

Office of the Secretary Of Defense (OSD)
Assistant Secretary of Defense (Research & Engineering)
11.B Small Business Technology Transfer (STTR)
Proposal Submission Instructions

Introduction

The Assistant Secretary of Defense (Research & Engineering) STTR Program is sponsoring five topics in the Cyber-Physical Systems technology focus area in this solicitation.

The Navy and Air Force are participating in the OSD program on this solicitation. The service laboratories act as OSD's Agent in the management and execution of the contracts with small businesses. The service laboratories, often referred to as a DoD Component acting on behalf of the OSD, invite small businesses to submit proposals under the Small Business Technology Transfer Research (STTR) Program Solicitation.

In order to participate in the OSD STTR Program, all potential proposers should register on the DoD SBIR/STTR Web site at <https://www.dodsbir.net/submission> as soon as possible. Follow the instructions for electronic submittal of proposals. It is required that all proposers submit their proposal electronically through the DoD SBIR/STTR Proposal Submission Website at <https://www.dodsbir.net/submission>. If you experience problems submitting your proposal, call the SBIR/STTR Help Desk (toll free) at 1-866-724-7457.

Refer to Section 1.5 of the DoD Program Solicitation for the process of submitting questions on STTR and Solicitation Topics. During the Pre-release period proposers have an opportunity to contact topic authors by telephone or e-mail to ask technical questions about specific solicitation topics, however, proposal evaluation is conducted only on the written proposal. Contact during the Pre-release period is considered informal, and will not be factored into the selection for award of contracts. Contact with the topic authors by telephone or e-mail subsequent to the Pre-release period is prohibited. To obtain answers to technical questions during the formal Solicitation period, please visit <http://www.dodsbir.net/sitis>. Refer to the front section of the solicitation for the exact dates

OSD WILL NOT accept any proposals that are not submitted through the on-line submission site. The submission site does not limit the overall file size for each electronic proposal; but there is only a **25-page limit**. File uploads may take a great deal of time depending on your file size and your internet server connection speed. If you wish to upload a very large file, it is highly recommended that you submit prior to the deadline submittal date, as the last day is heavily trafficked. You are responsible for performing a virus check on each technical proposal file to be uploaded electronically. The detection of a virus on any submission may be cause for the rejection of the proposal.

Firms with strong research and development capabilities in science or engineering in any of the topic areas described in this section and with the ability to commercialize the results are encouraged to participate. Subject to availability of funds, the ASD(R&E) STTR Program will support high quality research and development proposals of innovative concepts to solve the listed defense-related scientific or engineering problems, especially those concepts that also have high potential for commercialization in the private sector. Objectives of the ASD(R&E) STTR Program include stimulating technological innovation, strengthening the role of small business in meeting DoD research and development needs, fostering and encouraging participation by minority and disadvantaged persons in technological innovation, and increasing the commercial application of DoD-supported research and development results. The guidelines presented in the solicitation incorporate and exploit the flexibility of the SBA

Policy Directive to encourage proposals based on scientific and technical approaches most likely to yield results important to DoD and the private sector.

Proposal Submission

Refer to Sections 3.0 and 6.0 of the DoD Program Solicitation for program requirements and proposal submission instructions. Proposals shall be submitted in response to a specific topic identified in the following topic description sections. The topics listed are the only topics for which proposals will be accepted. Scientific and technical information assistance may be requested by using the SBIR/STTR Interactive Technical Information System (SITIS). OSD's technical proposal page limit is 25 pages.

Description of the OSD STTR Three Phase Program

Phase I is to determine, insofar as possible, the scientific or technical merit and feasibility of ideas submitted under the STTR Program and will typically be one half-person year effort over a period not to exceed six months, with a dollar value up to \$100,000. OSD plans to fund three Phase I contracts, on average, and downselect to one Phase II contract per topic. This is assuming that the proposals are sufficient in quality to fund this many. Proposals are evaluated using the Phase I evaluation criteria, in accordance with Section 4.2 of the DoD Program Solicitation. Proposals should concentrate on that research and development which will significantly contribute to proving the scientific and technical feasibility of the proposed effort, the successful completion of which is a prerequisite for further DoD support in Phase II. The measure of Phase I success includes technical performance toward the topic objectives and evaluations of the extent to which Phase II results would have the potential to yield a product or process of continuing importance to DoD and the private sector, in accordance with Section 4.3 of the DoD Program Solicitation.

Subsequent Phase II awards will be made to firms on the basis of results from the Phase I effort and the scientific and technical merit of the Phase II proposal in addressing the goals and objectives described in the topic. Phase II awards will typically cover two to five person-years of effort over a period generally not to exceed 24 months (subject to negotiation) with a dollar value up to \$750,000. Phase II is the principal research and development effort and is expected to produce a well defined deliverable prototype or process. A more comprehensive proposal will be required for Phase II.

For Phase II, no separate solicitation will be issued. Only firms awarded Phase I contracts, and have successfully completed their Phase I efforts, may be invited to submit a Phase II proposal. Invitations to submit Phase II proposals will be released approximately at the end of the Phase I period of performance. The decision to invite a Phase II proposal will be made based upon the success of the Phase I contract to meet the technical goals of the topic, as well as the overall merit based upon the criteria in Section 4.3. DoD is not obligated to make any awards under Phase I, II, or III. For specifics regarding the evaluation and award of Phase I or II contracts, please read the front section of this solicitation very carefully. Phase II proposals will be reviewed for overall merit based upon the criteria in Section 4.3 of this solicitation.

Under Phase III, the DoD may award non-STTR funded follow-on contracts for products or processes, which meet the component mission needs. This solicitation is designed, in part, to encourage the conversion of federally sponsored research and development innovation into private sector applications. The small business is expected to use non-federal capital to pursue private sector applications of the research and development.

This solicitation is for Phase I proposals only. Any proposal submitted under prior STTR solicitations will not be considered under this solicitation; however, offerors who were not awarded a

contract in response to a particular topic under prior STTR solicitations are free to update or modify and submit the same or modified proposal if it is responsive to any of the topics listed in this section.

Phase II Plus Program

The OSD STTR Program has a Phase II Plus Program, which provides matching STTR funds to expand an existing Phase II contract that attracts investment funds from a DoD acquisition program, a non-SBIR/non-STTR government program or Private sector investments. Phase II Plus allows for an existing Phase II OSD STTR contract to be extended for up to one and a half year per Phase II Plus application, to perform additional research and development. Phase II Plus matching funds will be provided on a one-for-one basis up to a maximum \$500,000 of STTR funds. All Phase II Plus awards are subject to acceptance, review, and selection of candidate projects, are subject to availability of funding, and successful negotiation and award of a Phase II Plus contract modification. The funds provided by the DoD acquisition program or a non-SBIR/non-STTR government program must be obligated on the OSD Phase II contract as a modification prior to or concurrent with the OSD STTR funds. Private sector funds must be deemed an “outside investor” which may include such entities as another company, or an investor. It does not include the owners or family members, or affiliates of the small business (13 CFR 121.103).

Fast Track Policy

The Fast Track provisions in Section 4.5 of this solicitation apply as follows. Under the Fast Track policy, STTR projects that attract matching cash from an outside investor for their Phase II effort have an opportunity to receive interim funding between Phases I and II, to be evaluated for Phase II under an expedited process, and to be selected for Phase II award provided they meet or exceed the technical thresholds and have met their Phase I technical goals, as discussed in Section 4.5. Under the Fast Track Program, a company submits a Fast Track application, including statement of work and cost estimate, within 120 to 180 days of the award of a Phase I contract (see the Fast Track Application Form on www.dodsbir.net/submission). Also submitted at this time is a commitment of third party funding for Phase II. Subsequently, the company must submit its Phase I Final Report and its Phase II proposal no later than 210 days after the effective date of Phase I, and must certify, within 45 days of being selected for Phase II award, that all matching funds have been transferred to the company. For projects that qualify for the Fast Track (as discussed in Section 4.5), DoD will evaluate the Phase II proposals in an expedited manner in accordance with the above criteria, and may select these proposals for Phase II award provided: (1) they meet or exceed selection criteria (a) and (b) above and (2) the project has substantially met its Phase I technical goals (and assuming budgetary and other programmatic factors are met, as discussed in Section 4.1). Fast Track proposals, having attracted matching cash from an outside investor, presumptively meet criterion (c). However, selection and award of a Fast Track proposal is not mandated and DoD retains the discretion not to select or fund any Fast Track proposal.

Follow-On Funding

In addition to supporting scientific and engineering research and development, another important goal of the program is conversion of DoD-supported research and development into commercial products. Proposers are encouraged to obtain a contingent commitment for private follow-on funding prior to Phase II where it is felt that the research and development has commercial potential in the private sector. Proposers who feel that their research and development have the potential to meet private sector market needs, in addition to meeting DoD objectives, are encouraged to obtain non-federal follow-on funding for Phase III to pursue private sector development. The commitment should be obtained during the course of Phase I performance. This commitment may be contingent upon the DoD supported development meeting some specific technical objectives in Phase II which if met, would justify non-

federal funding to pursue further development for commercial purposes in Phase III. The recipient will be permitted to obtain commercial rights to any invention made in either Phase I or Phase II, subject to the patent policies stated elsewhere in this solicitation.

The following pages contain a summary of the technology focus areas, followed by the topics.

Cyber-Physical Systems Technology Area Topics

The Department of Defense (DoD) is heavily reliant on computational power to perform rapid analysis of sensor data, search for records in databases, and to control complex machines. To retain our technical edge, the DoD must be able to take advantage of cutting edge techniques in parallel computing on multi-processor systems.

The topics written under this theme focus on:

1. Parallelization, multi-threading, scheduling, and concurrency are some of the active areas of basic research in software designed for multi-processor systems. In the context of this theme, multi-processor can mean multiple cores on the same die or multiple, distributed processors (e.g., as in utility computing). This theme has these technical goals:
2. Identify and measure the properties of multi-processor operating systems that most affect system cost, performance, and development time.
3. Create operating systems that take advantage of abstract concepts to efficiently partition, communicate, and execute programs on multi-processor systems. These abstractions should allow continued and efficient operation of applications when the number of processors, memory, and available communications bandwidth change; and they should be easily maintainable, adaptable, and usable by people with no more than a few days of specialized training.

Topics in this theme are:

- OSD11-T01 Multi-Processor Computer Supervisory Control Development, Verification, and Validation
- OSD11-T02 Programming Constructs to Enable Formation of Efficient Algorithm Mapping for ExaScale Processors
- OSD11-T03 Design and Analysis of Multi-core Software
- OSD11-T04 Operating System Mechanisms for Many-Core Systems

OSD STTR 11.B Topic Index

OSD11-T01	Multi-Processor Computer Supervisory Control Development, Verification, and Validation
OSD11-T02	Programming Constructs to Enable Formation of Efficient Algorithm Mapping for ExaScale Processors
OSD11-T03	Design and Analysis of Multi-core Software
OSD11-T04	Operating System Mechanisms for Many-Core Systems
OSD11-TD1	Information Salience

OSD STTR 11.B Topic Descriptions

OSD11-T01 TITLE: Multi-Processor Computer Supervisory Control Development, Verification, and Validation

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Develop techniques, algorithms, and architectures to enable automated system-level supervisory control of jobs running on multi-processor systems. Develop performance metrics to evaluate this supervisory control.

DESCRIPTION: As the DoD progresses toward more multi-processor systems, the need to automate dispatching of compute jobs and also to verify and validate this control becomes vital to efficient system performance. Currently, there is a human in the loop in many places for supervising the subdivision of compute jobs. Developing performance metrics for a supervisory control system and designing a system to meet these metrics is the first step toward automating this process. Subsequent verification and validation of the supervisory control system to perform more of this control function could lead to more efficient, effective, and reliable system performance.

Also, such a system could increase the autonomy and capability of an unmanned combat air vehicle (UCAV), though verification and validation (V&V) would be necessary to ensure that the supervisory control worked effectively and reliably. The verification process should evaluate whether the supervisory control system operates in compliance with any applicable specifications, regulations, or conditions which are determined prior to or during the design phase. The validation process shall ensure with a high level of assurance that the supervisory control system accomplishes its specified mission, meeting required performance metrics.

Multi-processor systems break into multiple jobs running simultaneously. These jobs are often interdependent. There is a need for supervisory controls to run the job allocation, ensuring that interdependent jobs are properly coordinated for efficient, rapid system performance. While device and processor level V&V has been established, a control mechanism is needed to supervise multiple jobs running together on larger multi-processor systems, with V&V implemented for evaluating the process. The possible use of a cloud computing environment to support an oversight function for the supervisory control could be considered.

Novel approaches are needed to develop computer controls in military operations with highly dynamic constraints and topologies. As the speed of battlefield operations increases, there exists a need to rapidly add, remove, and field assets across multi-vehicle scenarios, involving manned and unmanned aviation, and ground, as well as human assets. Proposed concepts should place special emphasis on autonomous Unmanned Combat Air Systems (UCAS), which deliver intelligence, critical information, and munitions on-demand. This will enable more autonomous, rapid-response system operation, with less need for humans in a high threat environment.

The supervisory control should have the capability of reliably supervising and controlling multiple interrelated computing processes on a multi-processor system, with a verification and validation approach developed to test this supervisor, ensuring that jobs are completed in the proper order to enable the appropriate data to be properly fed from one job to another. This supervisory control should emphasize efficient operation, so that it is practical to field on a UCAV.

PHASE I: Identify novel approaches, supporting algorithms, and architectures to effectively supervise multi-processor computing systems. Develop metrics to perform V&V of supervisory control. Demonstrate the feasibility of the supervisory control approaches through laboratory experimentation and, where appropriate, V&V tools.

PHASE II: Evaluate supervisory control designs with tests on multi-processor systems, using test results to determine performance metrics compliance. Develop a supervisory control framework prototype and evaluate by performing V&V testing.

PHASE III DUAL USE APPLICATIONS:

Military Application: Transition proposed architecture, software, and/or technology components into DoD system(s). An area of special emphasis is in UCAVs. Demonstrate benefits of approach in real-world operations or exercises.

Commercial Application: The technology will be compatible with a large class of multi-processor computing applications in commercial industry. Specifically, this research will benefit emerging commercial research in supervising large multi-user, multi-processor systems.

REFERENCES:

1. DeOrion, A., Wagner, I., & Bertacco, V. (2009). Dakota: Post-silicon validation of the memory subsystem in multi-core designs. High Performance Computer Architecture, 2009, 405-416.
2. Sargent, R. G. (2005). Verification and validation of simulation models (2005). Proceedings of the 2005 Winter Simulation Conference, 130-143.
3. Shimizu, K., Gupta, S., Koyama, T., Omizo, T., Abdulhafiz, J., McConville, L., et al. (2006). Proceedings of the 43rd Annual Design Automation Conference, 338-343.

KEYWORDS: Supervisory Control, Control Systems, Verification and Validation, Unmanned Combat Air Systems

OSD11-T02

TITLE: Programming Constructs to Enable Formation of Efficient Algorithm Mapping for ExaScale Processors

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: The overall objective is to develop new software tools for addressing the requirements of programming the emerging class of computational hardware for Exa-Scale computing. The DoD will benefit through reduced software production costs, and reduced overall life cycle costs through greater portability, fewer defects, and energy reduction.

DESCRIPTION: Researchers and commercial companies are advancing the state of the art in high performance computing through the development of computational technologies that can achieve 100 GFLOPS/W efficiencies. The route to Exa-Scale includes circuit innovations such as Near Threshold Voltage (NTV) operation, which reduces the energy expended per transistor switching event. However, NTV operation also reduces the speed of individual transistors, so that to reach a specific performance level, a greater degree of application parallelism must be provided to the hardware beyond what is provided to conventional multi-core. Movement of data between processors or from processors to and from memory now expends much greater energy than computation. Thus computing hardware increasingly provide for explicit controls for communications and explicit scratchpad memory management. The trends in new accelerators such as GPGPU or the Cell architecture exhibit these features.

The multiplicity of these complexities for Exa-scale computing will require software tools that enable the automatic extraction and formation of parallelism and computation choreographies that minimize communication, or in other words that increase spatio-temporal locality while simultaneously increasing parallelism. Parallelism and locality are considerations that are in tension. There currently exist analytic compiler optimization algorithms that can extract and balance parallelism and locality, so the program expression itself is now becoming the barrier to increasing performance. Program expressions in common high level languages such as C force sub-optimal bindings, e.g. array layout, and obscure semantic relationships and thus restrict the parallelism that can be extracted. The solution is even higher level program expressions, that provide for more precision in the expression of application semantics, and that reduce extraneous constraints. These higher level program expressions can provide for more succinct specifications of software, and also can enable new parallelization and mapping opportunities such as cross-function fusion or exploitation of mathematical properties for reformulation. Rather than new parallel computing languages, these higher level expressions can be accomplished by embedding them in an existing language, as surface syntax on a more structured expression model that exposes more parallelism with the

conventional high level language. Separating the semantics from the mapping is critical in providing the benefits of this approach.

Improving programming tools can reduce software lifecycle costs. Succinct expression also means fewer lines of code which allows proportionally greater productivity and reduced opportunities for costly errors. Higher level specifications enable greater portability by maximizing the semantic application-specific information while minimizing extraneous bindings to the original target hardware. Portability reduces costs by avoiding hardware platform lock-in, and also allows mission capability increases by more rapid adoption of new hardware capabilities.

PHASE I: Co-design higher level algorithmic expressions with specific optimizations for parallelism and locality. Perform proof of concept measurement of the improvements from the use of the higher level algorithmic expressions in increasing parallelism or locality beyond that which can be achieved with common high level languages. Develop mathematical basis of the underlying optimization model to illustrate the means by which it enables exploitation of parallelism and locality across stages of sensor signal processing computational chains. Benchmark on model kernels relevant to Air Force sensor missions such as image formation, moving target detection, or advanced trackers.

PHASE II: Implement new higher level algorithm specific language and optimizations. Test and tune on sensor mission relevant kernels to establish generality over air force missions; target toward multiple processing targets to establish porting benefits. Demonstrate capability to exploit parallelism and locality across stages of sensor computational chains. Demonstrate methodology for applying to tactical kernels on Air Force sensor missions.

PHASE III DUAL USE COMMERCIALIZATION: Deliver tool for tactical code development and optimization, feature enhancements, performance enhancements, and customer support. Commercial applications include energy exploration, manufacturing, bio-computing, and financial computing.

REFERENCES:

1. Kogge, Peter, Editor, "ExaScale Computing Study: Technology Challenges in Achieving ExaScale Systems," DARPA IPTO Technical Report, September 28, 2008.
2. Campbell, Dan, "The High Performance Embedded Computing Software Initiative: C++ and Parallelism Extensions to the Vector, Signal, and Image Processing Library Standard," DOD Users Group Conference, 2004.
3. Hartono, Albert et. al., "Automatic Parallelization and Locality Optimization of Beamforming Algorithms," High Performance Embedded Computing Workshop, MIT Lincoln Labs, 2010.
4. N. Frolov, M. Sjaelander, P. Larsson-Edefors, and Sally A. McKee, "A SAT- Based Compiler for FlexCore", Technical report - Department of Computer Science and Engineering, Chalmers University of Technology and Gothenburg University, ISSN 1652-926X, No: 11-04, 2011.
5. Kulkarni, Milind et. al., "Optimistic Parallelism Requires Abstractions," ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), 2007.

OSD11-T03

TITLE: Design and Analysis of Multi-core Software

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Develop innovative technologies for early design and analysis of multi-threaded software for multi-core systems enabling robust development of distributed, real-time systems targeted for dynamic environments.

DESCRIPTION: The multi-core era has significantly impacted the way software is developed. While multi-core processors are expected to increase system performance, attaining increased performance is not straightforward. Parallelization of software required for multi-core systems increases the complexity of the system, as additional requirements to account for shared resources brought about by concurrency introduce new performance bottlenecks,

such as lock contention, cache coherence overheads, and lock convoys [1] not seen in single core systems. Parallel programming models, such as PThreads, OpenMP, and Intel Thread Building Blocks, enable concurrent programming, however these programming models require the software developer be an expert in low level multi-threading programming (e.g. Synchronization, Communication, load balancing, and scalability) and require vendor dependent multi-core-specific architecture knowledge to program (e.g. cache size, shared resources, heterogeneous cores) [2]. Furthermore, portability is significantly hindered because multi-core architectures differ significantly, requiring applications be adapted to each new platform, resulting in limited portability of code [3]. Moving the programming model to a higher level of abstraction can overcome these difficulties, including providing platform independence, which will help to reduce the complexity introduced by concurrent programming. For example, an actor-oriented design might reduce the complexity and non-determinism of multi-threaded code by using these design constructs to hide the underlying thread implementation [4].

This topic seeks to increase the correctness of embedded mission critical software on multi-core processors through the development of early design and analysis methodologies and techniques that improve predictability and dependability. Due to the size (measured by lines of codes, components etc.) and complexity (multi-domain dimensionality, interconnection, etc.) of single-core software, analysis and testing during development and integration phases cannot readily verify and validate the correctness of the software system under development. As systems transition into the multi-core era, the complexity is further exacerbated as the software is expected to operate in a parallel manner across the cores, maintaining performance increases lost by the inability to continue increasing the clock frequency. The concurrency of multi-core processors requires the software developer to be aware of shared resources between cores to maintain correct operation of the software. Shared resources, such as L2 cache, have a significant impact on the performance of multi-threaded software; 20-plus percent performance reductions have frequently been reported due to inattentive developers and lack of tool support. To alleviate the complexity developers' face, model-centric software development and analysis tools, which raise the level of developer abstraction, allow for complete performance analysis early in the design cycle, and provide automated artifact generation (e.g. code, scripts, tests, deployment plans, etc.) are being sought.

PHASE I: Define the feasibility of novel approaches that will enable early design, analysis, test and validation of software targeted for multi-core machines. Select appropriate design, analysis, test and validation techniques and develop the conceptual approach and design to integrate the technologies. Define metrics to measure improvements offered by the concept and present an appropriate validation and verification plan.

PHASE II: Develop, design and demonstrate the novel approaches from Phase I in a prototype. Define and implement the appropriate tool and interface approaches, and methodologies for integration. Demonstrate the prototype's openness, scalability, and degree of automation in the exchange of design data by accomplishing performance, analysis, validation and verification against a representative DoD design.

PHASE III DUAL USE COMMERCIALIZATION: The ability to design and analyze software for multi-core systems has clear benefits for both the private and military sectors. With the shift to multi-core systems the only way to continue to achieve increases in performance is to develop parallel software. Design and analysis tools to help in the development of parallel software will be applicable to the private and public sectors since both domains are facing the same challenges of migrating to multi-core systems.

REFERENCES:

- [1] Sutter and Larus. Software and the Concurrency Revolution. ACM Queue Vol. 3 Issue 7 Sept 2005
- [2] Hsiung, S. Lin, Chen, Hsueh, Chang, Shih, Koong, C. Lin, Lu, Tong, Su, and Chu. Model-Driven Development of Multi-Core Embedded Software. 2009 ICSE Workshop on Multicore Software Engineering. May 2009
- [3] Pllana, Benkner, Mehofer, Natvig, and Xhafa. Towards an Intelligent Environment for Programming Multi-core Computing Systems. Euro-Par 2008 Workshops - Parallel Processing. August 2008
- [4] Lee Edward. The problem with threads, IEEE Computer, vol. 29, no. 5. May 2006

KEYWORDS: Multi-threaded Design, Software Design Analysis, Multi-core Software, Model-based Software Development

OSD11-T04

TITLE: Operating System Mechanisms for Many-Core Systems

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Design and develop innovative technologies for operating system services that take advantage of abstract concepts to efficiently partition, communicate, and execute programs on many-core, multi-processor systems.

DESCRIPTION: The Department of Defense (DoD) is heavily reliant on computational power to perform rapid analysis of sensor data, search for records in databases, and to control complex machines. To retain our technical edge, the DoD must be able to take advantage of cutting edge techniques in parallel computing on many-core systems.

In the past new hardware brought higher clock frequency which translated into increase performance. Due to thermal constraints, increasing performance the same way is no longer feasible so chip manufactures are now placing multiple processors (or cores) on a single chip. In doing so it is up to the software developer to increase performance. That is, applications must be developed to execute in parallel, where possible, to take advantage of multiple cores. However, as we move into the many-core era, new operating system (OS) techniques will be needed. Many-core systems will have 10s-1000s of cores, complex interconnects between cores, heterogeneous cores, and complex shared resource management. These disruptive changes to the hardware will cause substantial scalability and correctness challenges for OS designers [1]. "Current operating systems are designed for single processor or small number of processor systems and were not designed to manage such scale of computational resources. The way that an operating system manages 1,000 processors is so fundamentally different than the manner in which it manages one or two processors that the entire design of an operating system must be rethought." [2]

Research under this topic will explore OS mechanisms that take advantage of abstract concepts to efficiently and intelligently partition, communicate, and execute programs on many-core systems. For example a smart scheduler (dynamic or static) that places threads in an optimal layout accounting for shared resources (e.g. cache, interconnects, I/O, etc.) is one possible mechanism being sought. However, it is not the only mechanisms that could be proposed to enable the goals of this topic, other mechanisms that address the scalability issues [1,2,3] will be considered (e.g. memory management mechanisms). These mechanisms should allow continued and efficient operation of applications when the number of processors, memory, available communications bandwidth, interconnects, or other resources change; and they should be easily maintainable, adaptable, and usable by people with no more than a few days of specialized training. The solution needs to consider security aspects of the design in addition to performance and scalability. Solutions may target mechanisms to extend today's operating systems or fundamentally different OSs currently under research, e.g. Barrelfish [1], the Factored Operating Systems (FOS)[2] and OctoPOS [3].

PHASE I: Define operating system mechanisms that take advantage of abstract concepts to efficiently partition, communicate, and execute programs on multi-processor systems. Show how the mechanisms will allow continued and efficient operation of applications when the number of processors, memory, and available communications bandwidth change; and how they will be maintainable, adaptable, and usable by people with no more than a few days of specialized training.

PHASE II: Develop a prototype of the mechanism and demonstrate and validate the concepts and design created during Phase I. Include the solution's openness, scalability, and degree of automation; as well as metrics of performance or analysis against current practice.

PHASE III DUAL USE COMMERCIALIZATION: The ability to design and analyze software for multi-core systems has clear benefits for both the private and military sectors. With the shift to many-core systems the only way to continue to achieve increases in performance is to develop parallel software and scalable OS that handle the

shared resources in an intelligent manner. New OS mechanisms will be applicable to the private and public sectors since both domains are facing the same challenges of migrating to many-core systems.

REFERENCES:

[1] Andrew Baumann, Paul Barham, Pierre-Evariste Dagand, Tim Harris, Rebecca Isaacs, Simon Peter, Timothy Roscoe, Adrian Schüpbach, and Akhilesh Singhanian. The “Multikernel: A new OS architecture for scalable multicore systems”The 22nd ACM Symposium on OS Principles, Big Sky, MT, USA, October 2009.

[2] D. Wentzla and A. Agarwal. “Factored operating systems (fos): The case for a scalable operating system for multicores.” Operating Systems Review, 43(2), Apr. 2009.

[3] Benjamin Oechslein, Jens Schedel, Jürgen Kleinöder, Lars Bauer, Jörg Henkel, Daniel Lohmann, Wolfgang Schröder-Preikschat, “OctoPOS: A Parallel Operating System for Invasive Computing”, <http://research.microsoft.com/en-us/um/people/rmcilroy/sfina/papers%5Csfma-final7.pdf>

KEYWORDS: Multi-core, Many-core, Operating Systems, Scheduler, Memory Management, Multi-threaded Software Execution

OSD11-TD1

TITLE: Information Saliience

TECHNOLOGY AREAS: Information Systems

OBJECTIVE: Identify and understand methods for determining information value as a basis for future decision support systems. Devise mathematical, information science, and computer science representations of information saliience to provide basis for automation and subsequent development of generalized information saliience models and subsequent development of automated decision support systems.

DESCRIPTION: The understanding of how humans process information, determine saliience, and combine seemingly unrelated information is essential to automated processing of the large amounts of partially relevant information or information of unknown relevance. Recent neurological science research in human perception and that in information science regarding context-based modeling provides us with a theoretical basis for using a bottom-up approach for automating management of large amounts of information in ways directly useful for human operators.

Having a way of representing human perception and cognition, both active and unperceived is but the first step. Application of recent information science research in information representation of human-centered perception and cognition is the possibly the most critical piece in bridging the cognitive science to computer science in this multi-disciplinary research topic. Once this is accomplished, it is tractable to apply information contextual models and other techniques leading to development of generalized models and subsequent automation of human cognition-like processes.

Potential research areas are: 1) determining what human users consider to be essential pieces of information from large data elements in various situations (e.g., tasks, stress levels, distractions); 2) how does the human brain separate signal from noise; 3) what unconscious/unperceived processing do the brains of experts carry out that enable the extraction and interpretation of salient information; 4) how does the brain modify its “circuits” to enhance salient cue detection efficiency; 5) what are the salient features that are common across data types; 6) which features are most important as they map to human perception; 7) general description of information saliience and combinations of salient information; 8) general description of situational context; 9) general description of space, time and uncertainty; 9) managing the inexplicit relationship among independently-derived information values; 10) managing individual-to-individual variations that violate the general model; 11) describing essential pieces of information mathematically; and 12) development of efficient algorithms to implement the general mathematical descriptions of information saliience.

The intended environment for this capability would be a military tactical command center with a limited number of intelligence analysts tasked with extracting useful information from large amounts of information from multiple sources including sensors, intelligence reports, unit observations, and media sources. Ultimately, this capability would be used as a basis for developing anticipatory decision support systems for small military unit leaders.

PHASE I: Develop an empirical and then a mathematical framework for representing human perception and cognition. Show correlation of empirical and mathematical approaches using a representative data set.

PHASE II: Develop computer algorithms based on the basic Phase I work. Validate the algorithms against human generated semantic labels on a single or double parameter information set. Demonstrate correlation of the automated method versus the human-generated solution using a representative data set.

PHASE III: Mature Phase II work to develop an automated ontology generator that will process multiple parameter data sets. Validate the system against human analyzed information sets. Demonstrate time savings versus the human-generated data at an equivalent accuracy using a multi-faceted data set.

PRIVATE SECTOR COMMERCIAL POTENTIAL/DULL-USE APPLICATIONS: This STTR if successful, would enable quicker fielding of semantic web capabilities in that semantic labels (e.g., ontologies) could be generated manually allowing much more flexibility in providing new information content more quickly.

REFERENCES:

1. <http://fac-staff.seattleu.edu/duisberr/web/papers/TowardAnInformationMorality-2.pdf>
2. [http://www.tamu.edu/faculty/takashi/psyc689/Stroop%20effect/Melara%20\(stroop%20effect\)%202003.pdf](http://www.tamu.edu/faculty/takashi/psyc689/Stroop%20effect/Melara%20(stroop%20effect)%202003.pdf)
3. <http://www.satechnologies.com/Papers/pdf/InfantrySA.pdf>
4. <http://studentoffortune.com/question/812033/Read-article-and-answer-question/1320490-Article%20to%20read.pdf>
5. <http://delivery.acm.org/10.1145/1300000/1297163/p80-albers.pdf?key1=1297163&key2=4827575031&coll=DL&dl=ACM&ip=138.162.0.41&CFID=23827693&CFTOKEN=55916259>

KEYWORDS: Information value; ontologies; semantic web; decision support