Network-aware Container Distribution System Project Proposal

1. Vision and Goals Of The Project:

"Network-aware Container Distribution System" is a project which aims at creating a solution to use idle networking resource of a datacenter to distribute data/application inside that datacenter (private cloud).

Users/Personas Of The Project

The Project, if successful, will be used as a part of Jisto's private cloud solution. The project targets to deliver data/application inside the datacenter with a maximum network utilization without affecting the other processes. The user's of the project will likely to be those companies who use Jisto's solution to build private cloud on their idle computing resources.

2. Scope and Features Of The Project:

- We expect the solution will have the scalability between 60,000 - 1,000,000 nodes and be able to run a large cluster test and get performance with a number of different network loads.
- The System should be able to aware of the impact on the network and to react accordingly.

3. Solution Concept

Below is a description of the system components that are building blocks of the architectural design:

Global Architectural Structure Of the Project and a Walkthrough:

Below is a description of the system components that are building blocks of the architectural design:

- Network Monitoring Process: A process monitors the network bandwidth utilization of the datacenter in real-time. This is the most crucial part of the project, everything else in the project operates base on the data from this process.
- Docker: The container system we are going to use to wrap up data/applications to make them easily shippable.
- Node.js: Javasript platform for easily building fast, scalable network applications.Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.
- Bash Script: Required in order to automate the process of deploying the system.

Design Implications and Discussion

Key design decisions and motivation behind them.

- Node.js: The key point of this project is to monitor the network status of the datacenter in real-time and response to the result. Node is perfect for this because it does most I/O tasks using *non-blocking* techniques. So we can create callback function that will be executed in response to events.
- Docker: Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications. We will need this container to make the data/application easily shippable.

Accepting criteria:

Successfully send data wrapped in Docker to more than 2 machines that is under a busy network condition and doesn't interfere with the networking tasks the are already going.

Release Planning

Release #1 (due by Week 5):
Successfully simulate a busy network condition between two machines and get a third machine distribute file to both of them

Release #2 (due by Week 7):
Limit the speed of the file transmission.

Release #3 (due by Week 9):
Implement the network monitoring process and make it feed data to the file-sending process.

Release #4 (due by Week 9):
Refine the system, try to scale-up, test the system in actual datacenter.

Release #5 (due by Week 13):
Keep adding nodes to the system, test performance