Monster metrics info

Friday, October 06, 2017   7:51 AM

MonsterChart for JobCreationTimeOnProd:
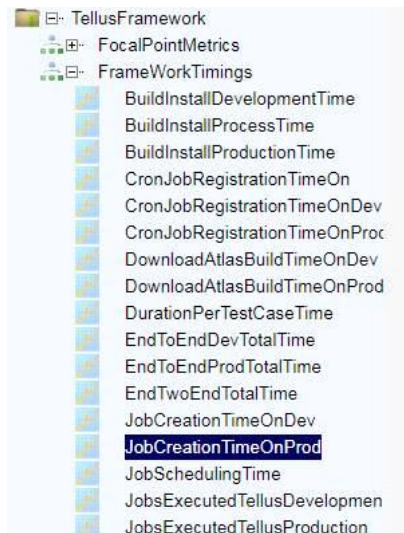http://monster.factset.com/monsterchart/chart/chart.aspx?statusid2=D1383151LI58162000

MonsterChart for CronJobRegistrationTimeOnProd:
http://monster.factset.com/monsterchart/chart/chart.aspx?statusid2=D1383151LI58125113

Monster metric for JobCreationTimeOnProd (no direct link)
1. Navigate to URL: http://monster.factset.com/metrics/
2. Go to TellusFramework -> FrameWorkTimings
3. Click on JobCreationTimeOnProd to see metrics about timing on prod (you can see GroupIDs for jobs that have been submitted and their timings)

For timing about other monster metrics, you can find them under the same category - TellusFramework -> FrameWorkTimings.

Alert profile in Monster: http://is.factset.com/Docs/monster/CreatinganAlertProfile.html

Here is the url used to create a new alert profile: http://monster.factset.com/Search/AlertProfile/New

Sample alert JobsExecutedTellusProduction:
http://monster.factset.com/Search/alertProfile/Display/f0fdd7a6-9b07-4725-b068-37b7c54d8539

Tellus data for job metrics

Tuesday, March 20, 2018   8:07 AM

Data Schema for Generating Metrics

```typescript
// (TypeScript)

/**
 * "Chunk" = execution worker,
 * 1-to-1 to resource (FPE VM, Selenium session)
 */
export interface TellusChunk {
  chunkID: number;
  groupID: number;
  eventID: number;
  scheduledStart: moment.Moment;
  start: moment.Moment;   // actual start
  end: moment.Moment;     // actual end
  passed: number;         // test cases executed and passed
  failed: number;         // test cases executed and failed
  total: number;          // test cases executed
  status: string;
  environment: string;    // test environment

  // platform, browser are not relevant to SITINT, SoapUI, WST
  platform?: string;
  browser?: string;
}
```

Data Query

```sql
SELECT c.chunkID, g.groupID, e.eventID,
       g.startDateTime AS scheduledStart,
       r.startDateTime AS [start],
       r.endDateTime AS [end],
       r.numOfPassedTests AS passed,
       r.numOfFailedTests AS failed,
       r.totalTestNums AS total,
       r.status,
       e.testEnvironment AS environment,
       e.platform,
       e.platformName,
       e.browserType,
       e.sgBrowserName
    FROM SchedulerTestGroups g
      JOIN SchedulerTestGroupChunks c ON g.GroupID = c.GroupID
      JOIN CalendarInstances i ON g.InstanceID = i.InstanceID
```

```sql
    JOIN CalendarEvent e ON i.EventID = e.EventID
    LEFT JOIN TellusTestResult r ON g.groupID = r.groupID AND c.ChunkID =
r.ChunkID
  WHERE g.IsDeleted = 0
    AND i.IsDeleted = 0
    AND e.IsDeleted = 0
    AND ((g.startDateTime >= '${startStr}' AND g.startDateTime < '${endStr}')
      OR (r.startDateTime >= '${startStr}' AND r.startDateTime < '${endStr}')
      OR (r.endDateTime >= '${startStr}' AND r.endDateTime < '${endStr}'))
  ORDER BY g.groupID
```

Data Transformation

```typescript
export function sanitize(row: TellusTestResult) {
  if (row.failed === null) {
    row.failed = 0;
  }
  if (row.passed === null) {
    row.passed = 0;
  }
  if (row.total === null) {
    row.total = 0;
  }

  if (row.passed + row.failed < row.total) {
    row.status = TestStatus.Error;  // incomplete => environmental error
  } else if (row.status === null) {
    row.status = TestStatus.Error;
  } else {
    row.status = row.status.toLowerCase();
  }

  row.environment = row.environment === null ? 'error' :
row.environment.toLowerCase();
  if (row.platform !== null) {
    row.platform = row.platform.toLowerCase();
  }
  if (row.platformName !== null) {
    row.platformName = row.platformName.toLowerCase();
  }
  if (row.browserType !== null) {
    row.browserType = row.browserType.toLowerCase();
  }
  if (row.sgBrowserName != null) {
    row.sgBrowserName = row.sgBrowserName.toLowerCase();
  }

}

/**
```

```typescript
 * Merge (platform, platformName) into platform.
 * @param platform platform column
 * @param platformName column
 * @return merged platform; returns undefined if platform is not applicable to
environment.
 */
export function mergePlatform(platform: string, platformName: string,
environment: string): string | undefined {
  // FPE is always 'windows' (no platform or platformName)
  if (environment === 'fpe') {
    return 'windows';
  }
  // SGE should have platform but not necessarily PlatformName
  if (environment === 'sge') {
    if (platform === null) {
      return 'error';
    }
    return platform === 'mac' ? platformName :  platform;
  }
  return undefined;
}


/**
 * Merge browserType and browserName columns into browser.
 * @param browserType browserType column
 * @param sgBrowserName browserName column
 * @return merged browser; undefined if platform is not applicable to
environment.
 */
export function mergeBrowser(browserType: string, sgBrowserName: string,
environment: string): string | undefined {
  if (environment === 'fpe') {
    // FPE TestComplete project type has browserType = '-1', which means
'factset' workstation
    // Note: FPE Selenium project type uses browserType = 'cefmp' for workstation
    if (browserType === '-1' || browserType === null) {
      return 'factset';
    }
    return browserType === null ? 'error' : browserType;
  }
  if (environment === 'sge') {
    return sgBrowserName;
  }
  return undefined;
}
```

Tellus Metrics Collector

Monday, March 19, 2018   12:57 PM

Factset.io Logs: https://dashboard.factset.io/apps/tellus-metrics-collector/logs
Git: https://github.factset.com/app-qa-automation/tellus-metrics-collector
Grafana: http://grafana.factset.io/dashboard/db/qa-automation-tellus-statistics?orgId=1