MEMO: May 7, 1993

TO: Geoffrey

FROM: Gang Cheng

RE: SRC CEM Proposal

CC:

After the meeting with Dr. Cha and Jay of SRC this Tuesday, I, together with Xianneng Shen(Harlington's PhD Student now working at NPAC), went to their office the next morning and further discussed with them issues we need to clarify, especially try to have more understandings about their application problem. Based on my reading through related materials and understanding of this project, I compiled the following info. and some of my thoughts on the technical approaches which need to further discuss with you. It's not elaborated and just reflects my personal view which may be biased, due to the short time involvement of this proposal preparation.

1. Goal of the proposed study (excerpted from the announcement):

The objectives of this project is to conduct an evaluation of existing CEM technology concepts on massively parallel processing(MPP) machines and determine which analytic concepts have potential for further development to model and predicate radar cross section(RCS) of full scale, material treated aerospace vehicles. and demonstrate the potential of advanced CEM(Computational Electromagnetics) techniques when couples to massively parallel computing systems.

Key objectives:
. migrate ARPA sponsored parallel computing technology to CEM techniques.
. develop and demonstrate a major increase in CEM capability
. spearhead the development of new computational methods, and physical modeling techniques that incorporate high levels of parallelism.

Phrase 1 (6 months):

A key element of Phrase 1 is to demonstrate the effective use of MPP computing systems and the rationale for scaling the approach to a full scale fighter aircraft. Criteria to evaluate the Phrase 1's work (listed in descending order of importance), based on evaluations on the EMCC benchmark cases :
. Accuracy
. scalability of the code (risk associated with the potential to model and predict the RCS of a scaled (1/30) materials treated fighter and a full scale PEC model)
. computational speed
. technical documentation

Phrase 2 (12 months):

Continue to develop and refine/modify Phrase 1 software, and as necessary, shall develop new algorithm based on physical modeling concepts, and/or integrate any promising analytical techniques and/or software developed under the separate NASA NRA effort. In Phrase 2, two government specified large-scale demonstrations must be conducted. One predicting the RCS of a full scale perfectly conducting fighter model; the other predicting a material-clad suoscale fighter model.

2. My major observations related to this proposal

Reading through the announcement and discussing with SRC, I got an impression of this proposed project with the following major concerns:

1) Feasibility and practicability issues vs. research and methodology issues

Among the practice and research issues involved in this project in both physical modeling techniques and parallel algorithm design, this project puts much more emphasis on effective and practical capability and evaluation of the parallelized existing code on a existing massively parallel system to solve real large scale problems.

2) Scalability vs. portability

The announcement indicates little attention on portability of the software development and corresponding architecture. Moreover, it seems they are trying to, through this project, evaluate which specific (existing or future) architectures are best suited to solve large real CEM industry applications. However, scalability issue appears throughout the announcement. One of the key evaluation criteria for the success of Phrase 1 is if the code both performs well in solving benchmark cases(small scale with matrix size between 3,000x3,000 - 5,000x5,000) and scales well to solve full-target real case(large scale with matrix size around 50,000x50,000 - 60,000x60,000).

3) advanced CEM modeling techniques vs. efficient parallel algorithms design

Accuracy (and stability) of the modeling in parallel implementation is the first concern of the project, which in large is related to the CEM modeling techniques. Partial reason is that the traditional computational limitations of most CEM models are first in memory capacity
and second computational speed due to large matrix size. The parallel algorithm design of CEM in large relies on linear algebra and data decomposition methodology. Here they are more interested in investigating effective parallel implementation for various advanced CEM techniques with high accuracy and stability.

3. Technical Approaches*

We have discussed two alternative approaches for this project:

1) Target Machines: CM5, Intel machines, IBM SP-1.
Message passing library: PVM
Advantages: Highly portability across the three platforms, and even to heterogeneous computing environment (networked workstations and MPP machines) which is a potential way to solve the memory limitation problem for large application case (I'll discuss this later). Good to compare computational and parallel implementation issues of CEM on different architecture MPP systems and find out which one is best suited for this application and which are not.

Disadvantages: The only MPP system to which PVM so far is ported is Intel iPSC/860, despite CM5 and IBM SP-1 claimed to support PVM on their machines. I have contacted the PVM developers about this issue and was told it is unlikely that the themselves are to start porting PVM on CM5 and IBM SP-1 in short time. Due to the nature of public domain software, there is no system technical support for PVM and (as I know and from other PVM users) PVM is still not in truly stable period, compared with those commercial products.

2) Target Machines: CM5(Phrase 1 & 2 working machine), Intel machines (Only Phrase 2), IBM SP-1(Only Phrase 2).

Message passing library: CMMD (Phrase 1 & 2), PVM(only Phrase 2).

Another alternative to start the Phrase 1 work is to use CM5 and CMfortran+CMMD to implement the existing Fortrann77 code, based on the above observations and the facts (or advantages) below:

. the time frame for Phrase 1 is relatively tight

. most of the effort in Phrase 1 is parallel algorithms design, coding and debugging which require intensive use of a working machine(an in-house machine is best to manage and speed-up the project progress in such a tight time frame and an application project)

. scalability is more important than portability in the Phrase 1, while the latter can be investigated in the Phrase 2 when hopefully PVM will be available on all the 3 platforms.

. It is really easy to port programs written in one message passing library (eg. CMMD) to programs in another library (eg. Intel's NX library) and CMMD supports most commonly used message passing paradigms(block, nonblock, and active message passing, host-node and hostless, multiple parallel I/O mode, interrupts and polling, etc.).

Disadvantages:

. portability becomes a non-major concern in the Phrase 1.

. several non-portable implementation techniques specific to CM5 architecture (will be discussed in technical discussion section later) will be either difficult or inefficient to port to other platforms, though without loss of generality in terms of methodology.

The reason choosing CM5 rather than iPSC/860 is explained below:

CM5's advantages:

. NPAC's in-house CM5 with flexible allocation time and good system support (even good expertised users at NPAC) which will greatly convenient the code development which is critical in Phrase 1.

. The modest system configration(machine and memory size, data vault, vector units, etc.) good enough to investigate computational issues for Phrase 1, including scalability, piplelined block algorithm design, modest matrix equation solver, modest benchmark production runs, etc.

. So far there has been no reported experiment on CM5 to use the techniques we plan to use for the large scale CEM application, while several works have been investigated on the Intel Touchstone architecture for the similar application. Our technical approach is "new" on CM5 architecture.

. The memory hierarchy on CM5 is very suitable to apply our approach, including node-level data parallel for matrix manipulation and global-level parallel message passing, pipe-lined LU block factorization, a off-shell block Gaussian elimination technique and an overall optimized pipelined computation and communication. So is its high-bandwidth balanced internal networks which are important to facilitate a efficient pipelined scheme.

. The high scalability of CM5's universal architecture and technology.

CM5's disadvantages(or iPSC's advantages):

. It is not easy, in the message passing program, to achieve the advantages of high performance hardware (e.g. Vector Units processing) on CM5. The CMMD program must be carefully programmed to obtain high efficiency.

. Unlike on Intel iPSC/860, Blas, whose routines will be intensively used in the parallel implementation, is not assembly coded.

. A majority of notable works in parallel CEM have been on Intel machines. A key previous work we'll apply to our CEM, the block LU decomposition algorithm design of NPAC's benchmark project, was implemented on iPSC/860, though it is optimistic to port to CM5 with high efficiency. The message passing implementation of our EM application(Lu's work) was also done on iPSC/860.

The reason to use CMfortran+CMMD:

So far the only way on CM5 in message passing program to make use of the vector-units is programming in CMfortran+CMMD. This is good for our application because most of the computations local to nodes are matrix manipulations, including block matrix filling, block LU factorization and block Gaussian elimination. It is also expected to be good for our pipelined algorithms from both local level and optimized global level.

Summary of technical approaches proposed:

The application can be abstracted as the following major components:
(% shows the projected percentage of computational time in sequential code)
. Fill the impedance matrix (40%)
. a linear solve for the matrix equation (50%)
  . LU factorization (40%)
  . Matrix solve (forward elimination and backward substitution) (10%)
. calculate field (10%)

Phrase 1:

. CM5, CMfortran+CMMD (NPAC's in-house CM5 system)

. Scalable data decomposition scheme, scaled and optimized by the scalable problem size and machine size. The scheme will be determined by a block matrix filling algorithm.
. Block algorithms, including block matrix filling, block LU factorization and block matrix solve.
. make efficient use of the CM5's memory hierarchy (node registers, vector-unit memory or/and Data Vault).
. Pipelined algorithms for different stages block Gaussian Elimination and possibly include the matrix filling.
. A off-shell block matrix filling and block Gaussian Elimination scheme (a kind of virtual memory scheme) utilizing CM5's parallel mass storage system Data Vault for large-scale problem whose memory requirements exceed the physical memory limitation. This scheme will only be considered in domain decomposition and algorithm design of Phrase 1 as required by the scalability concern. The real implementation will be done in Phrase 2.

Phrase 2:
. Port the software of Phrase 1 to other two platforms, namely iPSC/860 and IBM SP-1, using either PVM(if the library is available and stable on those machines at that time) or their primitive message passing libraries.
. evaluate three more schemes to solve the memory limitation problem:
. the off-shell virtual memory scheme mentioned above on modest CM5 system
. scale the software to a 1024 CM5 system
. a heterogeneous processing scheme distributing the memory requirements to networked MPP systems and/or workstations (in PVM).
. investigate more promising analytical techniques and/or software developed under the separate NASA NRA effort in our parallel implementation.

**: Detailed approach description is being prepared and will be sent under separate mail. Please give your opinion of this planned approach as it is somewhat different from the original discussion.*