

```
1: <?php
2:
3: /*
4:
5: Initialize Cluster - by Patent Perspectives, Inc.
6:
7: for 6 COTS servers (Linux 16.0x) DataNodes
8:
9: using Hadoop configured on AMD laptop
10:
11:
12: */
13:
14: $running = 'running';
15:
16: $commands = '
17: #!/bin/ksh
18:
19: safeRunCommand1() {
20:     typeset cmdnd="$*"
21:     typeset ret_code
22:
23:     echo cmdnd=$cmdnd
24:     eval $cmdnd
25:     ret_code=$?
26:     if [ $ret_code != 0 ]; then
27:         $hadoop fs -mkdir /user
28:     fi
29: }
30:
31: safeRunCommand2() {
32:     typeset cmdnd="$*"
33:     typeset ret_code
34:
35:     echo cmdnd=$cmdnd
36:     eval $cmdnd
37:     ret_code=$?
38:     if [ $ret_code != 0 ]; then
39:         $hadoop fs -mkdir /user/cfleshner
40:     fi
41: }
42:
43: safeRunCommand3() {
44:     typeset cmdnd="$*"
45:     typeset ret_code
46:
47:     echo cmdnd=$cmdnd
48:     eval $cmdnd
49:     ret_code=$?
50:     if [ $ret_code != 0 ]; then
51:         $hadoop fs -mkdir /user/cfleshner/folder
52:     fi
53: }
54:
55: ##command="$hadoop fs -test -e /user"
56: ##safeRunCommand1 "$command"
57:
58: ##command="$hadoop fs -test -e /user/cfleshner"
59: ##safeRunCommand2 "$command"
60:
61: $$command="$hadoop fs -test -e /user/cfleshner/folder"
62: $$safeRunCommand3 "$command"
63:
64: $hadoop fs -put -f 1Gb.dat /user/cfleshner/folder/blocks.dat
65:
66: ';
```

```
67:
68: $commands = trim ($commands);
69:
70: $commands = str_replace ('$hadoop', '/home/hadoop/hadoop/bin/hadoop', $commands);
71:
72: $y = 1;
73:
74: $z = 1;
75:
76: $quit = false;
77:
78: while ( !$quit ) # 4*25 is 100 GB of data - 1GB per script
79: {
80: {
81:
82: $x = 1;
83:
84: while ($x < 26)
85: {
86: {
87:
88: $folder = 'folder$y';
89:
90: $blocks = 'blocks$x';
91:
92: $pady = str_pad ($y, 3, '0', STR_PAD_LEFT);
93:
94: $padx = str_pad ($x, 3, '0', STR_PAD_LEFT);
95:
96: $padz = str_pad ($z, 3, '0', STR_PAD_LEFT);
97:
98: $folder = str_replace ('$y', $pady, $folder);
99:
100: $blocks = str_replace ('$x', $padx, $blocks);
101:
102: $template = $commands;
103:
104: $template = str_replace ('folder', $folder, $template);
105:
106: $template = str_replace ('blocks', $blocks, $template);
107:
108: $path = '/home/hadoop/';
109:
110: $path .= 'test';
111:
112: $path .= $padz;
113:
114: $path .= '.sh';
115:
116: if ($x == 1) { $change = ''; } else { $change = '##'; }
117:
118: $template = str_replace ('$$', $change, $template);
119:
120: if ($z == 1) { $change = ''; } else { $change = '##'; }
121:
122: $template = str_replace ('##', $change, $template);
123:
124: file_put_contents ($path, $template);
125:
126: chmod ($path, 0777);
127:
128: if ($x == 1) { exec ("$path"); }
129:
130: if ($x != 1) { exec ("$path > /dev/null &"); }
131:
132: echo $path;
```

```
133:
134: echo "\n";
135:
136: $x++;
137:
138: $z++;
139:
140: }
141:
142:
143: $y++;
144:
145: $toolong = false;
146:
147: $still_running = true;
148:
149: $counter = 1;
150:
151: while ($still_running)
152:
153:     {
154:
155:         sleep (10);
156:
157:         exec ("ps > $running");
158:
159:         $echo = file_get_contents ($running);
160:
161:         str_replace ('java', 'java', $echo, $jobs);
162:
163:         if ($counter > 30) { $still_running = false; $toolong = true; }
164:
165:         if ($jobs == 0) { $still_running = false; }
166:
167:         $counter++;
168:
169:     }
170:
171: }
172:
173: if ($y > 1000) { $quit = true; }
174:
175: if ($toolong) { echo "Hadoop taking too long to respond.  Job Terminated prematurely."
176:     ; $quit = true; }
177: }
178:
179: ?>
180:
```