In [27]:

```
import json
from datetime import datetime, timedelta

import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

In [28]:

```
#fname = 'bigpanda.hc_20324124.json'
#fname = 'bigpanda.hc20323822.tid1313.json'
fname = 'bigpanda.hc20324124.tid1337.json'
jd = json.load(open(fname))
```

In [29]:

```
jd.keys()
```

Out[29]:

```
dict_keys(['selectionsummary', 'jobs', 'errsByCount'])
```

In [30]:

```
len(jd['jobs'])
```

Out[30]:

```
959
```

In [ ]:

```
jd['jobs'][0]
```

In [31]:

```
df = pd.DataFrame(jd['jobs'])
len(df)
```

Out[31]:

```
959
```

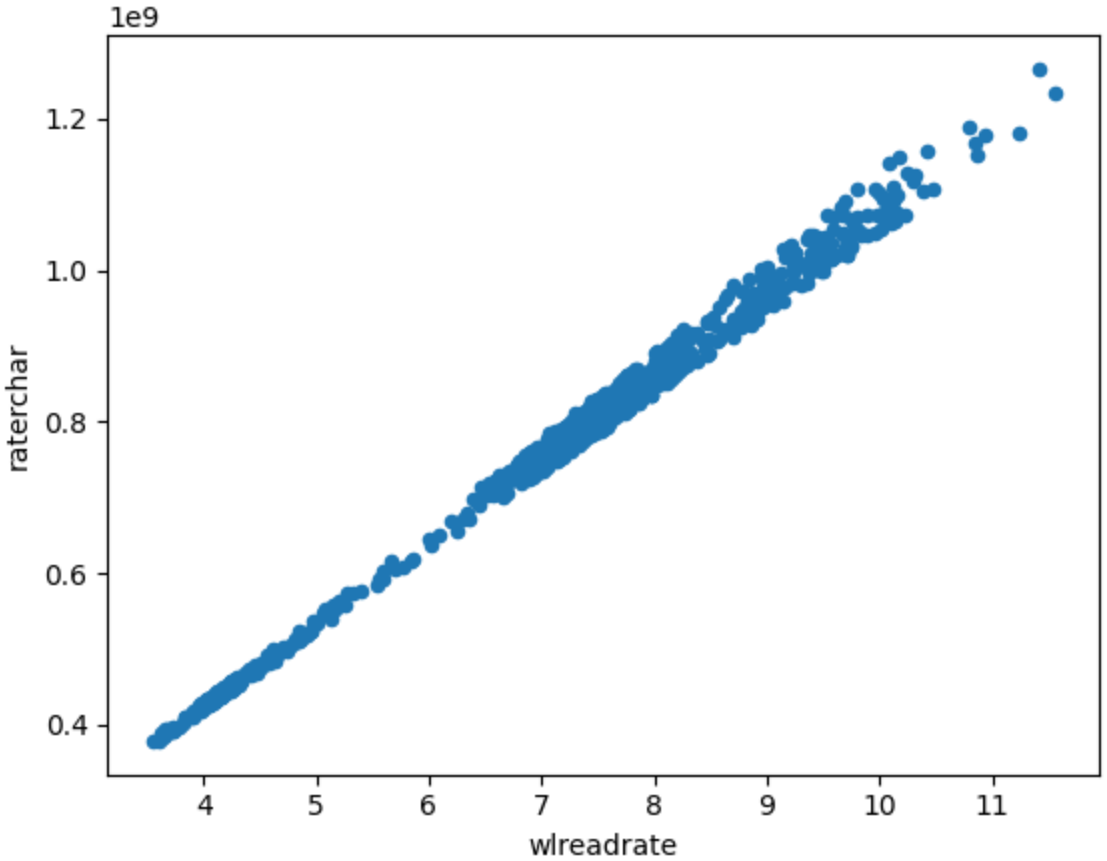
In [45]:

```
# cleanup dataset and add some further parameters
df = df[df.jobstatus=='finished'] # only finished jobs
print(len(df))

df['readfrac'] = df.totrchar*1024/df.inputfilebytes
# calculate input rate in MB/s
df['readrate'] = df.totrchar/1024/df.durationsec
df['evtrate'] = df.nevents/df.durationsec
# convert start/end time to date
df['starttime'] = pd.to_datetime(df['starttime'])
df['endtime'] = pd.to_datetime(df['endtime'])
df['cputype']=[x[2:16] for x in df.cpuconsumptionunit]
# work load run-time from pilottiming list
df['wlruntime'] = [int(x.split('|')[2]) for x in df.pilottiming]
df['wlreadrate'] = df.totrchar/1024/df.wlruntime
df['wlcpueff'] = df.cpuconsumptiontime/df.wlruntime
```

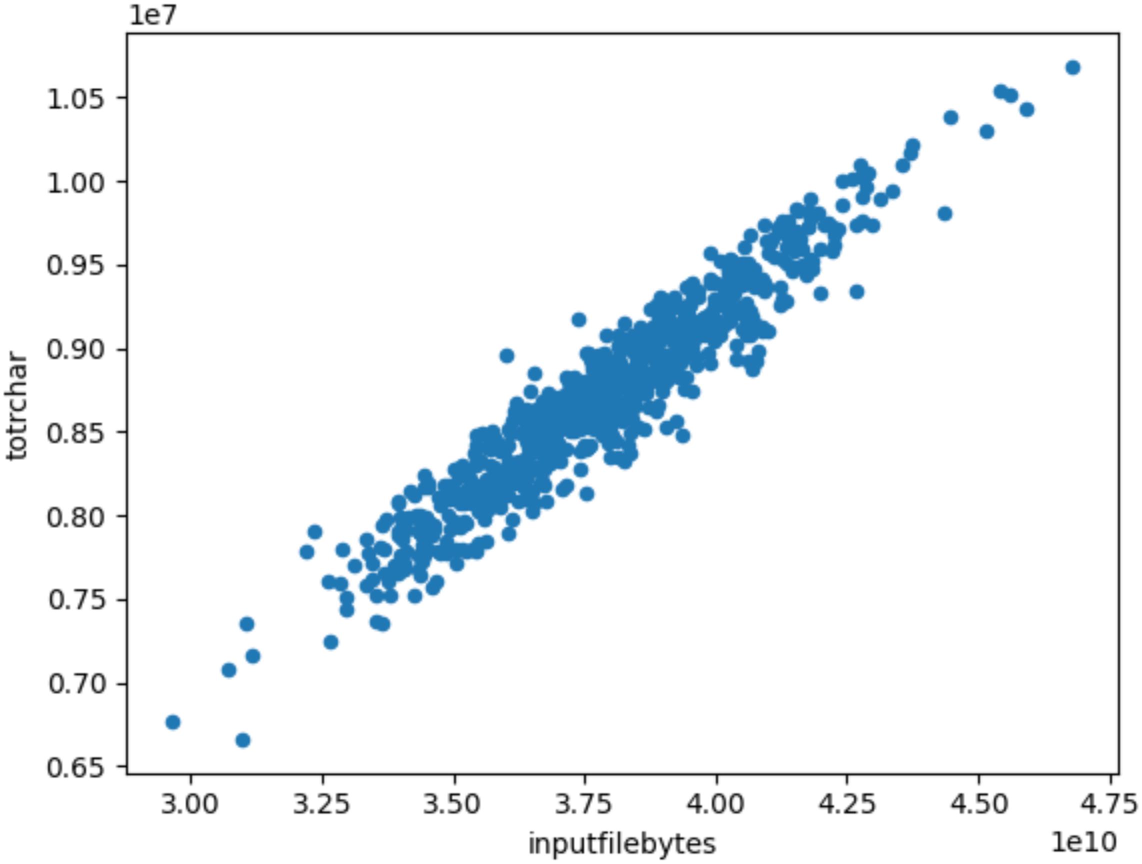
```
799
```

In [34]:

```
df.plot.scatter('wlreadrate','raterchar');
```



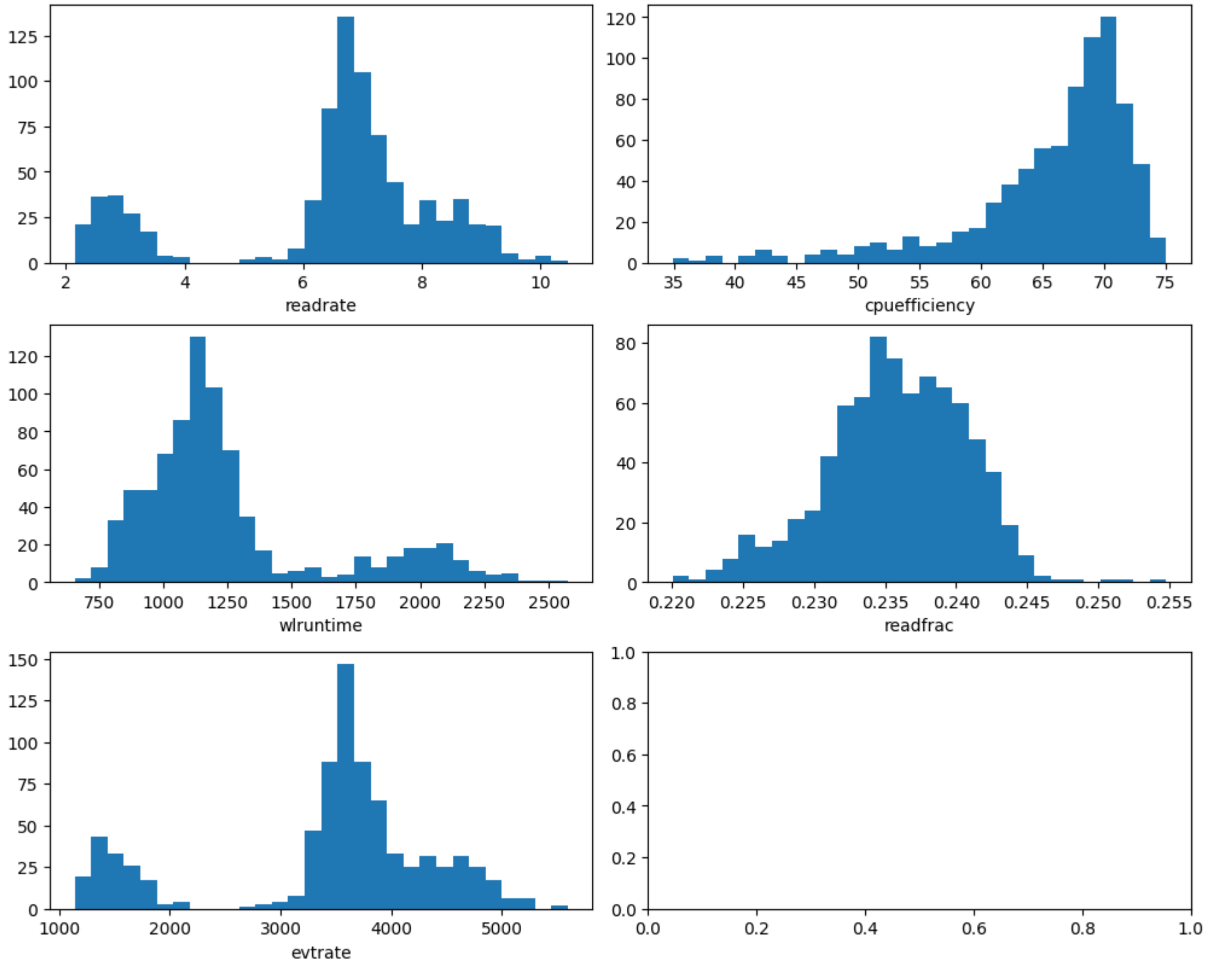
```
In [35]: df.plot.scatter('inputfilebytes','totrchar');
```



```
In [37]: print(f'total GB read {df.totrchar.sum()/1e6:.3f}\ntotal GB filesize {df.inp
```

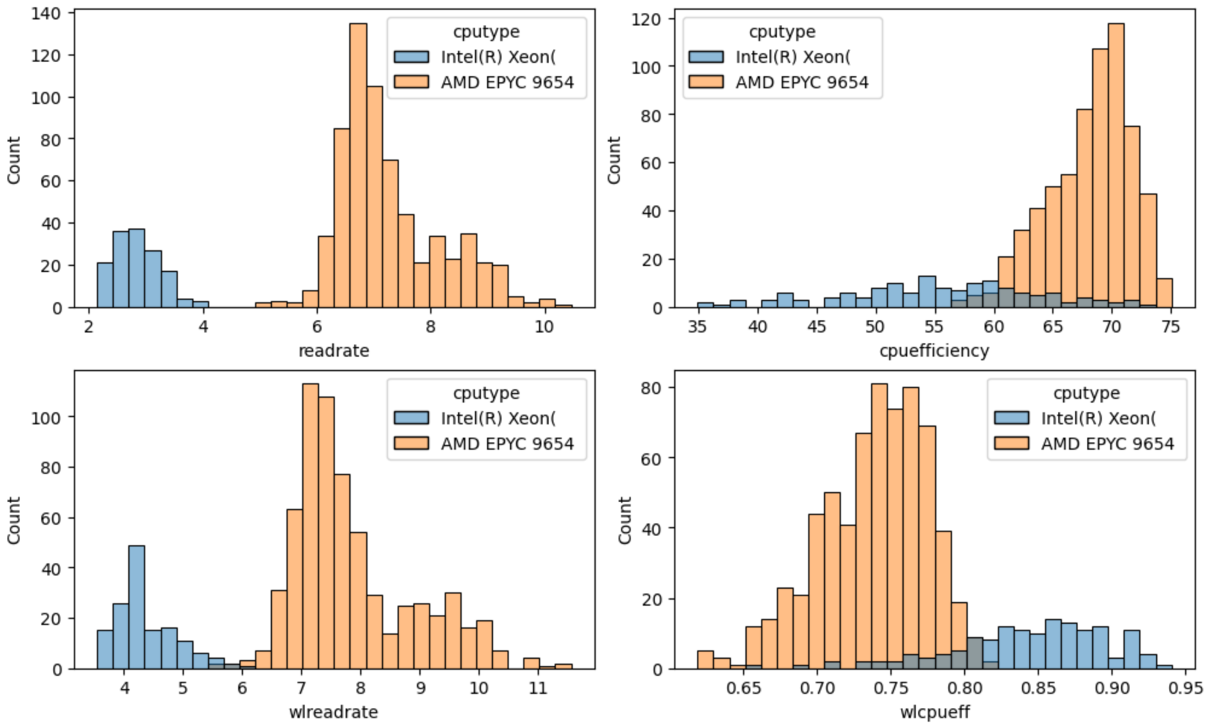
```
total GB read 6973.351
total GB filesize 30278.445
```

```
In [47]: # some basic dists

         fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(10, 8),constrained_layou
         pcols = ['readrate','cpuefficiency','wlruntime','readfrac','evtrate']

         for x,p in zip(axes.flatten(),pcols):
             x.hist(df[p],bins=30)
             x.set_xlabel(p)
```



```
In [48]: #df.hist('readrate',by='cputype',sharex=True)
         # Create subplots
         fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(10, 6), constrained_layo

         # Using seaborn's displot for overlayed histograms
         axes = axes.flatten()
         sns.histplot(data=df, ax=axes[0], x='readrate', hue='cputype', bins=30)
         sns.histplot(data=df, ax=axes[1], x='cpuefficiency', hue='cputype', bins=30)
         sns.histplot(data=df, ax=axes[2], x='wlreadrate', hue='cputype', bins=30)
         sns.histplot(data=df, ax=axes[3], x='wlcpueff', hue='cputype', bins=30)
```

```
Out[48]: <Axes: xlabel='wlcpueff', ylabel='Count'>
```



In [40]:

```
# timeline of jobs and IO rate

st = df.starttime.min().floor('min')
et = df.endtime.max().ceil('min')
minutes_diff = (et-st).total_seconds() / 60
st,et,minutes_diff
```

Out[40]:

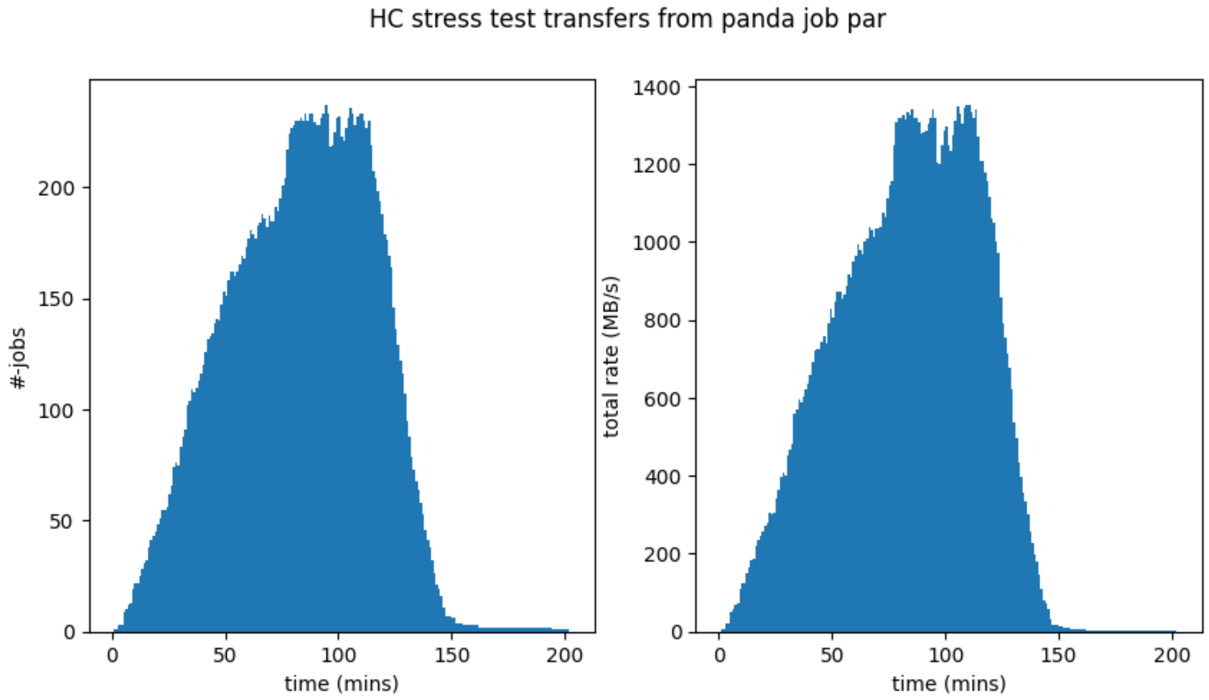
```
(Timestamp('2025-11-17 09:19:00'), Timestamp('2025-11-17 12:41:00'), 202.0)
```

In [56]:

```
# calculate sum of running jobs and sum of transfer-rate vs time

nbins = int(minutes_diff+1)
bins = np.arange(nbins+1)
counts = np.zeros(nbins)
trate = np.zeros(nbins)
ct = st
for i in range(nbins):
    counts[i] = df[(df.starttime<ct) &  (df.endtime>ct)].readrate.count()
    trate[i]  = df[(df.starttime<ct) &  (df.endtime>ct)].readrate.sum()
    ct += timedelta(minutes=1)
```

In [57]:

```
fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(10, 5))
ax = axes[0]
ax.hist(bins[:-1],bins,weights=counts)
ax.set_xlabel('time (mins)')
ax.set_ylabel('#-jobs');

ax = axes[1]
ax.hist(bins[:-1],bins,weights=trate)
ax.set_xlabel('time (mins)')
ax.set_ylabel('total rate (MB/s)');
fig.suptitle('HC stress test transfers from panda job par');
#fig.savefig('hc_stress_es_jobpar.png')
```



In [ ]: