SC 13 BOF report

Asynchronous and Adaptive Parallel Programming with Charm++

The 2nd Charm++ BOF was held at SC 13 on Tuesday Nov 19th, 2013. This BOF was attended by a community interested in parallel programming using Charm++, Adaptive MPI, and the associated ecosystem (mini-languages, tools etc), along with parallel applications developed using them. The BOF was broken up into two sessions - one on Charm++ language and runtime and the second on Charm++ applications. Each session was structured as a series of short presentations followed by a discussion session. The presentations included a broad introduction for the audience, followed by material of interest to the existing and potential user community. The discussion was structured as a moderated panel answering questions from the audience.

In the first session, audience were first introduced to the basic philosophy of Charm++ runtime system. Charm++ is a highly capable, general-purpose parallel programming system based on the notion of over decomposed, migratable units of work/data that are orchestrated by an intelligent, adaptive runtime system. The combination of these two design principles allows it to offer several proven performance and productivity benefits. Charm++ can automatically hide communication latencies, balance dynamically changing loads, exploit accelerators and balance load across devices, provide checkpoint/restarts, and seamlessly overlap the execution of independently developed modules.

Charm++ latest stable release 6.6.0 was announced in this session. Apart from several bug fixes, this release offers some recently added/extended features such as meta-balancer, interoperability, and parallel I/O (CkIO). A summary of these features was given to the audience. Meta-balancer is targeted at automating the decision on “when” and “how” to perform load redistribution in parallel applications which require periodic load balancing. Charm++ advocates interoperability in the parallel ecosystem and the run time as the vehicle to realize this inter-operation. It can interact with components written using MPI, providing an adoption path for existing applications. CkIO is a library which performs asynchronous parallel file system access to enhance I/O performance of applications. Charm++ participation in the collaborative project on Exascale operating system: Argo, was also summarized in this session, with Pete Beckman (ANL) presenting an overview of Argo and its relationship to Charm++ as well the role of the Parallel Programming Laboratory (the research group led by Prof. Kale that develops Charm++) in the overall project.

In the second session, audience were provided short reports from Charm++ application developers, and plans for near future. These applications included NAMD - Molecular Dynamics, EpiSimdemics - Contagion diffusion, Quantum Chemistry, ChaNGa - Cosmology, and Parallel Discrete Event Simulation (PDES). Audience, who were interested in learning more about parallel design considerations, methodology and challenges, and the use of various features of Charm++ in these applications were encouraged to explore the recently released book - Parallel Science and Engineering Applications: The Charm++ Approach.

Besides the two sessions, there was a demonstration of Charm++’s fault tolerance capability using a cluster of Beaglebones - low cost linux computers consisting of ARM processors. The cluster packed in a suitcase aroused the interest of audience. The Charm++ team ran a simple wave simulation on this cluster and showed live visualization of the simulation progress on a projector. During the simulation failures were introduced to this cluster by killing an application process. The failure was detected, and the application continued running after tolerating the failure.

In conclusion, this BOF provided an opportunity for interactions amongst the developers and users of Charm++. Key features of Charm++ were highlighted to introduce them to potential users. New features were presented to solicit feedback on their future development. Late breaking application achievements were discussed to elucidate how they necessitated improvements in Charm++. Future directions were proposed for discussion by the entire community. The BOF resulted in interactions among potential future collaborators - researchers interested in Charm++, those are being followed up.