**public void** selectVp(BulkloadContext bldCtx, **int** curOff, **int** curLen, **int**[] values,**float**[] disBuf, SelectVpResult result) {  
 **int** spSize = Math.max((**int**) (curLen \* conf.ratio), 1);  
 SampleResult sampleResult = **new** SampleResult(bldCtx.spBuf, spSize),  
 SampleResult sampleResultInner = **new** SampleResult(bldCtx.spBufInner, spSize);  
 *//随机抽取候选优先点* sampler.sample(values, curOff, curLen, sampleResult);  
  
 **float** maxStdev = -1;  
 **for** (**int** i = 0; i < spSize; i++) {  
 Geometry candidate = bldCtx.geometries[bldCtx.spBuf[i]];  
 sampler.sample(values, curOff, curLen, sampleResultInner);

//随机抽取参考点  
 **for** (**int** j = 0; j < spSize; j++) {  
 ……//计算候选点与对应参考点的距离  
 }  
 **float** current = computeStdev(disBuf, 0, spSize);

//计算当前候选点的标准差  
 **if** (current > maxStdev) {  
 maxStdev = current;  
 result.vpIndex = bldCtx.spBuf[i];  
 result.vpGeometry = candidate;  
 }  
 }  
}