Molecular Self-assembly for Nanoscale Spatial Computation

Chris Dwyer, Ph.D.

Dept. Electrical and Computer Engineering, Dept. Computer Science

Duke University, Durham NC 27708

Understanding molecular scale phenomena is a critical component of many scientific disciplines. The ability to retrieve nanoscale information from inside macroscale systems is particularly useful in biological fields where the diversity of molecular components and interaction dynamics within a cell make it difficult to monitor and quantify the underlying processes. Current methods rely on custom designed molecules—called molecular probes—that alter their observable properties to acquire real-time information about nanoscale phenomena.

Molecular probes are important members in the biological scientist's tool box, however they generally function as standalone sensors. Furthermore, their use requires costly equipment, highly specialized training and experiments that often span several days. These limitations prevent the application of molecular probes in monitoring complex biological processes with low cost. For example, at-home early disease detection could be achieved with a low-cost device capable of monitoring important markers of bioactivity and cellular health, such as concentrations of specific proteins or small messenger RNA (mRNA) molecules. The challenge is to develop techniques that provide low cost, efficient monitoring of complex molecular scale biological processes.

Computing is often used to monitor complex processes or automate tasks that require expert training. However, biological scale computing represents a new domain for computing with very different constraints from traditional computing systems. Although a CMOS processing core connected to bio-sensors could read and process chemical information, it would not be able to automate molecular probe applications because of its large aggregate size.

This work investigates a system of resource constrained nodes which must interact to solve a computational and sensing task. Much like any other spatial computation, the resource constraints on any given node have important implications on the architectural design of the system. Further, the ability to create nanoscale nodes which can freely diffuse through solution suggests an opportunity to fuse computation and sensing at length scales which are relevant to biological processes.

Within this context, an important system parameter is the period of time between node interactions (i.e., diffusion time) which is proportional to $\langle X \rangle^{2} \cdot D^{-1}$, where <X> is the average node-node separation and D is the diffusivity of a node. At the time of nearest approach the nodes could exchange information. Further, the period between contacts is determined (or limited) by the concentration and diffusion rates of the nodes—*diffusion limited computation* (DLC) describes this kind of spatial computation. For nodes to successfully collaborate on larger problems it is important to balance the size of the node (i.e., D), the concentration of nodes (i.e., <X>), and the available memory for computation and storage between interactions.

Diffusion limited computation could be used to compute the average binding rate of a set of analytes by incorporating an external time base. Information about the binding rate can be used to infer local concentrations of these molecules which are important markers of bioactivity and cellular health [1-3]. The challenge today is to perform this averaging at the nanoscale over a large set of possible proteins (e.g., about 4x10^{5} unique proteins can be found in any individual human cell). A distributed set of nodes, each designed to detect a subset of the total protein set, could employ diffusion to sample and average protein concentrations over a large observation window to track overall protein expression.

Similar to any computing system, the architecture must adapt to the application and resource constraints. Figure 1 shows the abstract relationship between node complexity and cooperativity required to achieve satisfactory application performance. An

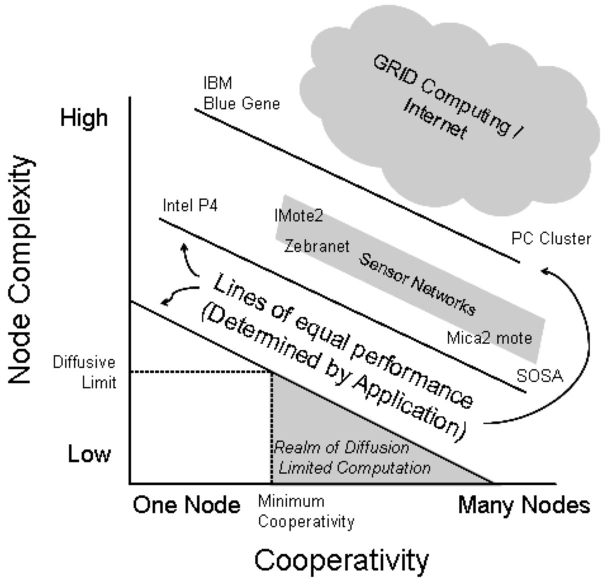

Figure 1: Complexity vs. cooperativity to achieve application performance.

isoperformance line is drawn between points in the space that can achieve the same performance with a given application. For example, SOSA [4] can achieve performance similar to a scaled Intel P4 on matrix multiplication, thus they are on the same isoperformance line, by trading complexity for cooperativity among large collections of nodes. Costs bound the complexity and cooperativity of any system and thus, some applications may map well to DLC. Note that DLC applications demand a minimum cooperativity due to the limited diffusion of complex nodes (i.e., the diffusive limit). Sensor networks and grid computing are shown to distinguish DLC from existing methods. *The important distinctions are that the nodes are small (nanoscale), heavily resource constrained, diffuse through solution, and actively interact with the environment.*

This work explores the design, synthesis and architectural implications of integrating computation and molecular probes to form a self-assembled nanoscale system of nodes capable of spatial computation.

[1] D. Endy, "Foundations for Engineering Biology," *Nature*, vol. 438 (7067), pp. 449-453, 2005.

[2] A. P. Arkin and D. A. Fletcher, "Fast, Cheap and Somewhat in Control," *Genome Biology*, vol. 7 (8), pp. 114.1-114.6, 2006.

[3] I. Tinoco, K. Sauer, and J. C. Wang, *Physical Chemisty: Principles and Applications in Biological Science*. Upper Saddle River, NJ: Prentice Hall, 762 pgs., 1995.

[4] J. P. Patwardhan, V. Johri, C. Dwyer, and A. R. Lebeck, "A Defect Tolerant Self-organizing Nanoscale SIMD Architecture," Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems, 2006.