Analysis of ’Web-Mesh’ Application in Multicore Architecture

Dr. Abu Asaduzzaman                 Danny Nguyen
Wichita State University            Wichita State University

April 27, 2012
Abstract

This report will introduce the idea of the self-titled “Web-mesh” topology, and conduct a basic analysis of its different properties, especially in regards as a network topology and mostly focused on its possible benefits to multicore architecture.

The web-mesh is a topology such that the interconnects between nodes make the overall shape that of a spider’s web. As far as network topologies go, this layout can be thought of as a mix between one star and multiple ring topologies. In computer architecture, specifically in multicore applications, we hope that the web-mesh will provide some perks over the 2-D mesh in an easily implemented design.
Introduction

The web-mesh works on the idea of a master node through which all traffic through the network will first go to this master. The idea is that the physical distance between itself and its immediate neighbours should always remain constant, which will aid in consistency of latency measurements and so forth. If this distance is some constant $r$, then the radius of the first ring will be $r$, and there will only be room for 6 nodes on this first level. This leads to 6 so called primary-lines which will be the only means of travel between levels.

This means ultimately the topology looks like a series of concentric circles which surround the master node, each circle representing a 'level' and each level having a difference of 1 $r$ between its neighbouring levels.

![Diagram of full level 2 web](image)

Figure 1: Example of a full level 2 web. Each dot represents a computing node, each circle is the corresponding level of connected nodes, and the dotted lines are the primary-lines between levels.

Survey

A brief survey of the concept on the internet and in literature shows that 2-D dies are the most popular means of processor design today, and as far as reported, rectangular meshes are the only means of 2-D implementation[3]. The buzz for future designs is focused on the development of 3-D dies[1], but maybe it’s not necessary to jump to that next level of complexity for some applications of processing, in which a web mesh layout would meet the goals with an easier implementation.
The 2D-Mesh

For comparison purposes, the 2D-mesh being measured against the web-mesh will have the limitations of: 1) Being square 2) Having an odd number of nodes in both width and height; a.k.a. it has a center node—a good position for a master server.

Maximum Number of Nodes

We can visually determine that the dimensions of a square-mesh where the level number is \( n \), must be \((2n + 1)x(2n + 1)\). For a level 1 2D-mesh, the surrounding square must have 3 nodes on a side. This means the mesh has the dimensions of a 3x3 square. A level 2 square must add two extra nodes to a border in order to enclose the previous level and still make a square. By inspection, it is true that every level makes a square with two more nodes on a side than the level before it.
Therefore, the maximum amount of nodes in a 2D-mesh is simply the area of a square with the series of odd numbers starting at 3 as its width and height:

\[ \text{Nodes}_{\text{max}}(n) = (2n + 1)^2, \{n \in \mathbb{Z}|1 \leq n \} \]

**Number of Hops to Farthest Node**

By inspection, the furthest node from the center, due to the symmetry of the mesh, will be the four corner nodes. The only way to reach a corner node is to travel through an equal amount of nodes horizontally and vertically. For a level 1 mesh, a total of two jumps must be made to reach the corner; for level 2, 4 jumps; for level 3, 6 jumps etc.

Since the mesh is always a perfect square, we can break each quadrant into 4 smaller squares, each with dimensions of \((nxn)\). Therefore, the number of jumps needed to reach the furthest node must be

\[ \text{Jumps}_{\text{max}}(n) = 2n, \{n \in \mathbb{Z}|1 \leq n \} \]

**The Web-Mesh**

We now begin to develop the properties of the web-mesh in order to compare it to the standard 2D-mesh laid out in the last subsection.

**Maximum Number of Nodes**

We see that with a set distance of \(\approx r\) between each node, the concentric circle of connections which form each level have corresponding radii of
nr, \{n \in \mathbb{Z}|1 \leq n\}, where \(n\) is the level number of the web-mesh. This means that the first level, by the properties of a circle, only has room for \(\approx 6\) nodes; and the second level has room for 12; the third, 18 etc.

This means that the total amount of nodes\(^1\) in a full mesh, as a function of its number of levels \(n\) is

\[
\text{Nodes}_{\text{max}}(n) = \left(\sum_{i=1}^{n} \lfloor 2\pi i \rfloor \right) + 1
= \lfloor 2\pi \rfloor \frac{n(n+1)}{2} + 1
= \lfloor \pi \rfloor n(n+1) + 1
\approx 3n(n+1) + 1
\]

**Number of Hops to Farthest Node**

For the six regions divided by the primary lines, assuming that all connections are working, the farthest distance from the center is when the master node must communicate with a node half-way between a primary line, on the furthest level ring.

\[
\text{Midpoint}(n) = \left\lfloor \frac{1}{3} \pi n \right\rfloor
\]

\[.\] \[\therefore \text{Jumps}_{\text{max}} = n + \left\lfloor \frac{1}{2} \pi n \right\rfloor \approx n + \left\lfloor \frac{1}{2} n \right\rfloor\]

That is, for a message to reach the midpoint with this topology must travel along a primary-line \(n\) hops between nodes, and then travel through the hops to reach the node at the midpoint of the level. Notice that the hops only increase over the previous level on every even level, since the web-mesh only gains a new discrete midpoint on these levels.

Another idea to entertain specifically for jumping between nodes, is the addition of extra primary-lines where there is symmetry, such as between the midpoint nodes of every even level. This means that only the even levels can benefit from this design, and that the max jumps which can be made on the even levels, is the jumps needed to reach that level plus the jumps needed to reach the midpoint between the primary-line and the secondary-line.

\(^1\)The floor function introduces an error of roughly 5%
Due to the symmetry, we see by inspection that for every even level, instead of increasing the maximum jumps by 1, the max jumps is the same amount as the odd-level before it. This means that the rate of increase in jumps is half that of just having 6 primary axes of movement. This can be represented with the piecewise function

\[
Jumps_{\text{max}} = \begin{cases} 
  n + \lfloor \frac{1}{2}n \rfloor & n = 2k + 1 \\
  n + \lfloor \frac{1}{4}n \rfloor & n = 2k 
\end{cases}
\]

Does this mean that the trend continues as we add more levels of interconnection? Undoubtedly, adding interconnections between corresponding layers of symmetry—for example, between every layer starting at \( n=3 \), to \( n=6 \) etc.—does reduce the amount of jumps, but the trend of reduction is not a linear factor of \( \frac{1}{2} \). First of all, we see that each new line of connections only affects the number of hops on levels which are factors of the source of the
new line; e.g. the demonstrated scheme only affects even levels (factors of 2). If all factors of 3 are then connected, the piecewise function for maximum jumps would have a case for levels that are not factors of both 2 and 3, a case for factors of only 2, a case for factors of only 3, and finally a case for factors of 2 AND 3.

This trend shows that which such a connection scheme focused on symmetry, will only show improvements on levels which have the new primary-lines’ sources as factors of its level number $n$; that is, if level 2 is connected to every even level, as demonstrated; and level 3 nodes are connected to every third level in a similar fashion; and level 4 nodes (besides the node already connected to level 2) are connected to every fourth level, only levels 6, 8, 9, 10, 12, 14,... will see an improvement over just having a primary line. Since the best symmetry is between the even levels, if we assume they are all connected to take advantage of this, the amount of levels unaffected by interconnected levels are those which have a level number that is a prime number that is not a factor of the connected levels before it.

Luckily prime numbers only get more and more rare as $n$ increases; however, even at $n = 10$, if all prime factors have primary-lines added to them, level 7 will essentially be a dead layer where it’s extremely undesirable to place a node there as the system designer because 1) It’s a prime level, meaning it takes the base max jump rate to reach a midpoint on the level $n + \left\lfloor \frac{1}{2} n \right\rfloor 2$) The size of the network is not large enough to warrant the next factors of 7: 14, 21, 28, 35,...
Figure 5: Plot of Max Node Capacity. The higher trend-line is the max nodes a 2D-mesh can hold, while the lower is the max nodes for a web-mesh.

Figure 6: Plot of Max Jumps. The higher trend-line is the max jumps to a node for a 2D-mesh, while the lower is the max jumps for a web-mesh.
Power Distribution & Heat Dissipation

Since there are roughly 75% the amount of nodes in a web-mesh compared to a similar 2D-mesh, occupying an area about 75% the size; the assumption would be that the power dissipation is proportional by the same amount. While this turns out to be true, one may worry that the heat distributed by each node may be more intense in a web-mesh layout than in a 2D-mesh.

Let’s assume that each node in the whole die in a web-mesh configuration is dissipating heat at a constant rate at some load proportional to the power dissipated by a resistive element:

\[ P \propto I^2 R \]

If we assume each node as a dimensionless point, the density of the power dissipated, \( \rho \), by the node a distance \( r \) away must be proportional to the surface area of a sphere with radius \( r \) concentric to the node:

\[
\rho(r) = \frac{P}{dV(r)} = \frac{I^2 R}{4\pi r^2}
\]

\[ \therefore P = \int_D \rho(r) dA \]

Since the web-mesh is only two-dimensional, let us say that the surface area of the circle at the point a radius \( r \) from the node is approximately equal to \( dA \), so the intensity of the heat being delivered a distance \( r \) from the node is proportional to the inverse of \( r^2 \):

\[ P(r) \propto \frac{1}{r^2} \]

This means that the total heat dissipation at any point on the plane of nodes is the superposition of the heat dissipation at that point by each node in the mesh. Now, the distance between each node along a primary axis is obviously the constant ‘\( r \)’ designated at design, while the distance between any neighbor nodes on a n-th level ring is the chord distance[4], \( D_n \), between the points on an equivalent circle of radius \( nr \).

\[ D_n = 2rn \sin \frac{\theta}{2} = 2rn \sin \frac{1}{2n} \]
When \( n = 1 \), the chord length is approximately 0.95\( r \), and by \( n = 3 \), the length is 0.99\( r \), so approximating the nearest neighbor distance as the constant \( r \) is a reasonable assumption\(^2\).

By inspection, we see that the distance between nodes off the primary lines from off-primary nodes on neighboring levels are greater than the constant distance \( r \), but like the chord distance, as \( n \to \infty \) the distance also approaches \( r \). That is, the heat intensity at some point in the web-mesh is approximately equal to the heat intensity in a similarly set up 2D-mesh, just with 25\% less nodes. Since the relationship between the amount of heat dissipation and total power from nodes linear (factor of 0.75), the heat dissipation at equivalent geometric points in a web-mesh set-up in a computing environment should just be 75\% the amount of the equivalent 2D-mesh; this is not surprising, since there are 75\% the amount of nodes supported by a web-mesh compared to the 2D-mesh.

![Figure 7: Sample behavior of heat intensity w/ edge node at 0 for both meshes.](image)

**Tables**

<table>
<thead>
<tr>
<th>( n )</th>
<th># of Nodes</th>
<th>Heat Dissipation (W/m(^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2D</td>
<td>Web</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>3</td>
<td>49</td>
<td>37</td>
</tr>
<tr>
<td>4</td>
<td>81</td>
<td>61</td>
</tr>
<tr>
<td>5</td>
<td>121</td>
<td>91</td>
</tr>
</tbody>
</table>

\(^2\)\( \lim_{n \to \infty} D_n = \lim_{n \to \infty} r 2n \sin \frac{1}{2n} = r \)

\(^3\)Power sampled with \( I^2R = 1 \), and \( r = 1 \)
Conclusion

Setting out to see any possible benefits that a web-mesh layout could provide over the traditional 2D-meshes used today, we’ve found that the the number of nodes required to be travelled in a worst case scenario in a mesh to always be less than the worst case scenario for the center of a mesh trying to communicate with a corner node. This could be a very big advantage for using a web-mesh topology over a 2D-mesh, since the interconnection layout can be visually mapped in a fashion that doesn’t require drawing ridiculous hypercubes. A good handle on the fundamental theorem of arithmetic will essentially answer every question about which layers will be interconnected with what. In situations where there isn’t an emphasis on having as many nodes populate a topology as possible, e.g. implementing an n-level web-mesh without enough nodes to completely populate each level, would very likely see massive latency benefits if arranged such that each node is in a position as close as possible to all of the primary-lines.

As far as physical implementation goes—like in a multicore processor—if both meshes have the same amount of cores, the total power consumption will be the same, of course, and we expect that the heat dissipation rate will be very near equivalent at similar locations between nodes. However, if the web-mesh is organized in a fashion such that the nodes are located manually spread out and clumped near primary lines instead of fully populating prime levels, the processor could see a non-negligible amount of decrease in latency on message passing due to the cut in amount of jumps made by a factor of at least \( \frac{1}{2} \) depending on what node needs to be contacted and its location in the mesh.
Bibliography


