

# A. Summary

This proposal is concerned with enabling parallel computation in a world dominated by Internet technologies. On the hardware side these will include massively parallel engines designed and deployed as Internet servers. On the software side, they will include software developed in network-aware programming languages like Java—software engineered to survive in heterogeneous and very dynamic environments. We will be refining MPI-like programming models and APIs for high performance programming in Java—researching ways to get the fastest possible message passing from Java, and ways to exploit novel Java technologies like Jini to produce richer message-passing environments.

Java introduces implementation issues for message-passing APIs that do not occur in conventional scientific programming languages. We will investigate how to apply ideas from projects such as Jaguar and JaVIA to MPI-like APIs, to reduce the overheads of the Java Native Interface. We will integrate ideas on efficient object serialization from projects like KaRMI with MPI-specific methods for handling complex datatypes. Simply providing efficient MPI-like APIs for Java is not enough. The programming model must address features specific to distributed computing. To better support computing in volatile Internet environments, we will need features like dynamic spawning of process groups, parallel client/server interfaces, and new mechanisms for handling failures. Jini provides a natural framework for implementing these features. An important emphasis will be on researching synergies between parallel message-passing programming and Jini-like systems. By combining ideas from MPI with ideas from Jini we aim to create an environment that encourages scalable, fault-tolerant parallel computing. Finally, we will explore uses of Jini in a middle tier for initiating parallel MPI jobs, where the parallel programs may be written in languages other than Java.