**Assignment 10 - A Cloud Application**

**Name: Shweta Pathak**

**UTA ID: 1001154572**

**Net ID: ssp4572**

**Section: 1.00 pm to 3.00 pm**

**Description:**

This application uses Docker on EC2 instance of Amazon web services. The functionality implemented is as follows:

1. **Dynamic Visualization - Stacked bar-charts on time scale:** Displays a stacked bar chart based on the number of messages per day. Also there is a feasibility to select the few dates and check the statics of the messages per hour on that day.
2. **Dynamic Visualization - Collision Detection:** This is a dynamic collision detection that changes as the mouse moves.

**Reference:**

1. <http://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html>
2. <http://bl.ocks.org/anupsavvy/9513382>
3. <http://mbostock.github.io/d3/talk/20111018/collision.html>

**Code:**

**Dockerfile:**

FROM ubuntu:12.04

# Install dependencies

RUN apt-get update -y

RUN apt-get install -y git curl apache2 php5 libapache2-mod-php5 php5-mcrypt php5-mysql

# Install app

RUN rm -rf /var/www/\*

ADD src /var/www

# Configure apache

RUN a2enmod rewrite

RUN chown -R www-data:www-data /var/www

ENV APACHE\_RUN\_USER www-data

ENV APACHE\_RUN\_GROUP www-data

ENV APACHE\_LOG\_DIR /var/log/apache2

EXPOSE 80

CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]

**Index.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Assignment 10</title>

<script type="text/javascript" src="http://code.jquery.com/jquery-latest.min.js"></script>

<script type="text/javascript" src="http://d3js.org/d3.v3.min.js"></script>

<script src="//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css">

</head>

<body>

<div>

<h3> Dynamic Visualization - Stacked bar-charts on time scale.</h3>

<form action="" method="post" enctype="multipart/form-data">

<input type="submit" name="Submit" value="Get Chart" /><br><br>

<h3> Dynamic Visualization - Collision Detection.</h3>

<input type="submit" name="Collision" value="Collision Detection" />

</form>

<?php

if ($\_SERVER["REQUEST\_METHOD"] == "POST"){

if (isset($\_POST['Submit']))

{

header('Location: barchart.php');

}

if (isset($\_POST['Collision']))

{

header('Location: collision.php');

}

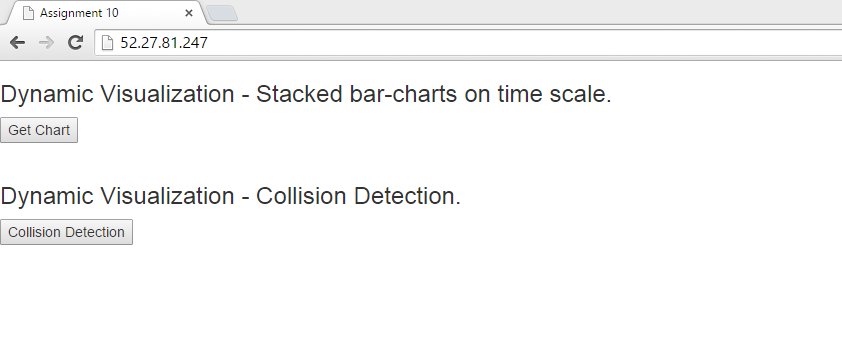
}

?>

</body>

</html>

**Output:**



**Barchart.php:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Assignment 10</title>

<script type="text/javascript" src="http://code.jquery.com/jquery-latest.min.js"></script>

<script type="text/javascript" src="http://d3js.org/d3.v3.min.js"></script>

<script src="//netdna.bootstrapcdn.com/bootstrap/3.1.1/js/bootstrap.min.js"></script>

<link rel="stylesheet" href="//netdna.bootstrapcdn.com/bootstrap/3.1.1/css/bootstrap.min.css">

<style>

.axis path,

.axis line {

fill: none;

stroke: black;

shape-rendering: crispEdges;

}

.axis text {

font-family: sans-serif;

font-size: 11px;

}

.dot {

stroke: #000;

}

.legend {

padding: 5px;

font: 10px sans-serif;

background: yellow;

box-shadow: 2px 2px 1px #888;

}

</style>

</head>

<body>

<h3> Dynamic Visualization - Stacked bar-charts on time scale.</h3>

<br><br>

<div>

<div class="btn-group pull-right">

<button type="button" class="btn btn-primary dropdown-toggle" data-toggle="dropdown">

Messages per hour <span class="caret"></span>

</button>

<ul class="dropdown-menu" role="menu">

<li><a class="m" value="2014-02-19" href="#">2014-02-19</a></li>

<li><a class="m" value="2014-02-20" href="#">2014-02-20</a></li>

<li><a class="m" value="2014-02-21" href="#">2014-02-21</a></li>

<li><a class="m" value="2014-02-22" href="#">2014-02-22</a></li>

<li><a class="m" value="2014-02-23" href="#">2014-02-23</a></li>

</ul>

</div>

<div id="mbars">

</div>

</div>

<script type="text/javascript">

var w = 600; //width

var h = 500; //height

var padding = {top: 40, right: 40, bottom: 40, left:40};

var dataset;

//Set up stack method

var stack = d3.layout.stack();

d3.json("mperday.json",function(json){

dataset = json;

//Data, stacked

stack(dataset);

var color\_hash = {

0 : ["Invite","#1f77b4"],

1 : ["Accept","#2ca02c"],

2 : ["Decline","#ff7f0e"]

};

//Set up scales

var xScale = d3.time.scale()

.domain([new Date(dataset[0][0].time),d3.time.day.offset(new Date(dataset[0][dataset[0].length-1].time),8)])

.rangeRound([0, w-padding.left-padding.right]);

var yScale = d3.scale.linear()

.domain([0,

d3.max(dataset, function(d) {

return d3.max(d, function(d) {

return d.y0 + d.y;

});

})

])

.range([h-padding.bottom-padding.top,0]);

var xAxis = d3.svg.axis()

.scale(xScale)

.orient("bottom")

.ticks(d3.time.days,1);

var yAxis = d3.svg.axis()

.scale(yScale)

.orient("left")

.ticks(10);

//Easy colors accessible via a 10-step ordinal scale

var colors = d3.scale.category10();

//Create SVG element

var svg = d3.select("#mbars")

.append("svg")

.attr("width", w)

.attr("height", h);

// Add a group for each row of data

var groups = svg.selectAll("g")

.data(dataset)

.enter()

.append("g")

.attr("class","rgroups")

.attr("transform","translate("+ padding.left + "," + (h - padding.bottom) +")")

.style("fill", function(d, i) {

return color\_hash[dataset.indexOf(d)][1];

});

// Add a rect for each data value

var rects = groups.selectAll("rect")

.data(function(d) { return d; })

.enter()

.append("rect")

.attr("width", 2)

.style("fill-opacity",1e-6);

rects.transition()

.duration(function(d,i){

return 500 \* i;

})

.ease("linear")

.attr("x", function(d) {

return xScale(new Date(d.time));

})

.attr("y", function(d) {

return -(- yScale(d.y0) - yScale(d.y) + (h - padding.top - padding.bottom)\*2);

})

.attr("height", function(d) {

return -yScale(d.y) + (h - padding.top - padding.bottom);

})

.attr("width", 15)

.style("fill-opacity",1);

svg.append("g")

.attr("class","x axis")

.attr("transform","translate(40," + (h - padding.bottom) + ")")

.call(xAxis);

svg.append("g")

.attr("class","y axis")

.attr("transform","translate(" + padding.left + "," + padding.top + ")")

.call(yAxis);

// adding legend

var legend = svg.append("g")

.attr("class","legend")

.attr("x", w - padding.right - 65)

.attr("y", 25)

.attr("height", 100)

.attr("width",100);

legend.selectAll("g").data(dataset)

.enter()

.append('g')

.each(function(d,i){

var g = d3.select(this);

g.append("rect")

.attr("x", w - padding.right - 65)

.attr("y", i\*25 + 10)

.attr("width", 10)

.attr("height",10)

.style("fill",color\_hash[String(i)][1]);

g.append("text")

.attr("x", w - padding.right - 50)

.attr("y", i\*25 + 20)

.attr("height",30)

.attr("width",100)

.style("fill",color\_hash[String(i)][1])

.text(color\_hash[String(i)][0]);

});

svg.append("text")

.attr("transform","rotate(-90)")

.attr("y", 0 - 5)

.attr("x", 0-(h/2))

.attr("dy","1em")

.text("Number of Messages");

svg.append("text")

.attr("class","xtext")

.attr("x",w/2 - padding.left)

.attr("y",h - 5)

.attr("text-anchor","middle")

.text("Days");

svg.append("text")

.attr("class","title")

.attr("x", (w / 2))

.attr("y", 20)

.attr("text-anchor", "middle")

.style("font-size", "16px")

.style("text-decoration", "underline")

.text("Number of messages per day.");

//On click, update with new data

d3.selectAll(".m")

.on("click", function() {

var date = this.getAttribute("value");

var str;

if(date == "2014-02-19"){

str = "19.json";

}else if(date == "2014-02-20"){

str = "20.json";

}else if(date == "2014-02-21"){

str = "21.json";

}else if(date == "2014-02-22"){

str = "22.json";

}else{

str = "23.json";

}

d3.json(str,function(json){

dataset = json;

stack(dataset);

console.log(dataset);

xScale.domain([new Date(0, 0, 0,dataset[0][0].time,0, 0, 0),new Date(0, 0, 0,dataset[0][dataset[0].length-1].time,0, 0, 0)])

.rangeRound([0, w-padding.left-padding.right]);

yScale.domain([0,

d3.max(dataset, function(d) {

return d3.max(d, function(d) {

return d.y0 + d.y;

});

})

])

.range([h-padding.bottom-padding.top,0]);

xAxis.scale(xScale)

.ticks(d3.time.hour,2)

.tickFormat(d3.time.format("%H"));

yAxis.scale(yScale)

.orient("left")

.ticks(10);

groups = svg.selectAll(".rgroups")

.data(dataset);

groups.enter().append("g")

.attr("class","rgroups")

.attr("transform","translate("+ padding.left + "," + (h - padding.bottom) +")")

.style("fill",function(d,i){

return color(i);

});

rect = groups.selectAll("rect")

.data(function(d){return d;});

rect.enter()

.append("rect")

.attr("x",w)

.attr("width",1)

.style("fill-opacity",1e-6);

rect.transition()

.duration(1000)

.ease("linear")

.attr("x",function(d){

return xScale(new Date(0, 0, 0,d.time,0, 0, 0));

})

.attr("y",function(d){

return -(- yScale(d.y0) - yScale(d.y) + (h - padding.top - padding.bottom)\*2);

})

.attr("height",function(d){

return -yScale(d.y) + (h - padding.top - padding.bottom);

})

.attr("width",15)

.style("fill-opacity",1);

rect.exit()

.transition()

.duration(1000)

.ease("circle")

.attr("x",w)

.remove();

groups.exit()

.transition()

.duration(1000)

.ease("circle")

.attr("x",w)

.remove();

svg.select(".x.axis")

.transition()

.duration(1000)

.ease("circle")

.call(xAxis);

svg.select(".y.axis")

.transition()

.duration(1000)

.ease("circle")

.call(yAxis);

svg.select(".xtext")

.text("Hours");

svg.select(".title")

.text("Number of messages per hour on " + date + ".");

});

});

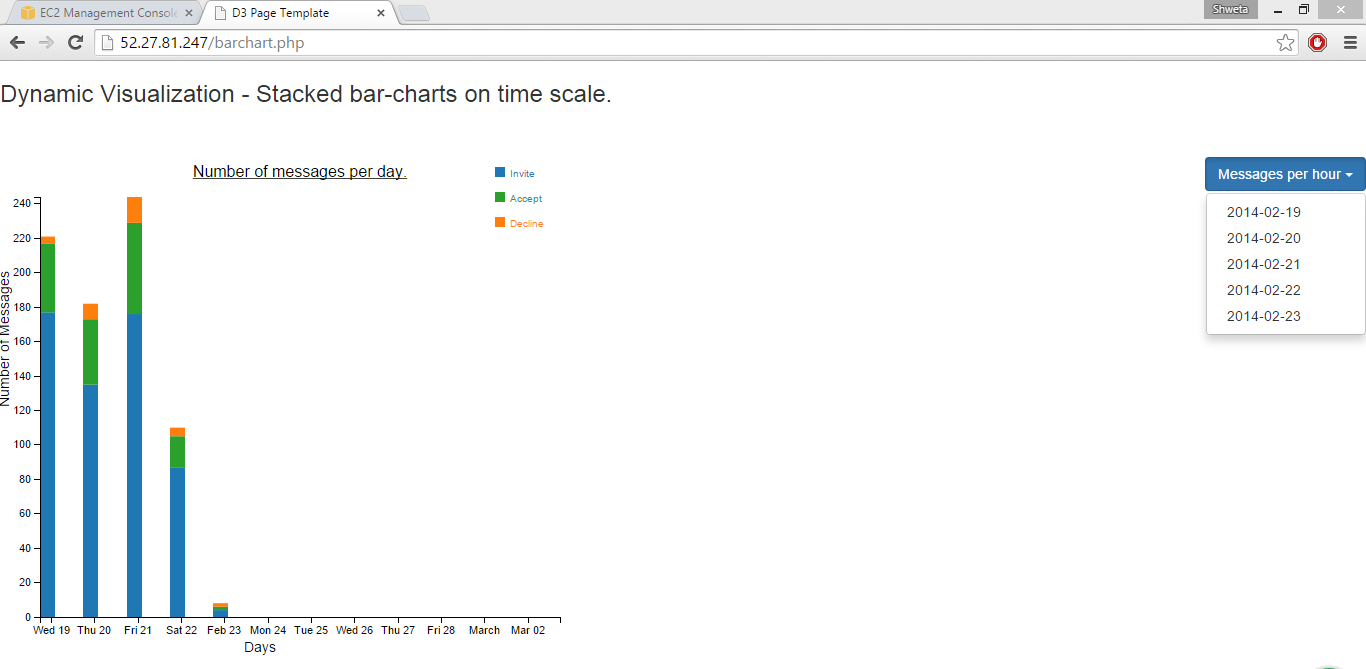
});

</script>

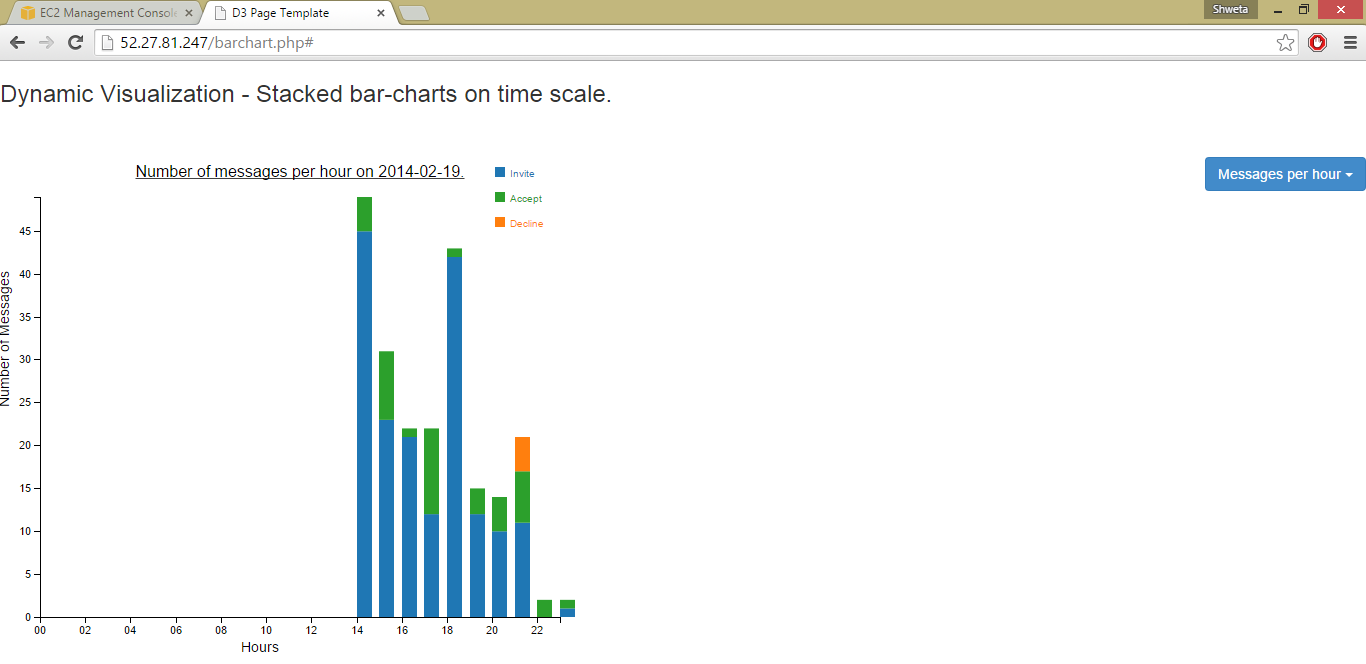
</body>

</html>

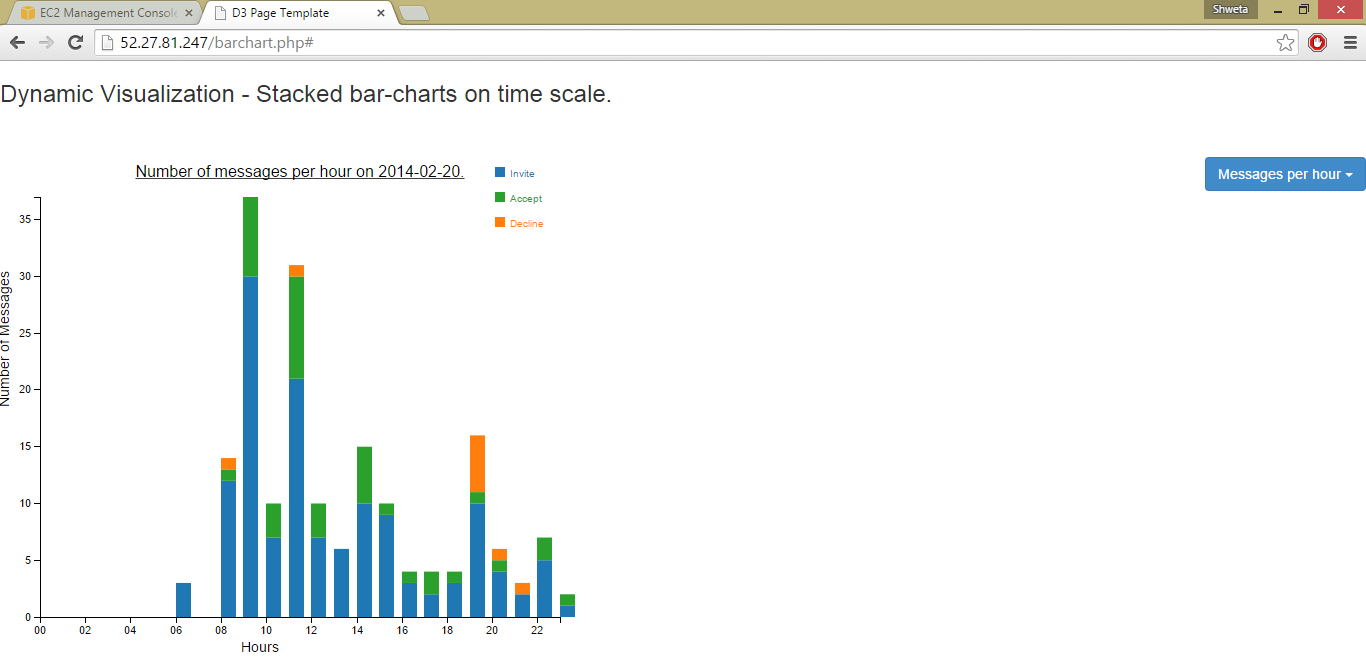
**Output:**



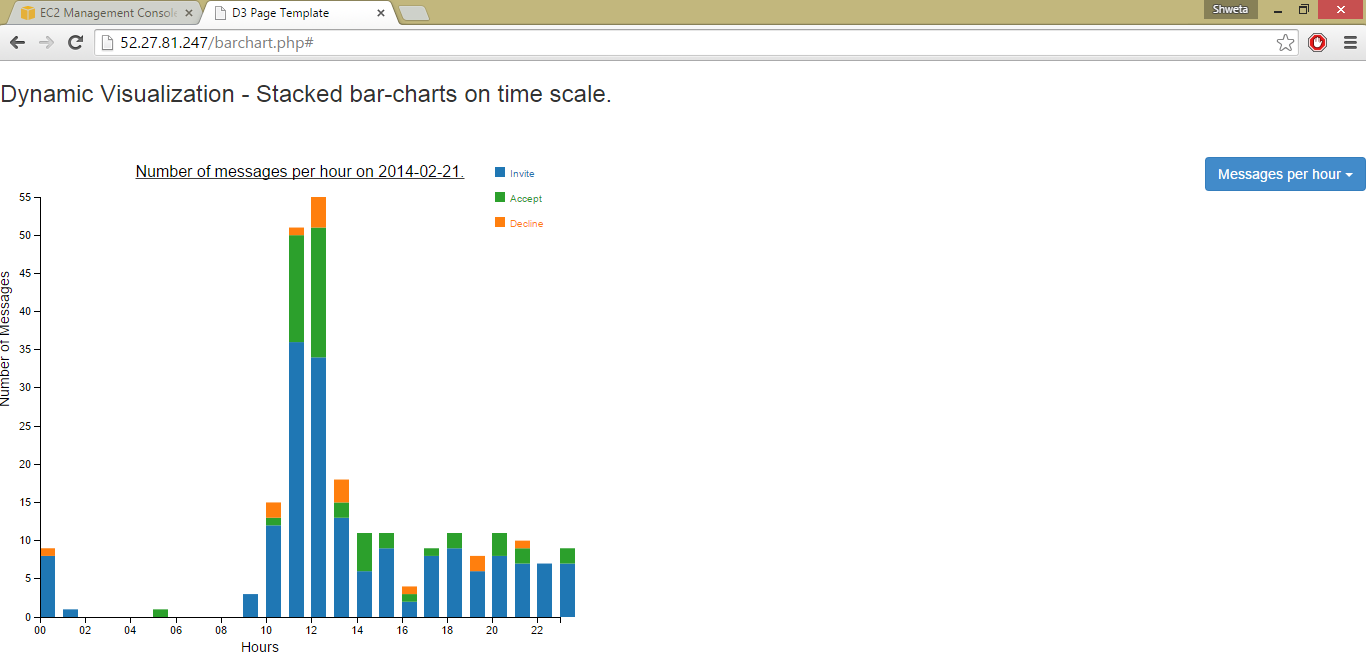
**For 2014-02-19:**



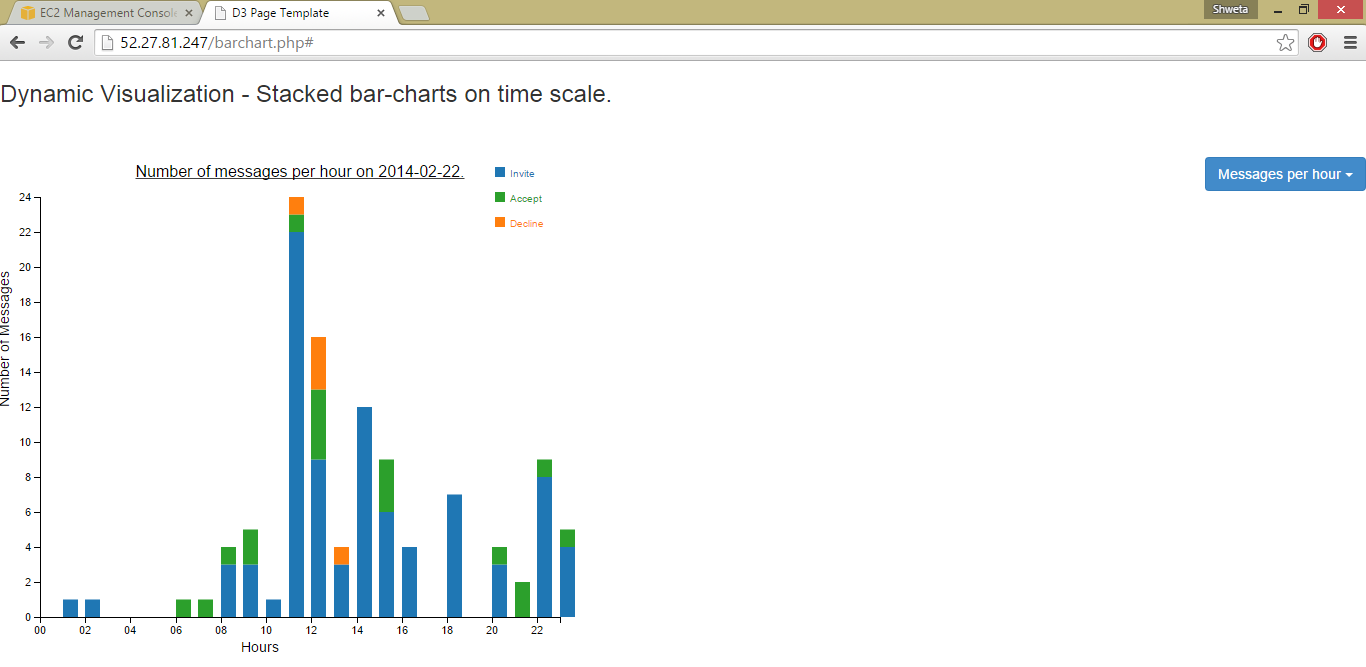
**For 2014-02-20:**



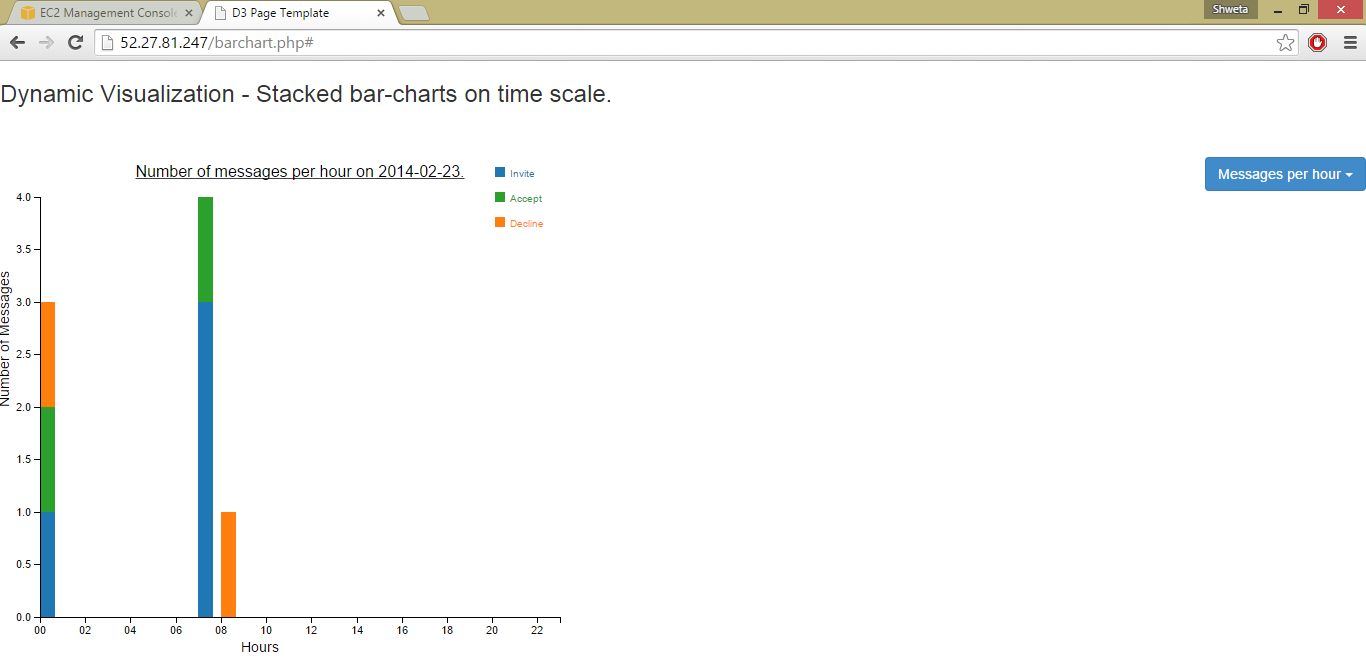
**For 2014-02-21:**



**For 2014-02-22:**



**For 2014-02-23:**



**Collision.php:**

<!DOCTYPE html>

<html>

<head>

<title> Assignment 10</title>

<meta http-equiv="Content-Type" content="text/html;charset=utf-8"/>

<script type="text/javascript" src="http://mbostock.github.io/d3/talk/20111018/d3/d3.js"></script>

<script type="text/javascript" src="http://mbostock.github.io/d3/talk/20111018/d3/d3.geom.js"></script>

<script type="text/javascript" src="http://mbostock.github.io/d3/talk/20111018/d3/d3.layout.js"></script>

<link type="text/css" rel="stylesheet" href="http://mbostock.github.io/d3/talk/20111018/style.css"/>

<style type="text/css">

circle {

stroke: #000;

stroke-opacity: .5;

}

</style>

</head>

<body>

<h3> Dynamic Visualization - Collision Detection. </h3>

<div id="body">

<div id="footer">

<div class="hint">move the mouse to repel nodes</div>

</div>

</div>

<script type="text/javascript">

var w = 1280,

h = 800;

var nodes = d3.range(200).map(function() { return {radius: Math.random() \* 12 + 4}; }),

color = d3.scale.category10();

var force = d3.layout.force()

.gravity(0.05)

.charge(function(d, i) { return i ? 0 : -2000; })

.nodes(nodes)

.size([w, h]);

var root = nodes[0];

root.radius = 0;

root.fixed = true;

force.start();

var svg = d3.select("#body").append("svg:svg")

.attr("width", w)

.attr("height", h);

svg.selectAll("circle")

.data(nodes.slice(1))

.enter().append("svg:circle")

.attr("r", function(d) { return d.radius - 2; })

.style("fill", function(d, i) { return color(i % 3); });

force.on("tick", function(e) {

var q = d3.geom.quadtree(nodes),

i = 0,

n = nodes.length;

while (++i < n) {

q.visit(collide(nodes[i]));

}

svg.selectAll("circle")

.attr("cx", function(d) { return d.x; })

.attr("cy", function(d) { return d.y; });

});

svg.on("mousemove", function() {

var p1 = d3.svg.mouse(this);

root.px = p1[0];

root.py = p1[1];

force.resume();

});

function collide(node) {

var r = node.radius + 16,

nx1 = node.x - r,

nx2 = node.x + r,

ny1 = node.y - r,

ny2 = node.y + r;

return function(quad, x1, y1, x2, y2) {

if (quad.point && (quad.point !== node)) {

var x = node.x - quad.point.x,

y = node.y - quad.point.y,

l = Math.sqrt(x \* x + y \* y),

r = node.radius + quad.point.radius;

if (l < r) {

l = (l - r) / l \* .5;

node.x -= x \*= l;

node.y -= y \*= l;

quad.point.x += x;

quad.point.y += y;

}

}

return x1 > nx2

|| x2 < nx1

|| y1 > ny2

|| y2 < ny1;

};

}

</script>

</body>

</html>

**Output:**

