE

HR001124S0035

August 3, 2024

This publication constitutes a Broad Agency Announcement (BAA) as contemplated in Federal Acquisition Regulation (FAR) 6.102(d)(2) and 35.016 and 2 CFR § 200.203. Any resultant award negotiations will follow all pertinent law and regulation, and any negotiations and/or awards for procurement contracts will use procedures under FAR 15.4, Contract Pricing, as specified in the BAA.

OVERVIEW INFORMATION:

- **Federal Agency Name** – Defense Advanced Research Projects Agency (DARPA), Information Innovation Office
- **Funding Opportunity Title** –Machine learning and Optimization-guided Compilers for Heterogeneous Architectures (MOCHA)
- **Announcement Type –** Initial Announcement
- **Funding Opportunity Number –** HR001124S0035
- **Assistance Listing Number:** 12.910 Research and Technology Development
- **Dates/Time - All Times are Eastern Time Zone (ET)**
  - Posting Date: August 3, 2024
  - Proposers Day: August 5, 2024
  - Proposal Abstract Due Date: August 22, 2024, at 1:00 PM ET
  - Question Submittal Closed: September 12, 2024, at 1:00 PM ET
  - Proposal Due Date / BAA Closing Date: September 26, 2024 at 1:00 PM
- **Anticipated individual awards** - Multiple awards are anticipated.
- **Types of instruments that may be awarded** – Procurement contract, Other Transaction for Agreement, or Cooperative Agreement.
- **NAICS Code**: 541715
- **Agency contact**

  The BAA Coordinator for this effort may be reached at:
  MOCHA@darpa.mil
  DARPA/I2O
  ATTN:HR001124S0035
  675 North Randolph Street
  Arlington, VA 22203-2114

Section I: Funding Opportunity Description

The Defense Advanced Research Projects Agency (DARPA) is soliciting innovative proposals in the technical areas of compiler design, programming languages, and optimization. Proposed research should investigate innovative approaches that enable revolutionary advances in science, devices, or systems. Specifically excluded is research that primarily results in evolutionary improvements to the existing state of practice.

Introduction

For over three decades, hardware and software developers could rely on improved performance through increases in microprocessor clock speeds. This scaling broke down in the mid-2000s and attention shifted to new architectural features, such as multithreading. That's because today's technology, clock speed alone cannot lead to higher performance. Multithreading also has its limits and modern systems are increasingly using a variety of specialized co-processors and accelerators that are designed for high performance in specific domains and on specific tasks.

Traditional compilers are not designed to generate efficient machine code for such heterogenous ensembles of Central Processing Units (CPUs), Graphics Processing Units (GPUs), and other accelerators. Instead, software developers write unique code and libraries to take advantage of specialized hardware, reducing productivity and losing much of the potential benefit of these hardware components. Extending compilers to handle this heterogeneity is currently a manual task that is performed by compiler experts. Adapting current compilers is time consuming and error prone, does not address the challenge of taking advantage of hardware accelerators, and does not improve our ability to upgrade mission-critical systems in a timely manner.

MOCHA seeks to build a new generation of compiler technology that can realize the full potential performance of a system comprising multiple heterogenous computational elements. It will accomplish this goal by 1) using data driven methods, machine learning (ML), and advanced optimization techniques to rapidly adapt compilers to new hardware components with little human effort and 2) developing new internal representations and programming languages that enable compilers to determine how to make optimal use of available hardware, rather than depending on humans to do so. Without this capability, the Department of Defense (DoD) and the commercial world remain constrained by current compiler technologies and lack the ability to fully and rapidly capitalize on emerging specialized hardware.

Background

Traditional compilers have an “hourglass” architecture:

- At the top of the hourglass, a variety of programming language-specific front ends analyze the source code and transform it into a single intermediate representation (IR).
- This single IR acts as the “neck” of the hourglass, containing an ensemble of transformations that optimize the IR.
- Finally, there are a variety of hardware-specific backends guided by performance models of the target hardware and transform the optimized IR into machine code. These serve as the base of the hourglass.

This hourglass model has worked well for many years, but it is not well matched to the challenges presented by current and future heterogeneous computing environments (Figure 1).

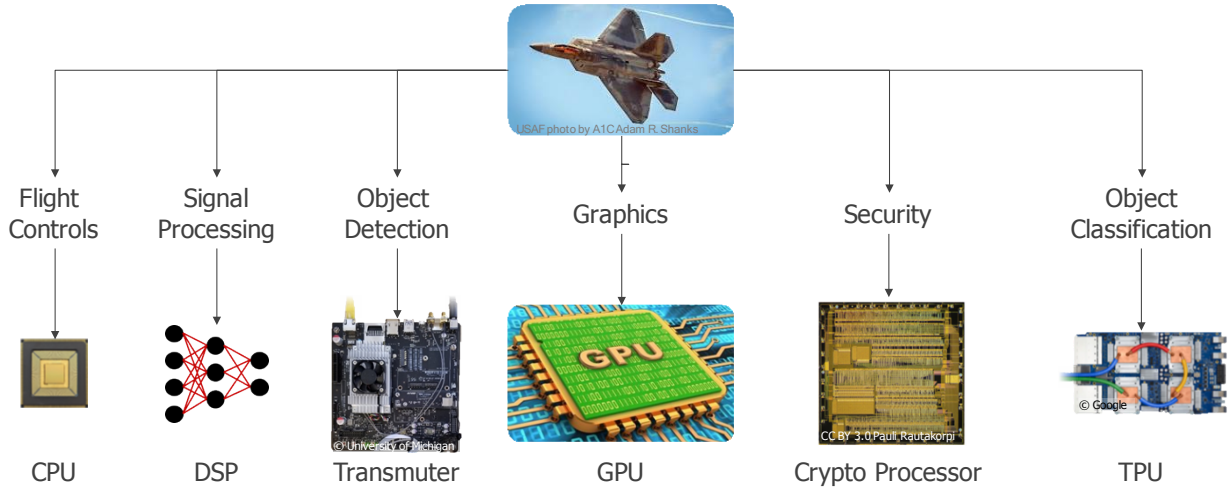
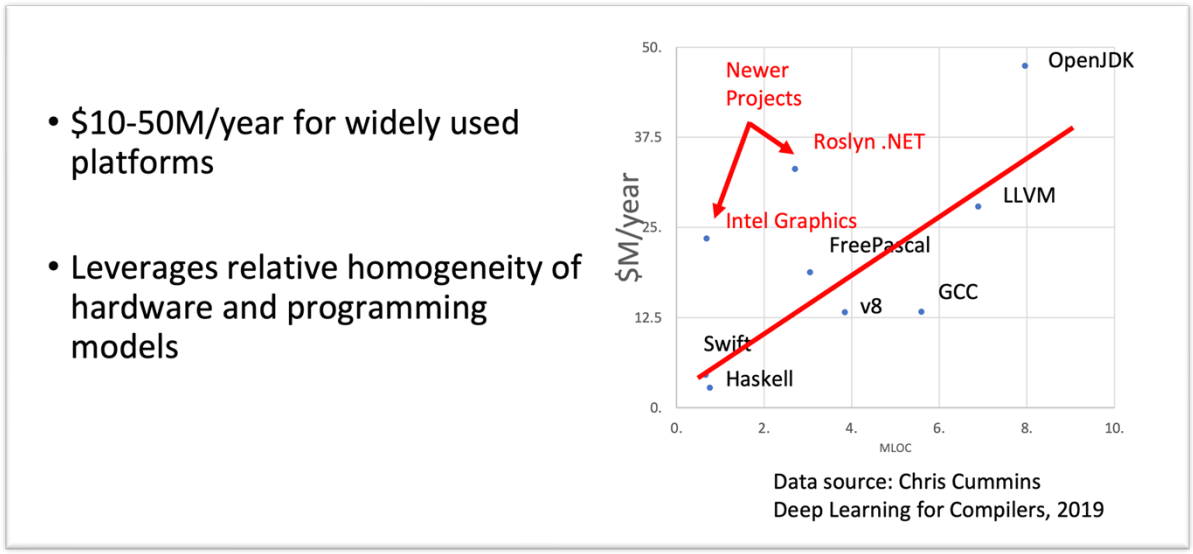


Figure 1 Tomorrow's compute environment

All components of today’s compilers are hand-crafted and hand-tuned. The cost of maintaining the compiler’s performance models and optimizations, in addition to the cost of adding support for a new hardware accelerator or source language, has been rising significantly over time. Meanwhile, the return on investment has been drastically declining (see Figure 2 and Figure 3).



[Credit: ISAT Heterogeneity Crisis]

Figure 2 Cost of compiler maintenance

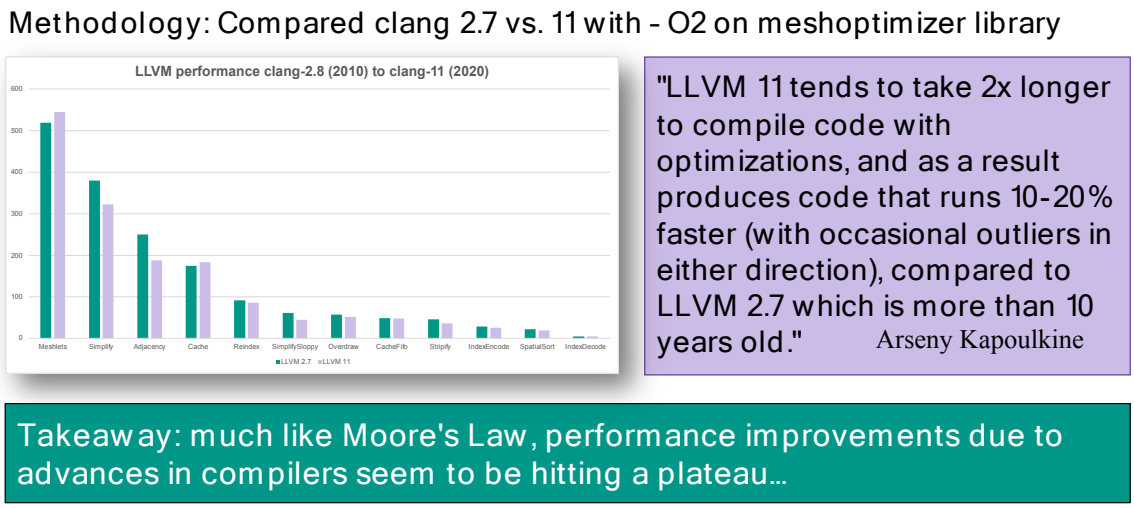


Figure 3 Benefit of compiler developer effort

The current approach to building compilers is also ill-suited to dealing with flexible and heterogenous target architectures. Today's alphabet soup environment of CPUs, GPUs, Tensor Processing Units (TPUs), Field Programmable Gate Arrays (FPGAs), Application-Specific Integrated Circuit (ASICs), and System-on-a-Chip (SoCs) will become even more complex as new packaging technologies, such as chiplets and 3D integration, enable rapid revisions of an architecture, replacing one hardware component with another and easily incorporating wholly new components.

Today's compilers will not work well for a program that is written in a hardware-agnostic manner, but targets a heterogeneous ensemble of CPUs, GPUs, accelerators, and other computational elements (CE). The single IRs of traditional compilers cannot adequately represent the characteristics of all the elements of the ensemble, making the optimization passes

ineffective. Furthermore, to perform optimizations, the compiler must have a model of the performance characteristics of each of the CEs, as well as the cost of switching computations between CEs. These performance models are expensive to build by hand and are not always accurate. In a future where static, monolithic hardware is supplanted by easily upgraded custom heterogeneous hardware, we will need to program at a higher, hardware-agnostic level and leave the determination of where to perform which computations up to the compiler.

Technical Area

The MOCHA program will have a single technical area: Compiler Technology. As discussed above and below, this technical area encompasses several distinct compiler components and proposals may address one or more individual components or the entire compiler toolchain.

One key insight motivating the MOCHA program is that each step of the compilation process is driven by performance models of the target CEs (and of the communication fabric). Building these models by hand is a bottleneck MOCHA intends to address by enabling rapid adaptation of compilers to novel CEs and to novel ensembles of existing components. It is a premise of the MOCHA program that these models can be generated using data-driven and ML techniques and that these techniques will not require costly (in human time) data collection and curation. Instead, the MOCHA approach would be to generate code and measure its performance on the target CE or to mine architectural documents for the relevant performance information.

The MOCHA program envisions updating the three-tier architecture found in current compilers to include:

- A front-end that incorporates hardware-agnostic domain-specific languages, determines how to partition the computation into modules, and decides which modules should be directed to which CE.
- A middle-end, potentially wider than the hourglass model, which will contain a variety of intermediate forms, each likely associated with its own optimization suite, that is targeted to a specific type of CE [5,10]. Intermediate forms and their associated optimization suites may traverse many levels of abstraction [5,10]. The choice of which optimizations to perform and in which order are again informed by performance models of the target CEs. These performance models will be learned using data-driven and ML techniques [2,3,4].
- A back end that will generate code sequences that are subject to multiple performance figures of merit (i.e., throughput, power consumption, memory footprint). Back-end performance models will also be learned from sources such as physical hardware, hardware simulations, and hardware specifications [1,9] and will incorporate both intrinsic CE performance, as well as costs to move data to/from/between CEs.
- Finally, advanced end-to-end optimization techniques and novel representations [7,8] will be needed to manage the huge search space of possible implementations of a source program.

The MOCHA program is soliciting proposals to address one or more of the technical challenges outlined above. A proposal may address one or more specific challenges (e.g., optimization selection, performance modeling for back-end code generation, partitioning and mapping of

source code) or may address the entire compiler toolchain. The MOCHA program assumes that a static collection of CEs available for code execution is specified at compile time. Although the potential value of runtime allocation of computations to CEs is recognized, such solutions are not sought under this announcement.

Proposals to assemble a complete compiler toolchain are strongly encouraged to include a plan to incorporate technology from other MOCHA program teams that are addressing a single or limited number of technical challenges.

All proposals should leverage or build on existing compiler frameworks such as LLVM[1] or the GNU Not Unix (GNU) compiler collection. Components that are not dependent on a larger compiler toolchain should nonetheless present a plan to accommodate test and evaluation within such an environment. Proposals building on proprietary compiler frameworks will be considered if the artifacts produced under the MOCHA program are independently evaluable and are delivered to the government.

A strong proposal will:

- Specify what aspect(s) of the compiler are being addressed and how the contributions are expected to aid in meeting program objectives (i.e., reduction of human effort and performance of compiled code).
- Show how the proposed contributions will interoperate with common compiler toolchains (e.g., LLVM).
- Identify the key technical risks of the proposed approach, describe how those risks will be tracked, and suggest possible mitigations.
- Include team members with a history of successful contributions to the compiler framework around which the proposed work will be centered.
- Incorporate a plan to assure the results are available and sustained after the program ends; for example, through integration into an open-source project developed during the MOCHA program.

In addition to the above, for proposals addressing the complete compiler toolchain, a strong proposal will also include a mechanism and plan for incorporating the work of other performers into the proposed compiler toolchain.

Program Structure

The MOCHA program is a 36-month effort, divided into annual segments with a preliminary and final performance assessment in each performance year (Figure 4). For pricing purposes, proposers should assume an April 2025 start date. For pricing travel, assume a kickoff meeting will be held in the Washington, DC area and Principal Investigator (PI) meetings will be held every 6-months, alternating between the east and west coasts of the United States.

[1] LLVM originally stood for *Low Level Virtual Machine*, however the project has expanded, and the name is no longer an initialism.

The government anticipates technologies generated and developed under MOCHA may require several years to mature. Each performance year, performers should plan to develop technologies that are technological prerequisites for the succeeding year.

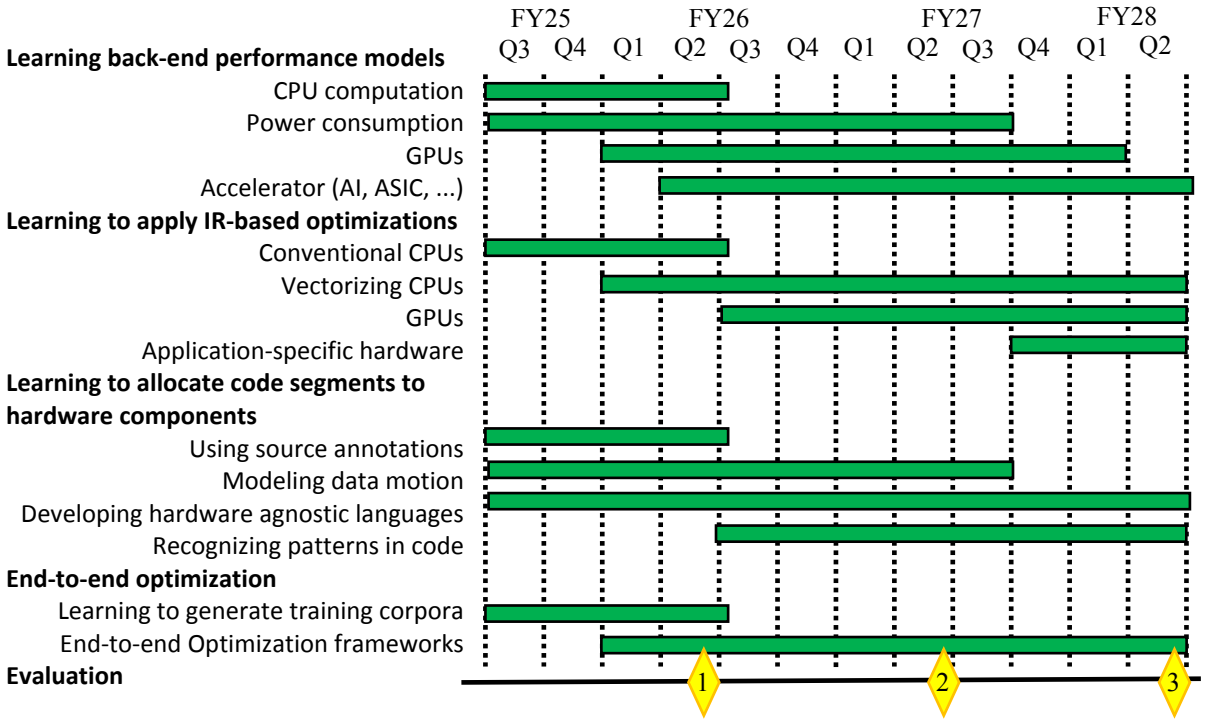


Figure 4 MOCHA program schedule

Figure 4 is intended to be an example of the types of detailed dependencies and prerequisites MOCHA performers will need to address. The program-level focus of MOCHA is encouraged to be at the roll-up level, with the primary focus for Performance Year 1 being on back-end technology that will allow the rapid construction of performance models to guide code generation. Performance Year 2 should incorporate data-driven and ML techniques that inform optimization selection and ordering, as well as the construction of relevant intermediate forms. Performance Year 3 should incorporate front-end techniques for partitioning source code into modules and mapping these to relevant CEs.

The experimental setup for assessment should involve several elements:

1. Number of CEs in the computing fabric
2. Number of distinct *types* of CEs in the computing fabric
3. Number of criteria (e.g., throughput, power consumption, memory footprint) in the compiler's figure of merit
4. Size of the source code being compiled.

These assessment elements should increase in each performance year (see Figure 5). For each annual assessment cycle there will be a mini-assessment at 6-month boundaries to validate the assessment suite and ensure performer progress remains aligned with program goals (1.1, 2.1, 3.1), and a full assessment at yearly boundaries (1.2, 2.2, 3.2).

Eval #	*# Component Types*	*Total # Components*	*Criteria in Figure of Merit*	*Source Code Size*
1.1	*2*	*2*	*1*	*10,000*
1.2	*2*	*4*	*1*	*25,000*
2.1	*3*	*6*	*2*	*50,000*
2.2	*4*	*8*	*2*	*100,000*
3.1	*5*	*10*	*3*	*500,000*
3.2	*6*	*12*	*3*	*1,000,000*

Figure 5 Experimental conditions

Program Metrics

Two key metrics will guide the program:

1. Human effort reduction, specifically, how much human effort is required to accommodate a new heterogeneous computing ensemble. This will be measured by the number of lines of compiler annotations that are required.
2. Performance of the compiled code (throughput, power consumption, memory footprint) compared to results from conventional approaches.

For middle- and back-end components, human effort will be measured by the number of lines of code or annotation (e.g., @GPU, @loop_unroll, @nosideefects) that is required to construct a new performance model or orchestrate a collection of optimizations using tools and techniques developed under the MOCHA program. For front-end components, human effort will be measured by the number of lines of code or annotation that is required to enable the compiler to effectively identify modules and code segments that can exploit distinct CEs and allocate those segments to the CEs. In both cases, the time required to build the toolchains or languages is not being measured, just the time to adapt them to specific test cases.

As discussed above, performance models are expected to model not just computational throughput, but also power consumption and memory requirements, which will in turn enable the

compiler to optimize code generation against any of those three measures. Thus, the MOCHA performance metric will be measured along each of those dimensions.

Program performance targets are shown in Figure 6 below.

Metric	Year 1	Year 2	Year 3
Human effort reduction	50%	75%	90%
Code performance gain	1x	2.5x	5x

Figure 6 MOCHA performance objectives

Intellectual Property

A goal of MOCHA is to develop new technologies that enable compilers to rapidly support a growing proliferation of heterogeneous computing elements. The majority of compiler toolchains today are open-source products. A successful incorporation of MOCHA research into the compilers used by DoD and the Defense Industrial Base (DIB) is expected to require incorporation into one of the open-source projects. Thus, to the maximum extent possible, creating open-source software is strongly encouraged. At a minimum the Government desires all noncommercial software (including source code), software documentation, hardware designs and documentation, and technical data generated under the program be delivered to the Government, with a minimum of Government Purpose Rights.

Section II: Evaluation Criteria

- Proposals will be evaluated using the following criteria listed in ***descending order of importance***: Overall Scientific and Technical Merit, Potential Contribution and Relevance to the DARPA Mission, and Cost Realism.

    - **Overall Scientific and Technical Merit**: The proposed technical approach is innovative, feasible, achievable, and complete. The proposed technical team has the expertise and experience to accomplish the proposed tasks. Task descriptions and associated technical elements provided are complete, with all proposed deliverables clearly defined such that a final outcome that achieves the goal can be expected as a result of award. The proposal identifies major technical risks of the proposed approach and planned mitigation efforts are clearly defined and feasible.

    - **Potential Contribution and Relevance to the DARPA Mission**: The potential contributions of the proposed effort bolster the national security technology base and support DARPA's mission to make pivotal early technology investments that create or prevent technological surprise. The proposer clearly demonstrates its capability to transition the technology into a self-sustaining compiler community. In addition, the evaluation will take into consideration the extent to which the proposed intellectual property (IP) rights structure will potentially impact the Government's ability to transition the technology.

    - **Cost Realism**: The proposed costs are realistic for the technical and management approach and accurately reflect the technical goals and objectives of the solicitation. The proposed costs are consistent with the proposer's Statement of Work and reflect a sufficient understanding of the costs and level of effort needed to successfully accomplish the proposed technical approach. The costs for the prime proposer and proposed sub awardees are substantiated by the details provided in the proposal (e.g., the type and number of labor hours proposed per task, the types and quantities of materials, equipment and fabrication costs, travel and any other applicable costs and the basis for the estimates). It is expected that the effort will leverage all available relevant prior research to obtain the maximum benefit from the available funding. For efforts with a likelihood of commercial application, appropriate direct cost sharing may be a positive factor in the evaluation. DARPA recognizes that undue emphasis on cost may motivate proposers to offer low-risk ideas with minimum uncertainty and to staff the effort with junior personnel to be in a more competitive posture. DARPA discourages such cost strategies.

- Unless otherwise specified in this announcement, for additional information on how DARPA reviews and evaluates proposals through the Scientific Review Process, please visit: Proposer Instructions and General Terms and Conditions.

Section III: Submission Information

- This announcement allows for the award of multiple instrument types to include Procurement Contracts, Cooperative Agreements, and Other Transactions. Some award instrument types have specific cost-sharing requirements (e.g., Research Other Transactions). The following websites are incorporated by reference and contain additional information regarding overall proposer instructions, general terms and conditions, and each specific award instrument type.

    - **Proposer Instructions and General Terms and Conditions**: Proposer Instructions and General Terms and Conditions
    - **Procurement Contracts**: Proposer Instructions: Procurement Contracts
    - **Assistance (Grants and Cooperative Agreements)**: Proposer Instructions: Grants/Cooperative Agreements
    - **Other Transaction agreements**: Proposer Instructions: Other Transactions

- This announcement contains an abstract phase. Abstracts are strongly encouraged, but not required. Abstracts are due August 22, 2024, at 1:00 PM ET as stated in the Overview section. Additional instructions for abstract submission through the Broad Agency Announcement Tool (BAAT) are contained within **the Bundle of Attachments: Abstract Instructions and Template**.

- Full proposals are due September 26, 2024, at 1:00 PM ET as stated in the Overview section. The Bundle of Attachments contain specific instructions and templates. **Required Attachments** constitute a full proposal submission. Please visit Proposer Instructions and General Terms and Conditions for specific information regarding submission methods through the Broad Agency Announcement Tool (BAAT).

- **Bundle of Attachments**:

    Use of the following templates is strongly encouraged for all proposal submissions to this BAA. While the following templates are not mandatory (although strongly encouraged), a full proposal submission must include the following required segments, as noted immediately below.

    - **(strongly encouraged)** Abstract Instructions and Template (MOCHA)
    - **(required)** Proposal Summary Slide Template
    - **(required)** Proposal Instructions and Volume I Template (Technical and Management)
    - **(required)** Proposal Instructions and Volume II Template (Cost)
    - **(strongly encouraged)** DARPA Standard Cost Proposal Spreadsheet
    - **(informational)** Associate Contractor Agreement (ACA) (MOCHA)

Section IV: Special Considerations

- This announcement, the Bundle of Attachments, and websites incorporated by reference constitute the entire solicitation. In the event of a discrepancy between the announcement, attachments, or websites, the announcement shall take precedence.

- All responsible sources capable of satisfying the Government's needs, including both U.S. and non-U.S. sources, may submit a proposal that shall be considered by DARPA. Historically Black Colleges and Universities, Small Businesses, Small Disadvantaged Businesses, and Minority Institutions are encouraged to submit proposals and join others in submitting proposals; however, no portion of this announcement will be set aside for these organizations' participation due to the impracticality of reserving discrete or severable areas of this research for exclusive competition among these entities. Non-U.S. organizations and/or individuals may participate to the extent that such participants comply with any necessary nondisclosure agreements, security regulations, export control laws, and other governing statutes applicable under the circumstances.

- As of the time of publication of this solicitation, all proposal submissions are anticipated to be unclassified.

- This MOCHA program is subject to an Associate Contractor Agreement (ACA). Because the technical scope of the program is broad, DARPA encourages collaboration and technology interchange between the performers and will require MOCHA performers to negotiate and sign an ACA. See the Bundle of Attachments: Associate Contractor Agreement for more information.

- Federally Funded Research and Development Centers (FFRDCs), University Affiliated Research Centers, and Government entities interested in participating in the MOCHA program or proposing to this announcement should first contact the Agency Point of Contact (POC) listed in the Overview section prior to the **Abstract** due date to discuss eligibility. Complete information regarding eligibility can be found at Proposer Instructions and General Terms and Conditions.

- As of the date of publication of this solicitation, the Government expects that program goals as described herein may be met by proposed efforts for fundamental research and non-fundamental research. Some proposed research may present a high likelihood of disclosing performance characteristics of military systems or manufacturing technologies that are unique and critical to defense. Based on the anticipated type of proposer (e.g., university or industry) and the nature of the solicited work, the Government expects that some awards will include restrictions on the resultant research that will require the awardee to seek DARPA permission before publishing any information or results relative to the program. For additional information on fundamental research, please visit Proposer Instructions and General Terms and Conditions.

  Proposers should indicate in their proposal whether they believe the scope of the research included in their proposal is fundamental or not. While proposers should clearly explain the

intended results of their research, the Government shall have sole discretion to determine whether the proposed research shall be considered fundamental and to select the award instrument type. Appropriate language will be included in resultant awards for non-fundamental research to prescribe publication requirements and other restrictions, as appropriate. This language can be found at Proposer Instructions and General Terms and Conditions.

For certain research projects, it may be possible that although the research to be performed by a potential awardee is non-fundamental research, its proposed sub awardee's effort may be fundamental research. It is also possible that the research performed by a potential awardee is fundamental research while its proposed sub awardee's effort may be non-fundamental research. In all cases, it is the potential awardee's responsibility to explain in its proposal which proposed efforts are fundamental research and why the proposed efforts should be considered fundamental research.

- DARPA's Fundamental Research Risk-Based Security Review Process (FERBS) (formerly CFIP) is an adaptive risk management security program designed to help protect the critical technology and performer intellectual property associated with DARPA's research projects by identifying the possible vectors of undue foreign influence. The DARPA team will create risk assessments of all proposed Senior/Key Personnel selected for negotiation of a fundamental research cooperative agreement award. The DARPA risk assessment process will be conducted separately from the DARPA scientific review process and adjudicated prior to final award. For additional information on this process, please visit Proposer Instructions: Grants/Cooperative Agreements.

  Additional Resources:

- The APEX Accelerators program, formerly known as the Procurement Technical Assistance Program (PTAP), focuses on building strong, sustainable, and resilient U.S. supply chains by assisting a wide range of businesses that pursue and perform under contracts with the DoD, other federal agencies, state and local governments, and with government prime contractors. See https://www.apexaccelerators.us/ for more information.

  APEX Accelerators helps businesses:

  - o Complete registration with a wide range of databases necessary for them to participate in the government marketplace (e.g., SAM).
  - o Identify which agencies and offices may need their products or services and how to connect with buying agencies and offices.
  - o Determine whether they are ready for government opportunities and how to position themselves to succeed.
  - o Navigate solicitations and potential funding opportunities.
  - o Receive notifications of government contract opportunities on a regular basis.
  - o Network with buying officers, prime contractors, and other businesses.
  - o Resolve performance issues and prepare for audit, only if the service is needed, after receiving an award.

- DARPAConnect offers free resources to potential performers to help them navigate DARPA, including "Understanding DARPA Award Vehicles and Solicitations," "Making the Most of Proposers Days," and "Tips for DARPA Proposal Success." Join DARPAConnect at www.DARPAConnect.us to leverage on-demand learning and networking resources.

- Project Spectrum is a nonprofit effort funded by the DoD Office of Small Business Programs to help educate the Defense Industrial Base (DIB) on compliance. Project Spectrum is vendor-neutral and available to assist businesses with their cybersecurity and compliance needs. Their mission is to improve cybersecurity readiness, resilience, and compliance for small/medium-sized businesses and the federal manufacturing supply chain. Project Spectrum events and programs will enhance awareness of cybersecurity threats within the manufacturing, research and development, as well as knowledge-based services sectors of the industrial base. Project Spectrum will leverage strategic partnerships within and outside of the DoD to accelerate the overall cybersecurity compliance of the DIB.

  www.Projectspectrum.io is a web portal that will provide resources, such as individualized dashboards, a marketplace, and Pilot Program to help accelerate cybersecurity compliance.

- DARPA has streamlined our Broad Agency Announcements and is interested in your feedback on this new format. Please send any comments to DARPAsolicitations@darpa.mil.

References

1. Chen, T., Moreau, T., Jiang, Z., Yan, E., Cowan, M., Shen, H., . . . Krishnamurthy, A. (2018, October 8). TVM: An automated end-to-end optimizing compiler for deep learning. OSDI'18: Proceedings of the 13th USENIX conference on Operating Systems Design and Implementation, 579-594.
2. Chen, Y., Mendis, C., Carbin, M., & Amarasinghe, S. (2021, April 17). VeGen: a vectorizer generator for SIMD and beyond. Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, 902-914. doi:10.1145/3445814.3446692
3. Ejjeh, A., Councilman, A., Kothari, A., Kotsifakou, M., Medvinsky, L., Noor, A. R., . . . Adve, V. (2022). HPVM: Hardware-Agnostic Programming for Heterogeneous Parallel Systems. IEEE Micro, 42(5), 108-117. doi:10.1109/MM.2022.3186547
4. Huang, B.-Y., Zhang, H., Subramanyan, P., Vizel, Y., Gupta, A., & Malik, S. (2018, December 21). Instruction-Level Abstraction (ILA): A Uniform Specification for System-on-Chip (SoC) Verification. (N. Chang, Ed.) ACM Transactions on Design Automation of Electronic Systems (TODAES), 24(1), 1-24. doi:10.1145/3282444
5. Lattner, C., Amini, M., Bondhugula, U., Cohen, A., Davis, A., Pienaar, J., . . . Zinenko, O. (2021, February 27). MLIR: Scaling compiler infrastructure for domain specific computation. 2021 IEEE/ACM International Symposium on Code Generation and Optimization (CGO). doi:10.1109/CGO51591.2021.9370308
6. Mendis, C., Renda, A., Amarasinghe, S., & Carbin, M. (2019, June). Ithemal: Accurate, portable and fast basic block throughput estimation using deep neural networks. (K. Chaudhuri, & R. Salakhutdinov, Eds.) Proceedings of the 36th International Conference on Machine Learning, 97, 4505-4515. doi:10.48550/arXiv.1808.07412
7. Mendis, C., Yang, C., Pu, Y., Amarasinghe, S., & Carbin, M. (2019, December 8). Compiler auto-vectorization with imitation learning. Proceedings of the 33rd International Conference on Neural Information Processing Systems, 14625-14635. doi:10.5555/3454287.3455597
8. Tate, R., Stepp, M., Tatlock, Z., & Lerner, S. (2009, January 21). Equality saturation: a new approach to optimization. Proceedings of the 36th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages, 264-276. doi:10.1145/1480881.148091
9. Trofin, M., Qian, Y., Brevdo, E., Lin, Z., Choromanski, K., & Li, D. (2021, January 13). MLGO: a Machine Learning Guided Compiler Optimizations Framework. arXiv preprint, 12. doi:10.48550/arXiv.2101.04808
10. Willsey, M., Nandi, C., Wang, Y. R., Flatt, O., Tatlock, Z., & Panchekha, P. (2021, January 4). egg: Fast and extensible equality saturation. Proceedings of the ACM on Programming Languages, 5(POPL), 1-29. doi:10.1145/3434304
11. Cummins, C., Seeker, V., Grubisic, D., Liang, Y., Elhoushi, M., Roziere, B., . . . Leather, H. (2023). Large Language Models for Compiler Optimization. doi:10.48550/arXiv.2309.07062
12. Cummins, C., Seeker, V., Grubisic, D., Rozière, B., Gehring, J., Synnaeve, G., & Leather, H. (2024). Meta Large Language Model Compiler: Foundation Models of Compiler Optimization.